Research on Web Application Vulnerability Scanning Technology Based on Active and Passive Combination

Yonggang Li *, Min Han, Aiyi Cao, Shanmin Pan

State Grid Information and Communication Industry Group Co., Ltd, Beijing 102211, China.

*Corresponding author Email: Liyonggang08@163.com

Keywords: Power web application system, Web Crawler, Passive scanning, Vulnerability detection.

Abstract: With the construction of the ubiquitous power Internet of Things in our country, web applications have gradually replaced client applications in the power industry, and security incidents in which attackers use web application vulnerabilities to conduct malicious attacks are increasing. In order to ensure the security of power web applications, it is increasingly important to find and repair web application vulnerabilities in a timely manner. This paper proposes a combination of active and passive power web application vulnerability scanning technology. It combines active scanning technology and passive scanning technology based on network intelligent crawlers to quickly and comprehensively obtain vulnerability information and improve the efficiency and coverage of power web application system vulnerability detection.

1. Introduction

With the construction of a new energy interconnected power grid with the characteristics of openness, interaction and extensive interconnection, and the application of the new information and communication technology of "internet plus", a large number of electric power web application systems have been deployed in the power grid in every link of transmission, transformation, distribution, utilization and dispatching, and web application has gradually become the main target of attackers. Due to the limitations of development technology, web management, open source web framework and other factors, power web applications inevitably have application vulnerabilities such as SQL injection, XSS, file inclusion, information leakage and so on. At present, the power web application system has a low degree of automation in safety testing, which mainly depends on the manual testing or safety testing services of various safety personnel, and it is difficult to meet the requirements of safety hazard investigation and safety protection supervision and inspection of mass power web application systems. There is an urgent need to study the high-performance vulnerability scanning technology for power web applications, so as to improve the vulnerability detection efficiency and coverage rate of power web applications.

Web application vulnerability scanner usually obtains web assets such as URLs through web crawlers, analyzes the information of web assets, finds out all relevant files and input points of web applications, imitates the operation of attackers, constructs data packets for detecting vulnerabilities, and sends requests to web application servers. By analyzing the response information of the server, we can judge whether there is a specific vulnerability, and then find out the security risks in the web application. On the premise that vulnerability test library and vulnerability identification rules are consistent, the effect of vulnerability detection mainly depends on the scanning effect of web assets.

According to the characteristics of large-scale power grid information network, large number of network nodes, complex structure and fast update speed, this paper proposes a set of high-performance web application vulnerability intelligent scanning technology based on the combination of active scanning and passive scanning of network intelligent crawler, which can improve the asset identification ability and vulnerability detection efficiency and coverage rate of power web application system.

  DOI: 10.25236/isaiee.2020.015

2. Active reptile

In essence, web crawler is a kind of network information collection tool, which was originally used in search engines to crawl information. There are three crawling strategies for web crawlers, namely, Depth First, Breadth First and Best First. Web crawlers are mainly divided into General Crawler, Focused Crawler, and Deep Crawler according to system structure, crawling strategy and implementation technology.

2.1 General Crawler

General Crawler uses Breadth First, starts from a given page or pages, completes the initial page crawling, and extracts new URLs. The new URLs are put into the queue in no particular order to continue crawling, and extend to the whole Web until the preset stop condition is met [2]. The strategy adopted by General Crawler doesn't analyze the URL, which leads to many useless webpages crawled, and reduces the efficiency of web crawlers. The Focused Crawler crawls the web page and extracts the new URL contained in the web page. It will not be put into the queue immediately, but will analyze the links through some webpage analysis algorithm, filter out the webpages irrelevant to the preset topic, and select the webpages with high relevance to the preset topic to continue crawling. The Focused Crawler uses Best First[3].

2.2 Focused Crawler

The workflow of the Focused Crawler is complicated, and a good webpage analysis algorithm can effectively filter links irrelevant to the topic, and then keep more useful links and put them into the URL queue waiting to be crawled. Then, through a certain search strategy, the URL of the webpage to be crawled next is selected from the queue, and the above process is repeated until a certain condition of the system is reached. In addition, in order to query and retrieve, all crawled web pages will be stored in the system, analyzed, filtered and indexed. The crawler will give feedback and guidance to the future crawling process according to the analysis results obtained in this process.

