Hindawi
Scientific Programming
Volume 2022, Article ID 8645969, 11 pages
https://doi.org/10.1155/2022/8645969

Research Article

Design of Web Security Penetration Test System Based on Attack and Defense Game

Bing Song, Li Sun, and Zhihong Qin

Department of Network Security, Henan Police College, Zhengzhou 450000, China

Correspondence should be addressed to Bing Song; songbing@hnp.edu.cn

Received 22 April 2022; Revised 10 May 2022; Accepted 18 May 2022; Published 10 June 2022

Academic Editor: Lianhui Li

Some sensitive data in the network will be leaked due to the loopholes or weaknesses of the web system itself, which will bring potential harm to the society or the public. Aiming at this, this study carries out the design of web security penetration test system. A test scheme comparing single method with an automatic comprehensive test method is designed. Based on this scheme, an automatic penetration test system script used under the terminal operation page is tested and designed. A security evaluation algorithm based on the prediction results of the game between attack and defense is proposed. Through this algorithm, different website systems are evaluated and scored, and the test results are compared through scoring. The automatic penetration test integration system designed and implemented in this study can meet the main objectives of web security and the protection requirements of websites against general, routine, and universal security attacks. The proposed evaluation algorithm is more detailed, accurate, and reference in scoring.

1. Introduction

With the popularization of Internet and the rapid development of web application technology, Internet provides an important basic platform for web applications, on which more and more web applications are set up [1–6]. Common online shopping malls, online banking, and other web applications have greatly changed people's lifestyle [7–10]. They can easily shop or deal with financial problems without leaving home. However, these new technologies not only bring convenience to our work life and even learning but also bring great risks that we have never had before. Due to the maturity of network technology, the threshold of web application attack technology is getting lower and lower. Hackers gradually transfer the attack object from the network server to the web application [11–14]. According to Gartner's survey data, 75% of information security attacks occur on Web applications, not on the network level. At the same time, it is also found that two-thirds of web applications are very vulnerable and vulnerable to attacks. However, it is a pity that many enterprises spend a lot of energy and financial resources on network security and server security and do not pay attention to the security problems of web applications, leaving an opportunity for hackers [15–17].

The main reason for the vulnerability of web applications is that users can submit data arbitrarily, but the server-side does not carry out reasonable verification. From the perspective of the software itself, the main reason is that the cycle of web application development is getting shorter and shorter, and the level of developers is uneven, which leads to the incomplete consideration of security issues in the process of software development. From the perspective of software deployment and configuration management, the staff may be careless, so there are great security risks. A comprehensive penetration test must be conducted on the web application before the attacker launches an attack to ensure the security of the web application and prevent it from happening [16, 17]. Due to the uneven level of web application developers and the shorter development cycle, it is inevitable for web applications to have security vulnerabilities. The traditional way to ensure network security is through firewall and IDS/IPS. It works on the network layer, while the security penetration test process works on HTTP protocol. It can make up for the deficiency of firewall relative static

defense. The two complement each other and jointly ensure the security of web applications.

Web application penetration testing technology is to simulate the attack means and attack methods of hackers, actively detect web applications, transform malicious URLs, send them to the server, and determine whether there are security vulnerabilities in web applications by analyzing the response returned by the server. Through penetration testing, we can understand the vulnerabilities of web applications in advance before they are attacked, classify the level of vulnerabilities, and find corresponding mitigation schemes according to the severity and urgency of vulnerabilities so as to greatly reduce the risk of web applications being attacked.

Due to the unsatisfactory efficiency and accuracy of manual penetration testing, security workers have tried to develop some security penetration testing tools. However, many security penetration testing tools can only test one or several vulnerabilities, and the test effect is not satisfactory. The purpose of this research is to design and develop a web application security penetration testing tool, which can comprehensively and automatically detect some common web application vulnerabilities and can give a more detailed vulnerability detection report.

2. Overall Demand Analysis and Design

In the whole process of the penetration test, it is necessary to formulate a test plan in advance, and various factors will affect the final test conclusion and results. The whole process is divided into three steps: penetration test, design and implementation of automated penetration test system, and safety assessment. Safety assessment is interspersed in the penetration test and automated penetration test. The overall process is shown in Figure 1.

2.1. Penetration Test Requirement Analysis. In the whole process, it is necessary to carry out penetration attack against the test target, and on the premise of not damaging the system, find out the problems existing in the system as much as possible, join other websites for testing, and compare the results, which is more convincing. The overall test is based on Kali Linux system [18, 19], including the use of mainstream test tools and the design of an automatic penetration test system. The specific test contents are shown in Figures 2 and 3.

