Intelligent detection system of asset security vulnerability hidden danger under multiple and heterogeneous Web network

Yonggang Li1,a*, Jie Li2
1State Grid Information and Telecommunication Group Co. LTD, Beijing, China
2State Grid Tianjin Electric Power Research Institute, Tianjing, China
a*hanmin@sgitg.sgcc.com.cn
a*liyonggang@163.com

Abstract—Traditional network security vulnerability detection system is affected by detection tools, and the detection accuracy of complex network vulnerability is low. Therefore, this paper designs an intelligent detection system of asset security vulnerabilities under the multi heterogeneous web network. By crawling the asset security data in web network by crawler, the crawling data is integrated by using the characteristics of multi heterogeneous network. According to different types of security vulnerabilities, vulnerability detection is carried out to realize the intelligent detection of network asset security vulnerabilities. In order to verify the detection effect of system security vulnerabilities, comparative experiments are designed. The results show that the average false alarm rate of the designed system is only 12.3%, which is nearly 40% lower than that of the traditional system, which effectively improves the vulnerability detection accuracy.

1. INTRODUCTION
Due to the particularity of asset management, asset management network has the characteristics of openness and freedom, while the traditional asset security vulnerability hidden danger detection system needs to use a number of tools, detection tools not only need to set more parameters, but also need to store a large vulnerability library. This traditional vulnerability detection system can only be applied to closed networks, and it is difficult to guarantee the detection effect of large and complex networks [1-2].

Multivariate and heterogeneous web networks can provide better QoS services for users according to their characteristics and business requirements. At the same time, compared with other types of networks, the security of multivariate and heterogeneous web networks is higher [3]. Therefore, according to the above analysis, this paper will design the asset security vulnerability hidden danger intelligent detection system under the multiple and heterogeneous web network.
2. DESIGN OF AN INTELLIGENT DETECTION SYSTEM FOR HIDDEN TROUBLE OF ASSET SECURITY VULNERABILITIES IN MULTI-COMPONENT AND HETEROGENEOUS WEB NETWORKS

2.1. Data integration of multi heterogeneous Web assets

After crawling the asset security data in the network, the crawler integrates the network asset security data by using the heterogeneity of multi heterogeneous web network. Assuming that the crawled data is \( X = [X_1, X_2, \cdots, X_n] \), the manifold learning algorithm is used to map the data to the low bit space to reduce the interference of noise data on security vulnerability detection[4-6]. According to the basic theory of manifold learning algorithm, after the data is mapped to the low dimensional space, each data point has \( k \) nearest neighbors, and the data points are linearly transformed according to the following formula:

\[
Z_j = B^T (X_j - d)
\]

In formula (1), \( X_j \) is the data point on the same local linear plane of the manifold structure; \( B \) is a linear transformation matrix; \( d \) is the linear translation vector of data points in the local linear plane[7]. The least weighted mean square value of the linear transformation matrix is obtained by eigen decomposition of the linear transformation matrix, and the minimum mean square value of the translation vector is calculated according to the following formula:

\[
d = \sum_{j=1}^{k} w_j X_j / \sum_{j=1}^{k} w_j
\]

\[
S = \frac{1}{k} \sum_{j=1}^{k} w_j (X_j - d)(X_j - d)^T
\]

In formula (2), \( S \) is the weighted covariance matrix of adjacent points; \( w_j \) is the weight value of the error between the forward and reverse transformation results in formula (1). After the parameters of manifold learning algorithm are initialized, the linear transformation matrix and translation vector are reduced. The minimum weighted linear transformation matrix and translation vector are obtained by calculating the error value and the error weight. If the distance between the obtained linear transformation matrix and translation vector value is small, stop iterative calculation and obtain the weight vector of all data points. According to the calculated weight vector, the data is denoised by boxplot. After data denoising, the data is scaled according to its own attributes. All scaled data fall into the same specific range. The switching, rotation and projection techniques in the data warehouse are used to generalize the data. After generalization, the data attributes that cannot describe the security degree of assets are removed.