2.3 Deep Crawler

Deep Crawler is a crawler specially designed for deep web pages. In addition to the surface pages obtained statically, there are also a large number of deep pages hidden on the network, which need to be obtained by extracting forms. Compared with other crawlers, deep crawler adds a key form processing module. It includes the structure of form analysis, processing and response, and completes the task of automatically extracting, analyzing and submitting the form.

2.4 Intelligent crawler

At present, there are many kinds of vulnerability scanner crawlers in the market, but crawler scripts are usually faced with the limitations of scanning range, outdated rule base and slow update, which requires a lot of manpower to test and repeat costs. Moreover, this kind of scanning crawler does not have the ability of complex attack dimension decision-making, only carries out compliance regular matching, lacks security test cases, and cannot achieve the purpose of crawling vulnerability characteristics in multiple dimensions. With the iterative update of WEB front-end framework, the complexity of using traditional regular crawler technology is further enhanced.

In response to this situation, we study a more intelligent web crawler vulnerability detection method based on an interfaceless browser. Compared with browser crawlers, web crawlers of interfaceless browsers reduce resource consumption and crawl faster. Through intelligent crawler technology, task scheduling and event management can be performed. First, the target can be interface preloaded, network idle state waiting, website content hijacking, page jump or shut down interception. Secondly, perform function hijacking of normal website requests, hijack web page popups, newly opened pages, timeout blocking waiting, etc., and then by enabling request interception filtering processing, hijacking the native class XMLHttpRequest, realizing the monitoring status of events and capturing AJAX request information, Obtain new binding event changes, etc., obtain node attribute change information after the event is triggered, and screen the first

batch of results. For events that need to be triggered interactively, traverse elements, traverse events, and traverse forms respectively. After loading the page, the rendering engine finishes rendering DOM and CSSOM. All node elements and events of the web page are registered ready, triggering the bound event information on the nodes, such as web page news scrolling paging, pop-up user selection items, data refreshed regularly in the background, etc., and performing the second batch of result screening. Finally, fill in the form, trigger the mouse click event, and get the data of GET type and post type, and then complete the screening of the third batch. After all the three steps are completed, the screening results are duplicated, and the new output results are brought into the whole subsequent process to further expand the results and expand the attack surface. The workflow of intelligent crawler is shown in Figure 1.

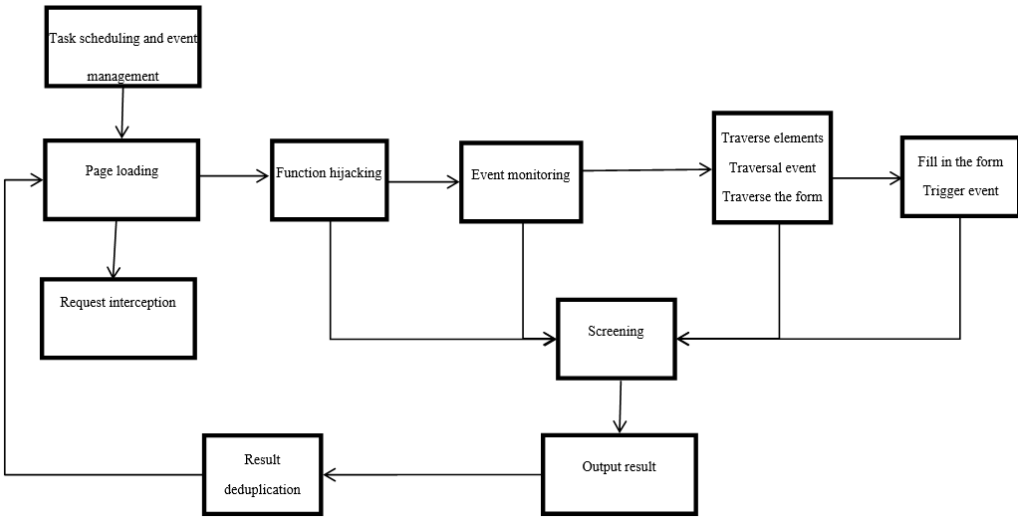

Figure 1 workflow of intelligent crawler

3. Passive agent

There may be scanning blind spots in the scanning process of web crawler, and some important services can only be accessed after authorized users log in correctly, so web crawler can't find out completely the information interaction unit of web application after user authentication. In addition, it is difficult for web crawlers to obtain independent pages, API interfaces and other information. On the other hand, it takes time for web crawlers to crawl information. Because of the large scale, large number of network nodes and complex structure of power grid information network, it is not ideal to collect assets only through web crawler. Passive scanning can be combined to improve the efficiency of asset collection and target coverage. Passive scanning does not initiate a request actively, but only obtains the content of the request for analysis. There are three common data sources for passive scanning: traffic mirror, web log and network proxy.

