Retraction Notice

The Editor-in-Chief and the publisher have retracted this article, which was submitted as part of a guest-edited special section. An investigation uncovered evidence of systematic manipulation of the publication process, including compromised peer review. The Editor and publisher no longer have confidence in the results and conclusions of the article.

LS did not agree with the retraction. YM, YL, and LC either did not respond directly or could not be reached.

Software development and design of network security system based on log data

Lin Shi,[a,b,c,*,†] **Yang Ma,**[a,†] **Yan Lv,**[a,†] **and Liquan Chen**[a,†]
[a]Southeast University, School of Cyber Science and Engineering, Nanjing, China
[b]Nanjing University of Aeronautics and Astronautics, College of Economics and Management, Nanjing, China
[c]Jiangsu Financial Information Management Center, Nanjing, China

Abstract. With the rapid development of computer network technology and the fact that the network brings convenience to people's lives and work, security issues cannot be ignored. Various countries and social organizations build a network security vulnerability database based on big data analysis technology, use network offensive and defensive platforms to train network security talents, and the target environment as a core component in the offensive and defensive platforms plays an important role in practice. We construct a large network security data set with a security knowledge system, analyze the current construction technology and vulnerability data characteristics of major vulnerability databases at home and abroad from multiple levels, and propose a vulnerability information correlation technology based on a combination of general vulnerability numbers and specific vulnerability numbers. Also, we comprehensively analyze each vulnerability database, assess the comprehensiveness and coverage of the vulnerability data information covered, and establish the source of the vulnerability data, data reference, security content automation protocol support, proof of concept, platform environment affected by the vulnerability exploitation code situation, and other fields of network security big data sets with security knowledge system. Here, we have completed the design and implementation of the user rights management module, the centralized monitoring module, and the service desk and configuration change management module. The system can accurately monitor the operation of the system in real time, find problems in time, warn of potential risks, and promote the operation and maintenance approach to intelligent. The transformation has improved the efficiency of fault diagnosis and greatly enhanced the safety of the system. Experimental research shows that the software system automatically screens and manually confirms the vulnerabilities that meet the target construction environment, and initially constructs a total of 43 effective target environments, which provides security researchers with the understanding of vulnerability principles, hazards, and targeted network security training. A stable and convenient experimental environment. © *2022 SPIE and IS&T* [DOI: 10.1117/1.JEI.32.1.011207]

Keywords: big data analysis technology; network security; system software development; target environment.

Paper 220301SS received Mar. 22, 2022; accepted for publication Jun. 30, 2022; published online Jul. 25, 2022.

1 Introduction

The research on network security vulnerability data is practical. For the development and design of network security software based on the environment of big data analysis, it is first necessary to layer the logic and determine the logical composition of software function realization. Logic stratification needs to be carried out from three levels: information transmission information extraction and network port docking. For each vulnerability exposed on the Internet, it must be evaluated in terms of its hazard methods and scope of impact. The most effective method is to reproduce the vulnerability through a simulated environment, allowing network security

*Address all correspondence to Lin Shi, shiling717@nuaa.edu.cn

†These authors contributed equally to this study.

1017-9909/2022/$28.00© 2022 SPIE and IS&T

researchers to monitor the vulnerability and analyze the cause of the vulnerability. By researching and analyzing exposed vulnerabilities, it is possible to gain a deeper understanding of vulnerabilities and secure known information systems. In addition, the training of network security personnel also requires a large number of target environments with real vulnerabilities. Through the construction of target environments, the hands-on capabilities of network security personnel will be strengthened to effectively deal with the increasingly serious network security problems.[1,2]

Feature extraction is the most critical step in image recognition. The quality of feature extraction is directly related to the quality of image recognition. The purpose of feature extraction is to convert the image information that cannot be directly recognized by the computer into digital information that can be directly calculated and processed by the computer. In the original image, there are often a large number of features, and these features are not necessarily conducive to image recognition. There are many commonly used feature extraction methods. In deep learning, convolutional neural network has excellent feature recognition ability, and Visual Geometry Group 16 (VGG-16) network is a good way of feature extraction. Therefore, to improve the performance of the image recognition algorithm, it is necessary to select the features of the image. To improve the recognition effect of the algorithm, it is necessary to select features with strong recognition ability; to improve the robustness of the algorithm, it is necessary to extract features with strong anti-interference ability. The features that need to be extracted are often multi-dimensional. To reduce the computational complexity and improve the performance of the algorithm, it is necessary to further simplify them so that the extracted features can more accurately refine the content of the image. Network security means protecting the network and system from illegal access and intrusion. With the popularization of informatization, network security has gradually become an indispensable part of the national information infrastructure, and it has a close relationship with the survival of the country, social stability, and economy. Sreedhar et al. put forward the concept of visualization of network data information. Up to now, network security technology has developed rapidly, especially the performance of various network security products such as: network monitoring, anti-virus software, firewalls and intrusion detection systems has been significantly improved compared to the past few years. However, the demand for network security visualization products is becoming more and more urgent.[3] Although the concept he mentioned is relatively advanced, the technical operation is too complicated and difficult. Gracey and Verones can display port data, denial of service attacks (DoS), and detect worms in a class B network by layering the network data, but scale invariant feature transform does not have real-time data display and automatic alarm functions.[4,5] To facilitate network security personnel to practice network security knowledge, Tang has also provided the configured target environment Metasploitable many times. The Metasploit2 vulnerability attack practice system is a target environment based on VMware technology released earlier, it is actually a packaged operation. The system virtual machine image uses the VMware format, and some typical vulnerability environments are preset in the environment. Users can use VMware Workstation locally to directly open the image to use.[6]