2.2. Security Assessment Needs Analysis. The overall safety assessment is carried out according to the penetration test results and the calculation results of the assessment algorithm. Security assessment refers to a series of security assessments for websites, systems, and platforms. At present, web security assessment [20, 21] is mainly carried out from internal and external aspects. Internal evaluation adopts black-box test or white-box test. Black-box test refers to the evaluation test without knowing the detailed information of the system, while white-box test refers to the evaluation test with a certain understanding of the information and conditions of the system.

External evaluation refers to the remote evaluation of the server and system initiated by the outside. Testers discover the security problems exposed by the system by simulating the malicious scanning and detection behavior of attackers. Internal evaluation refers to the internal security inspection conducted by testers for the server, code design, and configuration of the system. Compared with external testing, internal evaluation testing can find the problems of the system from a deeper level. The working process of safety assessment is shown in Figure 4.

The overall assessment is carried out by establishing the assessment model and method. At the same time, the requirements of the test objectives for safety assessment are analyzed and understood. The specific aspects should be considered from these aspects as shown in Figure 5.

(1) *Protection requirements*. At present, there are many security attack technologies, and the methods are updated quickly. Therefore, it is required that the cycle of security evaluation must be shortened to ensure the security protection and attack prevention of the whole network.

(2) *Isolation requirements*. Many security attacks are gradually infiltrated from the external network to the internal network. Although many enterprises and units have separated the internal and external networks on physical lines, when it comes to business activities and external information exchange, they will cancel some of the isolation instead of using special machines for access.

(3) *Verification requirements*. Network security is a multifaceted problem, which involves not only attack and protection but also authorization, authority, confidentiality agreement, and other internal problems. Different identity and authority verification for different login users can reduce the risk of being penetrated into the intranet to a certain extent.

(4) *System intrusion detection requirements*. Today's systems and platforms are basically protected by firewalls, but in our continuous research and testing, we also found that although firewalls are stable and can immediately tension access, they are static after all, while network attacks are dynamic, and there are countless methods. Therefore, intrusion detection methods and security evaluation methods must be equipped.

(5) *Vulnerability threat requirements*. Because the website system and platform are artificially coded and designed, errors often occur in the writing and logical specification of the code in the design. Most of them ignore the simplicity and preciseness of the code on the premise of realizing the function,

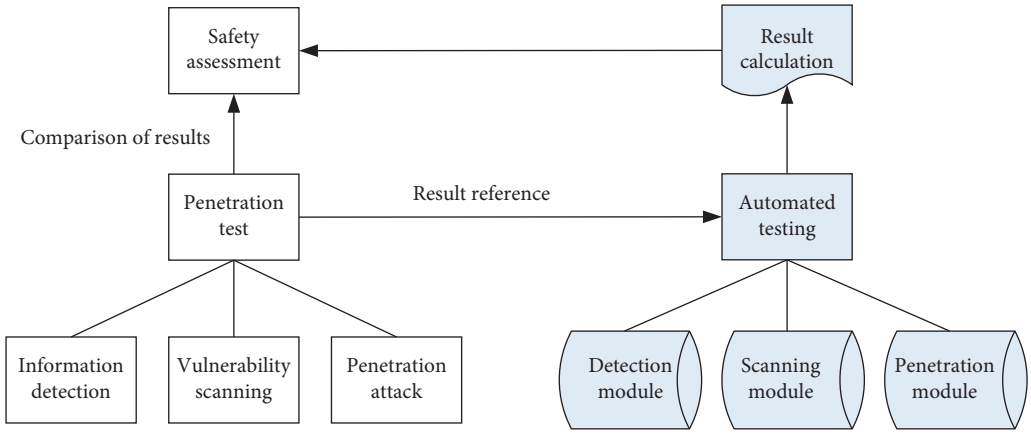

Figure 1: The overall process.