3.1 Passive scanning based on traffic

Passive scanning based on traffic can obtain traffic from switches and other devices in real time, and comprehensively analyze the protocol types and traffic contents of the traffic in each layer of the network structure by using analysis algorithms [4]. In the face of various new power web application systems, automatic recognition algorithm can be used to identify and analyze protocols to meet the requirements of high accuracy and high performance of protocol identification. Passive scanning based on traffic is convenient and quick to obtain traffic, which can scan a large number of web assets, but cannot parse encrypted traffic such as HTTPS. In practical application, the image location after unified decryption can be selected according to the actual network conditions for traffic analysis.

3.2 Passive scanning based on log

Passive scanning based on log can be combined with the enterprise's own web log system. By analyzing the known log rules, the web asset information such as URL and application interface in the log can be easily and quickly analyzed. It can be used to find interfaces that are difficult for web crawlers to find, and log parsing supports https encrypted traffic. However, due to the uncontrollable access log, the scanning target may be incomplete, which can be used as a supplement to passive scanning based on traffic.

3.3 Passive agent-based scanning

Agent-based passive scanning can be combined with manual detection and analysis, scanning deterministic targets, flexibly controlling scanning contents and improving the efficiency of manual detection and analysis. On the other hand, proxy-based passive scanning can receive the data submitted for form verification, and accurately scan the web pages that need to submit form verification, thus avoiding scanning blind spots caused by web crawlers repeatedly submitting verification or failing to pass verification.

4. Design of web application vulnerability scanning system based on active and passive combination

In terms of web application vulnerability scanning, many well-known commercial tools have been developed at home and abroad, such as Acunetix Web Vulnerablitity Scanner (AWVS), IBM Rational AppScan, NSFOCUS Aurora, A&H MatriXay WebScan, etc. [5]. Among these commercial tools, web asset information collection mainly adopts the active scanning method of web crawlers, which has low scanning efficiency and scan blind spots. This paper designs a web application vulnerability scanning system based on the combination of active and passive. The following will explain the design concept of the system through 4 modules, as shown in Figure 2.