To ensure the maximum popularization of vulnerability information in Internet information systems and devices, network security vulnerability research has become a contention for research in various countries and organizations in recent years. Various countries and organizations have invested a lot of energy in building Internet vulnerability data platforms and attack and defense platforms, to train cybersecurity personnel with practical experience. At present, most countries and organizations in the world are used to prevent network vulnerabilities and consolidate network security tools are mainly artificial intelligence, especially some algorithms of machine learning. The object is only for vulnerability data with a common vulnerability data number. Although the data source is authoritative, the vulnerability data are not comprehensive.[7]

The data of this system comes from the vulnerability data of major vulnerability databases at home and abroad. The vulnerability data contain a large amount, the implementation technology of each vulnerability database is different, and the data field standards are inconsistent. Therefore, it is necessary to study efficient vulnerability data collection technology and standardized data cleaning technology to achieve large-scale, real-time and accurate collection, and cleaning of vulnerability data. In the big data environment, after the software receives the data transmission request, it will automatically start the corresponding database, which maximizes the logical processing efficiency of the software during the operation. By studying the database design principles of major security vulnerability databases, along with the source, knowledge system, and data format of vulnerability data, on this basis use web crawler technology and data cleaning technology to collect and standardize and clean the vulnerability data of each vulnerability database to ensure collection that it is comprehensive, accurate, and effective.[8]

2 Design of Network Security Software System Based on Big Data Analysis Technology

2.1 *Design of a Target System Based on Cybersecurity Big Data*

The entire workflow of the system consists of the network security big data set building module in the upper part and the vulnerability target environment building module in the lower part.

2.1.1 *Vulnerability database building block*

This article uses XML document analysis technology to extract the original vulnerability webpage data according to the data function characteristics and store them in the library.[9] In the process of data cleaning, the CVSS information in the customer premise equipment (CPE) library and vulnerability information is also analyzed according to the field meaning. After all data preparations are completed, according to the system's standard vulnerability database construction standards, data fusion technology is used to build a basic ID-based vulnerability library, a standard vulnerability library with a security knowledge system, a vulnerability index library for vulnerability retrieval, and a visualization. The module displays and applies the vulnerability information.[10–12] The system stores all the processing information in the database for the target environment construction module to extract the target environment construction elements from. It works when you need to defend against vulnerability data and fix your own system.

2.1.2 *Vulnerability target environment building block*

The main function of the vulnerability target environment construction module is to extract the target environment construction element information that can be used for the system from the knowledge-based standard vulnerability library and complete the target environment construction element information through automatic download module and data analysis. After the data analysis and module download are completed, the next focus of the vulnerability target environment construction is on the modularization of virtualization technology. Docker virtualization technology can be used to modularly build a target environment for security vulnerability practice.[13,14] The target environment construction element information extraction stage needs to extract the vulnerability number, vulnerability verification information, vulnerability utilization information, vulnerability type, and software information affected by the vulnerability in the vulnerability entry information according to the target environment construction requirements, and divide them into essential elements and reference elements

2.2 *Target Environment Construction Technology*

2.2.1 *VMware virtual technology*

The advantage of building a target machine environment based on VMware lies in its versatility, it can be applied to almost all target systems of vulnerable platforms. The network virtualization technology provided by VMware can realize the network working mode of the target machine environment, which is conducive to building a complex target system, but it also has shortcomings. It occupies storage resources, computing resources are high, and it has no advantage for the large-scale construction and release of the target environment.

2.2.2 *Docker virtual technology*

Docker provides a set of portable standardized configuration mechanisms that can run the same container on different hosts without any difference. Since Docker containers are reusable, users can expand locally based on a basic container, which greatly saves Storage resources and computing resources. Compared with traditional virtual machine technology, Docker container technology is more lightweight and easier to deploy.[15,16]

The biggest advantages of deploying a target environment based on Docker technology are light weight, less resource consumption, strong portability, simple operation, and easy batch deployment. However, due to the design problem of Docker virtualization technology, i.e., containers running on Docker share kernel resources with the host operating system, and the container kernel information cannot be changed. Therefore, Docker virtualization technology is very useful for target environments that involve Linux kernel vulnerabilities. The target environment that also involves vulnerabilities outside the Linux platform cannot be built using Docker virtualization technology.[17,18]

2.3 *Network Security Information Visualization Technology*

In the information visualization pipeline (VisualizationPipeline), the entire process of data from the data source to the user needs to be-series of data transformation and processing. This process can be divided into three parts, details follow:

2.3.1 *Raw data preprocessing process*

This process is mainly to convert the original data into regular data and store it in the database. Some experts have classified the data types of raw data into seven types as shown in Fig. 1.