2.2. Implementation process of asset security vulnerability detection

Aiming at the more harmful injection vulnerabilities, this paper selects SQL injection detection technology based on abnormal feedback to detect. When detecting the injection vulnerability, first construct the detection script and send HTTP request to the detection system server. When the server prompts that there is an error in the HTTP request, analyze the HTTP request message returned by the server. According to the feature sample set of capital security vulnerability in the security vulnerability database, according to the following formula, determine whether the detection script contains vulnerability feature information through clustering algorithm.

\[
sim(X_i, Y_j) = \left( \sum_{i=1, j=1}^{n} |x_i - y_j|^p \right)^{1/p}
\]
In formula (3), $X_i$ is the vulnerability feature set in vulnerability database; $Y_j$ is the test script data set; $p$ is the dimension of the dataset; $\text{sim}(X_i, Y_j)$ is the similarity between the elements in vulnerability feature set and detection script data set. The larger the similarity value is, the more vulnerability features are contained in the detection script data set. According to the scale of capital management data, the threshold level is set and the similarity level is divided to determine the detection of injection vulnerability. For XSS vulnerability detection, we first extract tags that may be input by HTML and URLs with parameters. After that, the attack vector is constructed together with the test script, and the request is sent to the server. The script response returned by the server is analyzed. If the response request passes the verification, the XSS vulnerability is marked; if the response fails to pass the request, whether the test of the test script is completely completed is judged, and the completion of the test script ends the detection of XSS vulnerability. For other types of capital security vulnerabilities, we can plan and process test scripts, traverse detection and other forms to prevent and detect. So far, we have completed the design of asset security vulnerability intelligent detection system under the multi heterogeneous web network. The following will verify whether the system achieves the expected function through experiments.

3. SYSTEM PERFORMANCE TEST

Asset security vulnerability management is very important for the operation and development of enterprises. Asset security vulnerability is a potential threat to the survival of enterprises. Based on the hardware part of the traditional asset security vulnerability management system, this paper designs an asset security vulnerability intelligent detection system under the multi heterogeneous web network. This section will analyze the advantages and disadvantages of the designed system by comparing with similar traditional systems.

3.1. Test content

In the performance test of the asset security vulnerability detection system, the hardware experiment platform with the two systems has the same parameters, and the same software program is used to process the experimental data, so as not to affect the authenticity of the experimental results due to other factors. According to relevant test requirements in the asset safety management industry, the comparison index of this experiment test is the number of vulnerabilities detected by the two asset safety vulnerability detection systems, detection time, false alarm rate and missing alarm rate. By comparing the corresponding experimental data of the above four indexes, the final experimental conclusion is obtained after analysis and processing.

3.2. Test preparation

This system test uses the software source code of 9 open source projects as the test set to test the running performance of the two systems. The specific parameters of the test set are shown in the table below.

| Item number | The project name | Number of project files | Lines of code |
|-------------|------------------|-------------------------|---------------|
| 1           | amsn-0.97        | 76                      | 15130         |
| 2           | avisynth         | 82                      | 48915         |
| 3           | dgvideo          | 67                      | 23600         |
| 4           | libprojectM-1.2.0| 38                      | 17342         |
| 5           | optiping-0.6     | 73                      | 37781         |
| 6           | Playa            | 42                      | 18964         |
| 7           | reacTIVisio-1.4pre1 | 58                      | 21685         |
| 8           | SpGateWay        | 21                      | 5697          |
| 9           | sphinxbase-0.3   | 44                      | 19352         |
The hardware experiment platform equipped with two asset security vulnerability detection systems has a CPU dominated by 3.75ghz Intel Core i7, a running memory of 16G, and a supporting operating system of Windows 8.1. Before the experiment, the two systems were run separately to ensure that the system could normally detect vulnerabilities. Two groups of systems were used to detect the vulnerability of the test items at the same time, and the experimental data corresponding to each test index was obtained. Compare and analyze the experimental data and compare the advantages and disadvantages of the two systems in this test.

3.3. Test results
The comparison results of the number of vulnerability detection and the time spent in vulnerability detection for the 9 test items tested by the two systems are shown in the following table.

| Item number | This paper's system | Traditional system |
|-------------|---------------------|--------------------|
|             | Detection time /s   | Number of files processed successfully | Number of vulnerability detection/true value | Detection time /s | Number of files processed successfully | Number of vulnerability detection/true value |
| 1           | 104.5               | 76                  | 7/7                       | 102.1          | 72                  | 5/7                       |
| 2           | 261.8               | 82                  | 4/4                       | 247.1          | 78                  | 4/4                       |
| 3           | 103.4               | 67                  | 1/1                       | 98.6           | 64                  | 0/1                       |
| 4           | 98.6                | 38                  | 3/3                       | 94.5           | 36                  | 0/3                       |
| 5           | 67.2                | 73                  | 16/16                     | 63.2           | 73                  | 12/16                     |
| 6           | 68.9                | 42                  | 5/5                       | 65.7           | 42                  | 2/5                       |
| 7           | 54.2                | 58                  | 3/3                       | 53.1           | 55                  | 0/3                       |
| 8           | 33.7                | 21                  | 7/7                       | 32.4           | 20                  | 1/7                       |
| 9           | 36.5                | 44                  | 12/12                     | 34.8           | 40                  | 7/12                      |