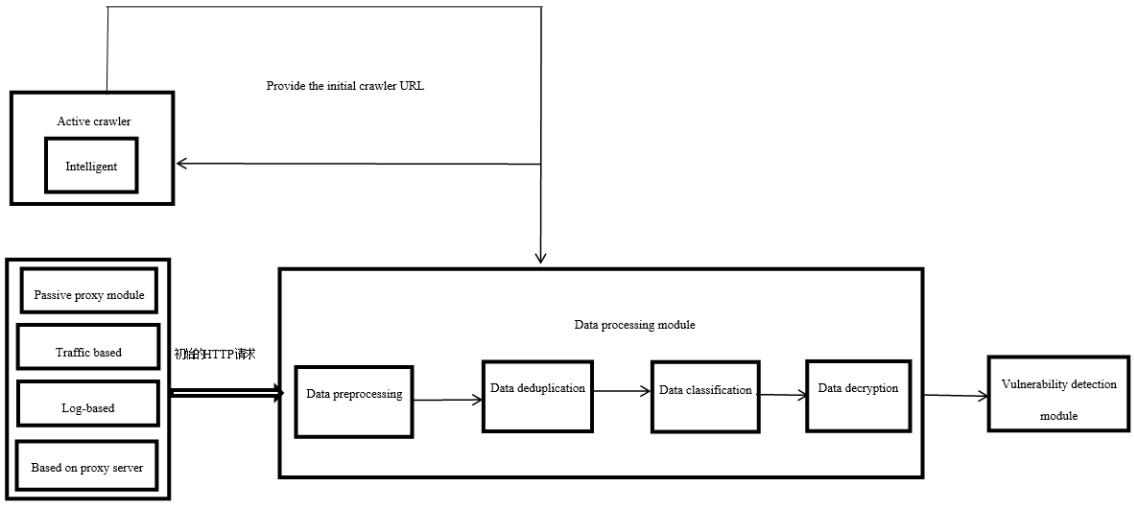

Figure 2 Design flow of Web application vulnerability scanning system

Passive agent and active crawler module are mainly responsible for collecting relevant information of original Web assets, and adopt passive agent as the main mode and active crawler as the auxiliary mode. Among them, the web crawler adopts intelligent crawler without interface browser to improve crawling efficiency. Passive proxy adopts traffic-based, log-based and agent-based methods to solve the problems of web framework and interface which can't be supported by active crawler, scanning blind spot caused by form verification and so on. Active crawler and passive agent complement each other, which makes the information collection of web assets faster, more comprehensive and less interference to business systems, thus improving the effect of vulnerability detection. The data processing module preprocesses, deduplicates, classifies and decrypts the data obtained by passive agents and active crawlers, and the vulnerability detection module precisely attacks different targets according to the data provided by the front data processing module, which is divided into two ways: vulnerability rule base based and POC based.

4.1 Passive agent module

Testers take the initiative to click on the Web application or click on the Web application by machine simulation, simulating normal users to access the foreground function of the web application. HTTP traffic generated by access first passes through the proxy server, and then is sent to the Web server by the proxy server. The proxy server outputs the processable HTTP requests to the data processing module. Passive agents have some situations to consider, as follows:

1) The proxy server has a functional interface for inputting traffic and logs at the same time, and the tester can actively input the Web log or device traffic existing in the Web system to the functional interface. The passive proxy module sends the processed HTTP request that has been automatically identified and decrypted to the data processing module.

2) In order to prevent traffic from being hijacked, some website designers use SSL to encrypt traffic, so the proxy server needs to generate an SSL certificate in advance and install it in the tester's browser in advance. During the test, HTTPS traffic can also be decrypted into data identifiable by the vulnerability scanning system because the certificate is installed in advance.

3) Some website architectures not only use HTTP protocol, but also use websocket to transmit data. Therefore, the proxy server adds websocket traffic proxy to complete the general conversion of wesocket data packets and HTTP data packets.

4) Cross-platform, support to adapt to mainstream operating systems, including windows, mac and linux.

5) Increase the blacklist and whitelist mechanism. The blacklist prevents unintended traffic from passing through the proxy server to attack unauthorized websites, such as government websites, school websites, and hospital websites. The whitelist is more targeted to control website access.

6) Set proxy thresholds and monitor traffic queues in real time. When a certain threshold is exceeded, the proxy server will reduce the capacity of the traffic queue to prevent high concurrent HTTP requests from causing excessive CPU load and redirecting DOS attacks on the WEB server.

Fine-grained target configuration. For example, for the same website www.test.com, there are two paths, one is user and the other is item, corresponding to user functions and product functions respectively. If the tester only wants to test the vulnerabilities of the product function and ignore the user function, a more fine-grained target configuration is required. The vulnerability scanning system designed in this paper can set a path whitelist or blacklist at the passive proxy module. For the above-mentioned situations, only need to configure the whitelist as path=/user.

4.2 Active crawler module

The active crawler module starts when the data processing module outputs the URL result, the URL result generated by the data processing module is sent to the pre-queue of the active crawler module, and the active crawler module starts to work. The active crawler module analyzes by the intelligent crawler, grabs new URL data, and sends it to the data processing module for further processing. There are some issues that need to be considered in the active crawler module:

1) When the network environment is relatively poor, the efficiency of active crawlers will be greatly reduced, and network congestion will further affect the operating efficiency of the vulnerability detection module. Therefore, when running the active crawler module, the system pre-tests the average return time of network requests, and then actively reduces the running speed of the active crawler module. Within a certain range, it can effectively improve the accuracy of crawlers and reduce the impact on the vulnerability detection module.

2) When the web application is equipped with a network firewall, high-intensity traffic access may cause the request ip to be blocked by the firewall and the test may be forced to be suspended. Therefore, when the active crawler module is running, the system will also detect whether there is a WAF page in the returned page or whether there is a mainstream WAF mark in the HTTP request, and set the running speed of the active crawler according to the returned situation.

4.3 Data processing module

The data obtained by the system through the above are all unprocessed HTTP requests. If it is necessary to perform vulnerability analysis on HTTP requests, the data still needs to be preprocessed, deduplicated, classified, and decrypted, and then submitted to the vulnerability detection module for vulnerability analysis, vulnerability identification, and vulnerability verification. The data processing module is divided into 4 functional points, namely preprocessing, deduplication, classification and decryption.