2.3.2 *Display the data table using visual structure*

This process is mainly to find a novel and suitable visualization structure, and map the data table to the visualization structure to display the information that needs to be expressed. This is also the most critical core technology in information visualization.[19,20] Network devices, external

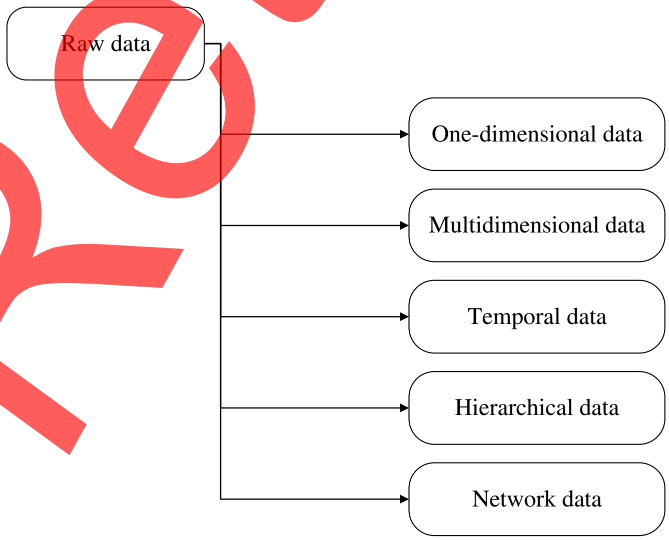

Fig. 1 Classification of raw data types.

systems, and raw traffic were collected by sys-log, webservice, SFTP, and NetFlow, respectively. The collected data were transferred into the data processing queue kafka through flume or ftpd, and the collected files were saved directly into HBase.

2.3.3 *View drawing process*

This process is to establish a view of the final visual structure by defining the position, zooming the graph, and filtering the graph parameters. The arrow from right to left indicates the user's control and adjustment in the entire transformation process, which also shows the good interactive design of information visualization technology.

2.4 *System Functional Framework Design*

2.4.1 *CMDB configuration management system*

Configuration management database (CMDB) configuration management system is the core part of IT business service management, IT services all other IT management platforms. It has a wide application in the financial industry, the Internet industry, and the service industry, and provide configuration items and relationships, business system modeling, visualization, and other configuration information functions.[21,22] Performing unified CMDB configuration data processing for various IT management tools reduces operation and maintenance costs, reduces risks, and simplifies integration work.[23] The configuration of the IT technical architecture as a single data source can help administrators understand in real time, accurately, and make correct management decisions.

2.4.2 *Centralized monitoring platform*

Strengthen the effective management of IT resources, establish a centralized monitoring management platform, the existing monitoring system expands in the breadth and depth of management, the platform fully realizes the monitoring of systems, databases, middleware, applications, storage management systems, desktops Management systems, other systems in the computer room, etc., adopt a unified performance and event management platform for integration and centralized processing.[24,25] Fully consider the business service model evaluation in CMDB, the impact of failures, integrate IT operation and maintenance process management platform, automatically create and assign task work orders, help operation and maintenance personnel to locate and analyze system failures in detail, and strive to discover and solve the problem and improve the stable running time of the business system.[26,27]

2.4.3 *Automation management platform*

Compared with the monitoring platform that collects, processes, and displays information from the bottom up, the automation management platform uses top-down control and operation actions. It can cover from physical and virtual servers, networks to applications, etc..[28,29] Provide a variety of functions related to operation and maintenance management, such as device discovery, script execution, configuration backup, inspection, change and correction, operating system installation, patch analysis and distribution, application distribution, job scheduling, operation compliance audit, remote management, and data monitoring etc., convert manual operations into automation, reduce the workload of operation and maintenance personnel, and enforce compliance with the requirements of implementing best practices and safety compliance.

2.5 *Detailed Design of Building Blocks of Network Security Big Data Sets*

2.5.1 *Data extraction and cleaning technology*

Metadata are data about data and is the cornerstone part of the entire system platform. We analyzed the original webpage data of each vulnerability database and defined different data conversion rules to convert the collected original data objects into another set of target objects.[30,31]

Both the original data object and the target object are elements in the data object, and the target object is the metadata described by the system.

2.5.2 *Vulnerability data standardization*

The purpose of the standardization of vulnerability data is to unify the format and language expression of the field information of the same expression in different vulnerability libraries. Due to the large differences in the vulnerability information structure and text format of the vulnerability libraries belonging to different institutions, the vulnerability description of the same vulnerability information and even the format display of the same vulnerability knowledge field are also different. We mainly adopted the following three methods to standardize the vulnerability data:

1. Extracting common attributes: we analyzed the raw data of the vulnerabilities of various websites collected by the system and found that all sources of vulnerability data in the vulnerability database have common attributes, including the vulnerability title, vulnerability release time, vulnerability description, vulnerability reference, vulnerability source, etc. Therefore, we first extract these common attributes from the vulnerability information source during the vulnerability data cleaning process.
2. Unify the format of data expressions in similar attributes. We take the release time of vulnerabilities as an example to unify the different time expressions of different database websites into a standard format.