It can be seen from the above table that the detection time of the security vulnerability detection system in this paper is slightly longer than that of the traditional system. The system in this paper can fully process all files in the project set, although the detection time is a little longer. On the contrary, the traditional system fails to fully process the files in the test item set although the detection time is short. In addition, the system can fully detect all security vulnerabilities in the project set, while the traditional system cannot. The above data indicate that the system in this paper improves the detection accuracy of security vulnerabilities by sacrificing the detection rate.

When the two systems conduct vulnerability detection on test items, the false positive rate and missing negative rate of asset security vulnerability detected by the system are shown in the figure below. The data information in the figure is analyzed and the final experimental conclusion is drawn. Among them, the missing rate is the ratio of the number of missed security vulnerabilities to the total security vulnerabilities. The false positive rate is the ratio of the number of false positive security vulnerabilities to the number of all detected vulnerabilities.
Figure 1. Comparison of false alarm rate and missing alarm rate for system detection of vulnerability

The analysis of the relationship between the graphs in the figure above shows that the false alarm rate and missing rate of the system in this paper are far lower than the traditional system. The average false alarm rate of the system in this paper is 12.3%, while that of the traditional system is 53.1%. The system in this paper can effectively reduce the false alarm rate of the system. After analysis, the reason why the failure rate of traditional system is high is that the security vulnerability with minor problems is not identified when the system rules are made. The failure rate of the system in this paper is all lower than 15%, but the reason of missing report is that the vulnerability database of the system designed in this paper does not contain all the vulnerabilities. The above data show that the system can effectively reduce the false alarm rate and missing alarm rate of security vulnerability detection.

To sum up, the intelligent detection system of asset security vulnerability hidden danger designed in this paper improves the accuracy of vulnerability detection by sacrificing the time of vulnerability detection, which is more suitable for practical use than the traditional vulnerability detection system.

4. CONCLUSION
The detection of asset security vulnerabilities is very important to ensure the stable operation of enterprises. Aiming at the problems existing in traditional systems, this paper designs an intelligent detection system for the hidden dangers of asset security vulnerabilities in a multiple and heterogeneous web network. Through the comparison experiment with the traditional system, it is verified that the system designed in this paper can improve the detection accuracy of security vulnerabilities and hidden dangers, and the effect is better when used in practice. Since the design system in this paper sacrifices the detection rate to improve the detection accuracy, it is necessary to improve the detection rate on the premise of ensuring the detection accuracy in future studies.

ACKNOWLEDGMENT
The study was supported by “Science and technology project of State Grid Corporation of China, “Intelligent detection system of asset security vulnerability hidden danger under multiple and heterogeneous Web network” (Grant No. SGTJDK00DWJS1900105)”.

REFERENCE
[1] Wang, P., Li, X.F., Han, J.F.(2019) Analysis of Vulnerability Scanning in Power Information Network System.Inner Mongolia Electric Power,37,2:33-35.
[2] Qi, L.J.(2019) Mobile Network Hybrid Security Vulnerability Detection Simulation under Static Defense.Computer Simulation,36,4:282-285.
[3] Huang, K., Yang, X.D., Xu, K., et al.(2019) Network Security Situation Analysis of Large Power Grid Enterprises Based on Multi-Dimensional Heterogeneous Model.Electric Power Information and Communication Technology,17,1:72-77.
[4] Zhao, Y.H., Wu, D.Y.(2019) Analysis and Design of An XSS Vulnerability Detection System.Software Guide,18,3:162-167.
[5] Zhang, Z.H.(2018) Real Time Detection of Network Security Vulnerability Based on Penetration Test.Science Technology and Engineering,18,20:297-302.
[6] Zhao, J., Wang R., Li, S.Q.(2018) Research on Smart Home Vulnerability Mining Technology Based on Taint Analysis.Netinfo Security,6:36-44.
[7] Luo, J., Li, L., Meng, K.(2018) An architecture for network security operation with penetration.Intelligent Computer and Applications,8,1:123-126+130.