1) Data preprocessing: There are many reasons for web application vulnerabilities. According to the structure of HTTP data packets, any field may have vulnerabilities. Therefore, in the data preprocessing stage, the complete HTTP request packet needs to be split to obtain the field values of different field names, including HOST, PATH, User-Agent, Accept, Referer, Content-Type, Cookie. Parameters and field information customized by different web applications. After splitting, it is stored in a dictionary.

2) Data deduplication: Whether it is a passive proxy or an active crawler, it will inevitably generate a large number of repeated HTTP requests. Therefore, the preprocessed data needs to be deduplicated. According to the preprocessed array, take out the DOMAIN, PORT, PATH and parameter fields to form a structure such as:

"Http(s)://"+domain+‘:’+port+’/’+path+’?’+parameter

The obtained complete structure is hashed. If the same hash value appears, the two HTTP request structures are considered to be the same, and only one of the two HTTP requests is selected. This greatly reduces the functional loss of the system in the vulnerability detection part and improves efficiency.

3) Data classification: According to the results of preprocessing and deduplication, the data is classified. Classification according to different functions, including the classification of 4 dimensions according to the file suffix in the PATH field. The first classification is based on the file suffix name. For example, some unusable static files and pictures can be classified, such as js files, css files, picture files, etc., and some text files or other files that may contain sensitive information can be directly output to the sensitive file leakage part of the vulnerability detection module. The second classification is based on the value of Content-Type in the HTTP request. Different Content-Types have different ways of exploiting vulnerabilities. The third category is classified according to the HTTP request method, including GET, POST, HEAD, PUT, DELETE, TRACE, CONNECT, OPTIONS. Different request methods have different ways of exploiting vulnerabilities. The fourth classification is based on the similarity of PATH. This classification method is more like a method of classification and deduplication. For example, "http://www.test.com/1" and "http://www.test.com/2" can actually be regarded as the same type of request. The similarity of the regular paired PATH and the similarity of the returned packets are used to determine whether the requests are of the same type.

4) Data decryption: In order to facilitate data transmission, some website designers add custom data encoding methods in the back of the website to encode parameter content, such as base64 encoding or hexadecimal encoding. These codes may reduce the detection efficiency of subsequent vulnerability detection modules. Therefore, common coding methods can be identified and decoded in advance during data processing, which effectively improves the efficiency of subsequent vulnerability detection.

The specific flow chart is shown in Figure 3.