2.5.3 *Deletion of duplicate records*

In the process of standardization of vulnerability data, due to the update of a certain piece of vulnerability information, two similar records will be generated, and similar duplicate records in the same data set will be deleted through keyword detection.

Generally, the common method used in the deletion of duplicate records is to compare the vulnerability data fields one by one, find the time attribute or other attributes similar to delete, this method is inefficient, and very high accuracy can be achieved while taking the time, plus it is easy to delete useful non-duplicated information by mistake. We have adopted an improved multi-field comparison method with keywords as the core. The method is as follows: first analyze the characteristics of the vulnerability data source to generate keywords, e.g., CVE is used as the keyword in the CVE vulnerability database data source, and the CNNVD vulnerability database data source. Use the combination of CNNVD number and CVE number as the key, and then sort the records according to the keywords in the unified data set so that the duplicate records will be arranged in adjacent positions. If an attribute of the duplicate record has multiple values, take a record with a longer attribute record value and a newer modification time attribute. If there is a conflict, throw an exception message and perform manual analysis and processing in expert mode. Data tracking and positioning are mainly aimed at risk information. To achieve the network security goal in the big data analysis environment, the data tracking and positioning function should be introduced in the software program development to automatically find the data source according to the data security results analyzed.

2.5.4 *Detailed design of vulnerability database*

Basic vulnerability database. The basic ID-based vulnerability database is the first-level library of the entire vulnerability database system. It uses the vulnerability data fusion method based on the general vulnerability number and supplemented by the specific vulnerability number through the correlation of the vulnerability data collected by the system. The primary vulnerability database constructed. The primary goal is to ensure the comprehensiveness of the vulnerability information and reflect different vulnerability databases in the basic vulnerability database. Redundancy and heterogeneity of vulnerability data is minimized according to the correlation between data.

Standard vulnerability database. The standard vulnerability database is to further standardize and analyze the vulnerability data based on the ID-based basic vulnerability database.

The purpose of the standard vulnerability database is to learn and standardize the vulnerability data according to the standard elements of the vulnerability based on security content automation protocol (SCAP). Knowledge analysis of vulnerability information in the vulnerability database according to system design standards. The standard vulnerability database provides important data support services for building target environments. The knowledge system included in the vulnerability data standardization module contains basic vulnerability information, part of SCAP element descriptions, vulnerability verification information, vulnerability utilization information and fine-grained element information related to vulnerability utilization, vulnerability patch information and solutions, etc.

3 Experimental Design of Network Security System Software Based on Big Data Analysis Technology

3.1 *Build a Target Environment Based on Docker Virtual Technology*

The target environment construction is based on the completed and extracted target environment construction elements, using Docker virtualization technology to build a target environment containing real vulnerabilities that can be learned and used by network security researchers. If the software in the filtered vulnerability entry has been downloaded, we prefer to use the saved software to install in Docker. The downloaded packages have no special requirements for installation, but reserve a minimum of 2 GB.Taking the application software of the Debian/Ubuntu operating system as an example, install the downloaded deb package through the dpkg command. After the specific version of the software is installed, professionals need to participate in the basic configuration of the software environment to meet the vulnerability verification. And the final is to verify the vulnerabilities of the target environment. The implementation method is to obtain the exploit file or script code from the construction elements of the vulnerability target environment and to test or verify the vulnerability of the target environment. This process also requires professionals to test and modify the corresponding exploit Files, deploy the operating environment of the exploit program, and test whether the exploit is effective in the target environment.

After completing all the above steps, rebuild the Docker container through the Dockerfile method, and after confirming that it is correct, upload the Dockerfile and the required software installation package, configuration file, vulnerability exploit file, and other contents to the Gitlab server that has been built. Remote security researchers log in to Gitlab from anywhere, find the target environment they need, download all the information of the target environment, construct the target environment container through Dockerfile, and verify it with the given vulnerability exploit or verification file. The target environment can also be directly deployed in the existing offensive and defensive platform for the training and competition of team members.

3.2 *Extraction of Elements to Construct Target Environment*

We analyze the vulnerability element information that is closely related to the deployment of the target environment, including the software information affected by the vulnerability, vulnerability utilization information, and vulnerability verification information that are required to build the target environment. Therefore, we must first complete the information of these vulnerability elements. The implementation of the supplementary method of the vulnerability link element of the download link is as follows:

1. Search the vulnerability data in the standard vulnerability data in sequence, extract the ID value of the vulnerability data items that need to be completed, and establish a directory identified by the path of the vulnerability ID number.
2. Extract software download links, vulnerability exploitation files, or vulnerability verification file download links affected by the vulnerability from the fields corresponding to the vulnerability data that needs to be completed.
3. Pass the URL download link extracted in the previous step to the download function module of the system, and the download function downloads the corresponding file to the specified directory according to the URL link. The implementation of the download

function module mainly uses wget technology. Wget is a free software tool that supports agent's batch download and breakpoint resume, which meets the needs of the system to download data in large quantities.

3.3 *Construction of Standard Vulnerability Database*

The CVE number collection in the CVE vulnerability database is the most comprehensive. The website is responsible for the application and confirmation of the CVE number. The number determination process is: first, security researchers find a specific product vulnerability, apply for the vulnerability number to MITRE Corporation, and MITRE will number the vulnerability. The endowment work was transferred to its CNA members. CNA submits the vulnerability to members for review and generates a vulnerability number. Mitre identified the vulnerability as reserved, indicating that Mitre has received the vulnerability and assigned a number, but the quality of the vulnerability has not been verified. Then the vendors on the Mitre and CNA lists disclose the vulnerability information and confirm that the disclosure is true and effective. The national vulnerability database (NVD) website only includes valid CVE vulnerabilities that have been confirmed and verified. The most collected is the CVE vulnerability database, which contains a total of 111,700 CVE data, followed by the vulnerability data in NVD, which contains a total of 94,000 CVE data.

4 Experimental Analysis of Network Security System Software Based on Big Data Analysis Technology

4.1 *Database Implementation*

In the selection of the database, considering that the read speed requirements are very high, the efficiency requirements are more stringent. Among the popular multiple database platforms, MySQL has multiple database engines, and the read speed of ISAM and MyISAM is very fast, which can provide very good performance and meet the needs. However, their disadvantage is that the information omission is easy to occur when the reading speed is too fast, so the accuracy needs to be improved.

As shown in Table 1, after the pre-processing of the original data, a regular data source will be obtained. After the data fields are counted, PowerDesigner is used for data design. This article discusses the characteristics of the original data after analysis, such as which fields. Table 1 gives the firewall log database table design and intrusion detection log database table design for the fields analyzed in this article.

This paper searches the vulnerability data in the standard vulnerability data in turn, extracts the ID value of the vulnerability data items that need to be completed, and establishes a directory identified by the path of the vulnerability ID number. Extract the download link of the software download link, vulnerability exploit file, or vulnerability verification file affected by the vulnerability from the field corresponding to the vulnerability data that needs to be completed. Pass the URL download link extracted in the previous step to the download function module of the system, and the download function downloads the corresponding file to the specified directory according to the URL link. The implementation of the download function module mainly uses wget technology. Wget is a free software tool that supports agent's batch download and

Table 1 Firewall log PowerDesigner database design.

Id	Bigint	Identity
Time	Datetime	Not null
Syslog	Varchar (20)	Null
Protocol	Varchar (20)	Null
Sourceport	Integer	Null
Direction	Varchar (20)	Null

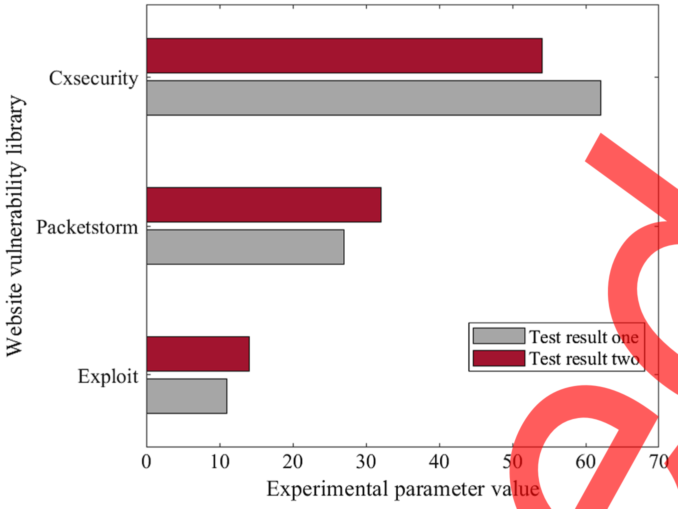

Fig. 2 The amount of exploit information from different vulnerability libraries.

breakpoint resume, which meets the needs of the system to download data in large quantities. The experimental results are shown in Fig. 2.

It can be seen from the data that the information mainly comes from three vulnerability libraries, the Packetstorm website vulnerability library, the Exploit-DB website vulnerability library, and the Cxsecurity website vulnerability library. The proportion of the vulnerability exploitation information is shown in Fig. 2. _id indicates the source of exploit information in the element information, and exploit_range indicates the execution category of exploit that is effective for exploit, including five types of remote, DoS, webapps, local, and shellcode. The source and reference page of the exploit information marked in the exploit field.