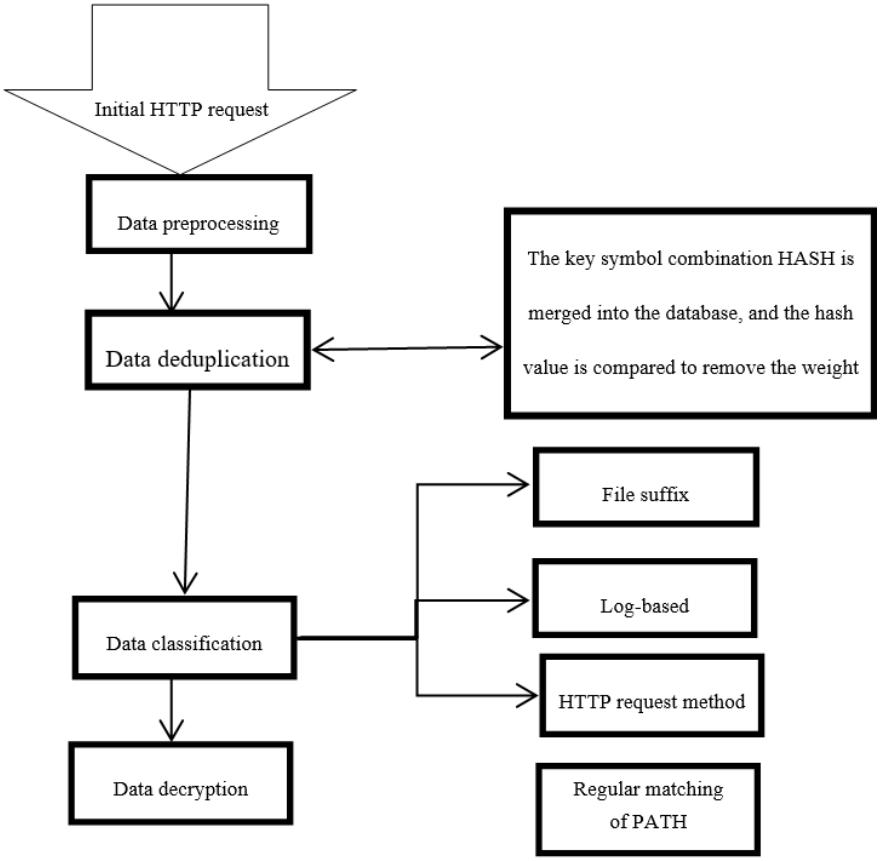

Figure 3 Data processing module flow

4.4 Vulnerability detection module

1) Based on the vulnerability rule library: The first detection method of the vulnerability detection module is based on the "key-value" matching pair generated by the data processing module, and is aimed at general-purpose vulnerability detection. According to the generation conditions and principles of different types of vulnerabilities, different vulnerability rule bases are generated, and each possible vulnerability key value is generated. Traverse the vulnerability rule base and assign the value as value. When the returned package has a vulnerability flag, it is determined that the vulnerability exists. For example: there are key-value matching pairs of {PATH:"/test/1"} in the output dictionary of data processing, the possible vulnerabilities in PATH are the leakage of sensitive files, and the system-defined vulnerability rule base contains various possible existences. Sensitive files. The specific detection steps are: 1. Take out a vulnerability rule from the vulnerability rule library, for example: /admin.php. 2. Assign value to the vulnerability rule, for example: {PATH:"/admin.php"}. 3. Combine all key-values of the same HTTP request and send the request to the web application server. 4. If the returned status code is 200 and the length of the returned data packet is greater than a certain threshold, the PATH is considered to exist. 5. If the loophole rule traversal is completed, the detection is ended, otherwise, repeat 1-4.

Based on POC: The second detection method of the vulnerability detection module is based on the POC (Proof of Concept) confirmatory test operation and is aimed at non-universal vulnerability detection. This type of vulnerability exists in a specific version of a specific application. Targeted test steps and verification methods require predetermined test rules, execution of test rules, and verification of return requests. For example: when verifying the CVE-2018-7600 vulnerability, you need to send a POST request to the server first, and cache the specific vulnerability rules to the WEB server. At the same time, it is necessary to obtain the ID value of a specific field according to the returned packet, and then send a POST request, and carry the ID value obtained in the previous step and the vulnerability verification rule in the body value of the request, and judge whether the vulnerability verification rule is executed in the returned request Does this vulnerability exist?

5. Conclusions

In view of the characteristics of large scale, large number of network nodes, complex structure and fast update speed of power grid information network, this paper analyzes the problems existing in vulnerability scanning of web applications, and puts forward an intelligent scanning technology of vulnerability hidden danger of web application system based on the combination of active scanning technology and passive scanning technology of network intelligent crawler, so as to realize fast, small interference and high coverage rate of web asset information collection. In the future, we will further study the vulnerability point identification and vulnerability verification based on intelligent algorithm to further improve the accuracy of vulnerability identification.

Acknowledgments

This paper is supported by the Science and Technology Project of the Headquarters of State Grid Co., Ltd. (Research on Intelligent Detection and Verification Technology of Power Information Network Security Risks +SGTJDK00DWJS1900105)

References

[1] Chen Jingjie. Design and implementation of a high-performance Web application vulnerability scanning system [D]. Beijing University of Posts and Telecommunications, 2019.

[2] Min Wu, Junliang Lai. The Research and Implementation of Parallel Web Crawler in Cluster[J]. Computational and Information Sciences (ICCIS), 2010 International Conference on, Dec. 2010: 704-708.

[3] M. Yuvarani, N. Ch. SNIyengar, A. Kannan. LSCrawler:A Framework for an Enhanced Focused Web Crawler:Based on Link Semantics[J]. Web Intelligence, 2006. WI2006. IEEE/WIC/ACM International Conference, Dec. 2006:794-800.

[4] Zhou Tao. Data mining technology in network security [M]. Beijing: Tsinghua University Press, 2017: 162-167.

[5] Zhang Nan. Research on Web Application Security Vulnerability Scanning Technology [D]. Zhejiang University, 2015.