4.2 *System Vulnerability Data Analysis and Display*

Based on the network security big data set target system, the vulnerability information of the major vulnerability libraries at home and abroad is collected, and a database containing elements of vulnerability data sources, vulnerability utilization information, vulnerability patch information, vulnerability scoring information, and other elements is constructed. The security element information is correlated and contains a wealth of security knowledge information. Based on the vulnerability data set, we can conduct a series of vulnerability big data analysis and display the relevant content of the vulnerability information elements of the vulnerability database through the visualization page, mainly including the number of vulnerabilities trend graphs, distribution maps of vendor vulnerabilities, distribution maps of vulnerability scores, distribution maps of vendor vulnerabilities, and display maps of industrial control vulnerability databases based on standard leak databases. The front page of the visualization page shows a real-time quantitative graph of the vulnerability information collected by the system.

The system counts the number of vulnerabilities in all authoritative vulnerability databases on the Internet since 2015, and draws and displays the relevant distribution curves. As shown in Fig. 3, it can be intuitively seen that with the rapid development of the Internet, the amount of vulnerability data has been increasing year by year.

4.3 *Vulnerability Search*

Querying vulnerability information by manufacturer or product name. Enter the name of the manufacturer or product to query all vulnerability information related to the manufacturer or product. Enter Microsoft to find all the vulnerability information about the manufacturer, the effect is shown in Table 2.

As shown in Table 2, query the CWE-related vulnerability information according to the CWE number. If you enter CWE-119, the system will find all the vulnerability entries of this type

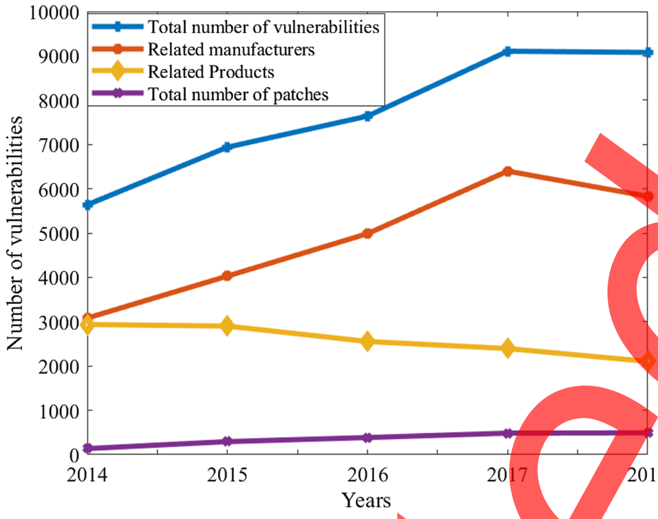

Fig. 3 Distribution curve of vulnerability data.

Table 2 Vulnerable product information retrieval result graph.

ID	CVE
1103	CVE-2017-6852
2027	CVE-2017-0195
3017	CVE-2017-0093
8023	CVE-2017-0092

based on the vulnerability type for system users to refer to. This article queries the corresponding vulnerability information in the platform field of the database based on the vulnerability platform information. Keyword query, as long as the user enters any keyword information related to the vulnerability, the fuzzy matching method can be used to query the related information. The regular matching query, and the user enters the regular expression, can search for related vulnerability information in the vulnerability description information.

4.4 *Target Environment Deployment Example*

The main focus of target environment deployment is the accuracy of the extraction and matching of the target environment components and the performance of the target environment based on Docker container technology. This paper shows the general vulnerability that is extracted from the network security big data set by the target environment component extraction algorithm. The theoretical results of the number, vulnerability software information, and vulnerability utilization information are shown in Fig. 4.

As shown in Fig. 4, the total number of vulnerability data that can be deployed in the target environment is 30,177. Since 2011, the total number of vulnerability data has been 9218. The number of changes in each year is shown in Fig. 4. According to the system design requirements, the platform field in the vulnerability data is filtered, and the target environment entries that can be used for Docker virtual technology deployment are 3380.

The standard vulnerability database application is expanded, and the standard vulnerability database contains a wealth of security knowledge information. On the basis of this database, a proprietary vulnerability database required by security researchers can be built. We take the industrial control vulnerability database as an example, first targeting industrial control systems Industry characteristics, the industrial control keywords are selected from the standard vulnerability database, such as Siemens, Schneider-Electic, etc. The keyword collection is used to

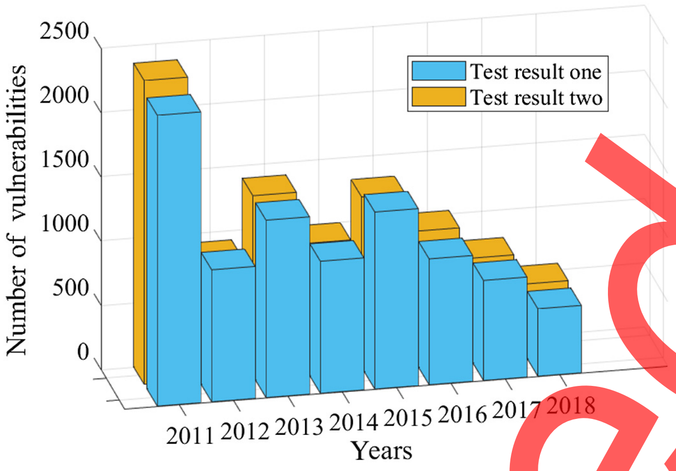

Fig. 4 A graph of the number of universally numbered vulnerabilities that can build targets.

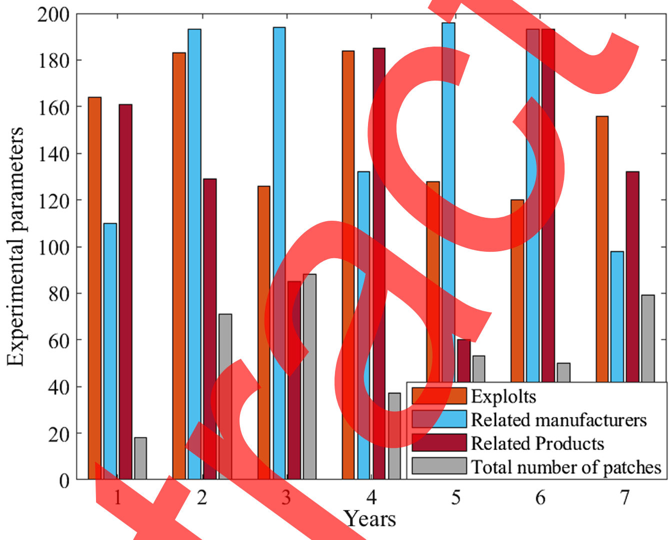

Fig. 5 Industrial control vulnerability database view display.

match the CPE fields in the standard vulnerability database to generate an industrial control system vulnerability database with a data volume of 1330, the visual view of the industrial control vulnerability database is shown in Fig. 5.

From the test examples given above, we can see that the overall function of the system is complete. The target system based on secure big data has a simple and convenient use process, which can quickly restore the real vulnerability environment and effectively test and verify the vulnerabilities in the environment. We use this system to automatically screen and manually confirm the vulnerabilities that meet the target construction environment, and initially construct a total of 43 effective target environments. We solve various challenges in system construction from both technical theory and engineering practice. The feasibility and practicality of the construction of a target system for cybersecurity big data provides a stable and convenient experimental environment for security researchers to understand the principles of loopholes, hazards, and targeted cybersecurity training.

5 Conclusions

This paper studies and implements the construction of a standard vulnerability database. The current data characteristics of the major security vulnerability databases at home and abroad, as well as the correlation characteristics between the data were analyzed. Based on the system design requirements and standards, natural language understanding, SCAP semantic analysis and other technologies were used to split, merge, and complete the necessary information. A vulnerability data fusion technology combining a common vulnerability number and a specific vulnerability number is constructed, and a standard vulnerability database with a security knowledge system is constructed. Provide strong data support for research on security vulnerability data and construction of vulnerability target environment.

The work done in this paper is to mine network security logs, i.e., firewall logs and IDS intrusion detection logs, extract host connection, and traffic information from the firewall logs, and display them through parallel coordinate axes-host connections within a fixed time situation, such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) connection information. The connection request of the host can provide a reference for the log analyst to focus on the phenomenon of the rapid increase in the number of connections, thus providing clues for the confirmation of a botnet or a distributed denial of service attack in a short period of time. From the IDS intrusion detection log, by analyzing the snort log, we can analyze the attack risk warning at different periods and find out the main attack source so that the network security level is improved.

This paper studies and implements the target environment construction based on the standard vulnerability database. Using the standard vulnerability database that the system has established, research the target environment construction conditions, automate the completion of vulnerability utilization information and software information affected by the vulnerability, and extract the target environment construction elements according to the system design requirements. Target environment construction conditions, using Docker host virtualization technology, Linux system environment software dependent time tracing technology, and other standardized, modular construction and deployment of target environments containing real vulnerabilities, providing lightweight and convenient testing for network security vulnerability research and practice training environment.

Disclosures

The authors declare that they have no conflict of interest to disclose.

Acknowledgment

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. S. Wan, “Topology hiding routing based on learning with errors,” *Concurrency Comput. Pract. Experience* **34**(14), e5740 (2020).
2. Z. Lv, A. K. Singh, and J. Li, “Deep learning for security problems in 5G heterogeneous networks,” *IEEE Netw.* **35**(2), 67–73 (2021).
3. C. Sreedhar, N. Kasiviswanath, and P. Chenna, “A survey on big data management and job scheduling,” *Int. J. Comput. Appl.* **130**(13), 531–542 (2015).
4. E. O. Gracey and F. Verones, “Impacts from hydropower production on biodiversity in an LCA framework-review and recommendations,” *Int. J. Life Cycle Assess.* **21**(3), 412–428 (2016).
5. R. Gifty, R. Bharathi, and P. Krishnakumar, “Privacy and security of big data in cyber physical systems using Weibull distribution-based intrusion detection,” *Neural Comput. Appl.* **32**, 23–34 (2019).
6. M. Kuzlu, C. Fair, and O. Guler, “Role of artificial intelligence in the internet of things (IoT) cybersecurity,” *Discov. Internet Things* **1**, 7 (2021).
7. S. P. Samyuktha et al., “A survey on cyber security meets artificial intelligence: AI– driven cyber security,” *J. Cogn. Hum.-Comput. Interact.* **2**(2), 50–55 (2022).

8. I. Butun, P. Österberg, and H. Song, "Security of the internet of things: vulnerabilities, attacks, and countermeasures," *IEEE Commun. Surv. Tuts.* **22**(1), 616–644 (2020).
9. X. Li et al., "Big data analysis of the internet of things in the digital twins of smart city based on deep learning," *Future Gener. Comput. Syst.* **128**, 167–177 (2022).
10. O. Chergui et al., "Can a chaos system provide secure communication over insecure networks?—Online automatic teller machine services as a case study," *J. Electron. Imaging* **27**(3), 033045 (2018).
11. D. Moon et al., "DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks," *J. Supercomput.* **73**(7), 2881–2895 (2017).
12. Z. Lv and H. Song, "Trust mechanism of feedback trust weight in multimedia network," *ACM Trans. Multimedia Comput. Commun. Appl.* **17**(4), 1–26 (2021).
13. J. Neel et al., "The role of context in cognitive systems," *J. Signal Process. Syst.* **78**(3), 243–256 (2015).
14. N. Makitalo et al., "Safe, secure executions at the network edge: coordinating cloud, edge, and fog computing," *IEEE Softw.* **35**(1), 30–37 (2018).
15. M. Samir et al., "PYGRID: a software development and assessment framework for grid-aware software defined networking," *Int. J. Netw. Manage.* **28**(5), e2033 (2018).
16. P. Debiec and A. Materka, "Information technology networked system for student mobility support," *Int. J. Inf. Learn. Technol.* **32**(1), 17–31 (2015).
17. G. Gorbil et al., "Modeling and analysis of RRC-based signalling storms in 3G networks," *IEEE Trans. Emerg. Top. Comput.* **4**(1), 113–127 (2016).
18. E. Y. Jung et al., "Development of U-healthcare monitoring system based on context-aware for knowledge service," *Multimedia Tools Appl.* **74**(7), 2467–2482 (2015).
19. F. Wu et al., "A new and secure authentication scheme for wireless sensor networks with formal proof," *Peer-to-Peer Netw. Appl.* **11**(1), 1–20 (2018).
20. C. Yoon et al., "Flow wars: systemizing the attack surface and defenses in software-defined networks," *IEEE/ACM Trans. Netw.* **PP**(6), 1–17 (2017).
21. S. Hyun et al., "Interface to network security functions for cloud-based security services," *IEEE Commun. Mag.* **56**(1), 171–178 (2018).
22. D. Moon et al., "RTNSS: a routing trace-based network security system for preventing ARP spoofing attacks," *J. Supercomput.* **72**(5), 1740–1756 (2016).
23. T. Shinozaki, Y. Yamamoto, and S. Tsuruta, "Context-based counselor agent for software development ecosystem," *Computing* **97**(1), 3–28 (2015).
24. S. Jaiswal and D. Gupta, "Security engineering methods—in-depth analysis," *Int. J. Inf. Comput. Secur.* **9**(3), 180–211 (2017).
25. B. Yang and M. Yang, "Data-driven network layer security detection model and simulation for the Internet of Things based on an artificial immune system," *Neural Comput. Appl.* **33**, 655–666 (2021).
26. E. Deveci and M. U. Caglayan, "Model driven security framework for software design and verification," *Secur. Commun. Netw.* **8**(16), 2768–2792 (2015).
27. T. Dash, "A study on intrusion detection using neural networks trained with evolutionary algorithms," *Soft Comput.* **21**(10), 2687–2700 (2017).
28. J. Doyle, "Routing TCP/IP Volume I (CCIE Professional Development)," *Tailieu Vn* **21**(3), 392–393 (2017).
29. I. Garitano, S. Fayyad, and J. Noll, "Multi-metrics approach for security, privacy and dependability in embedded systems," *Wireless Personal Commun.* **81**(4), 1359–1376 (2015).
30. Z. Li, Z. Tang, and Y. Yang, "Research on architecture of security video surveillance network cascade system with big data," *World J. Eng.* **13**(1), 77–81 (2016).
31. G. Chen et al., "FUSO: fast multi-path loss recovery for data center networks," *IEEE/ACM Trans. Netw.* **26**(3), 1376–1389 (2018).

Lin Shi is a doctoral candidate. He received his master's degree from the University of Electronic Science and Technology, P.R., China. His research interests include computational intelligence, information security, and big data analysis.

Yang Ma is a senior engineer in the field of cyber security and the director of network security management department of Jiangsu Provincial Communication Administration. His research interests include information security, cryptography, and network security protocol.

Yan Lv received his BS degree in information engineering from Southeast University, China, in 2006, and his MS degree in system on chip design from Lund University, Sweden, in 2009. His current research interests include 5G industrial internet and network security.

Liquan Chen received his PhD from Southeast University, China, in 2005. He was a visiting scholar at National University of Singapore from 2011 to 2012. He has been a professor at Southeast University since 2018. His research interests include information security, cryptography, and network security protocol.