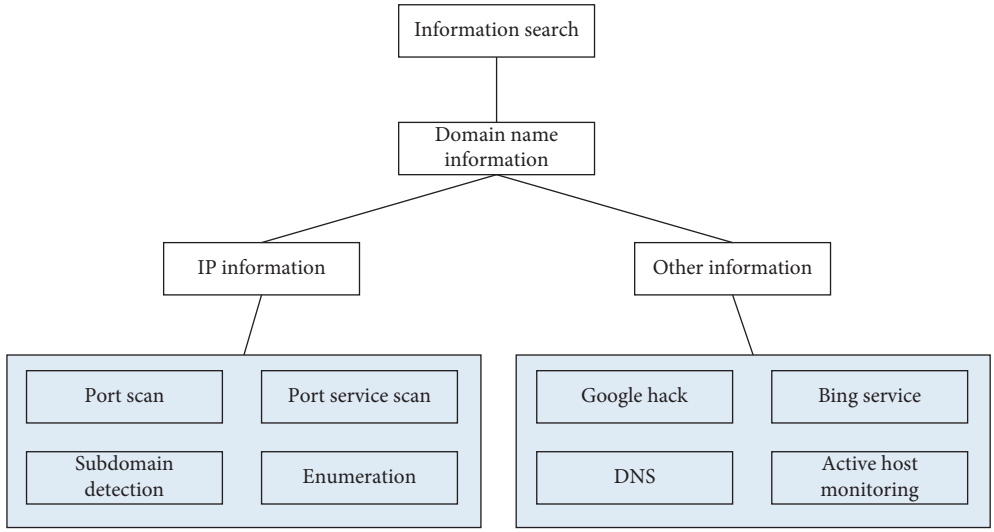

Figure 2: Information search.

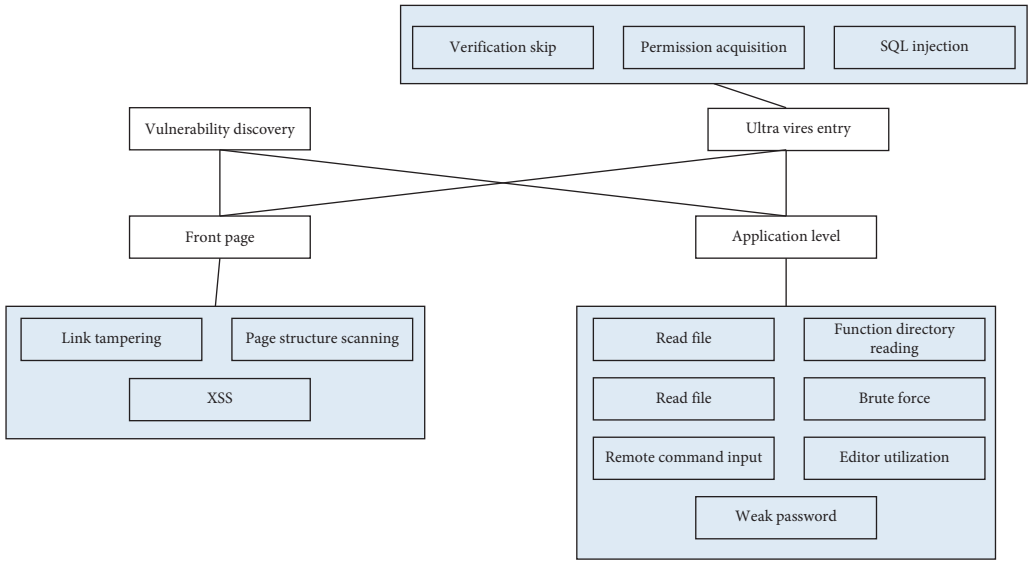

Figure 3: Vulnerability discover.

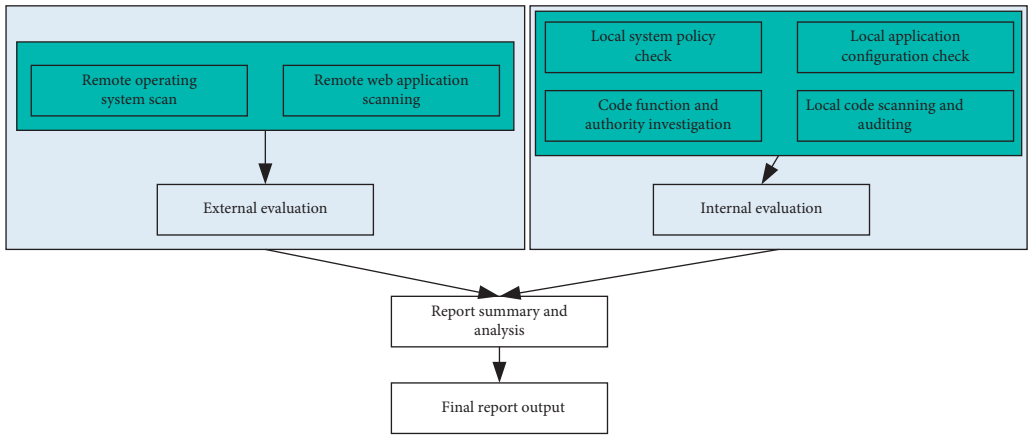

Figure 4: The working process of safety assessment.

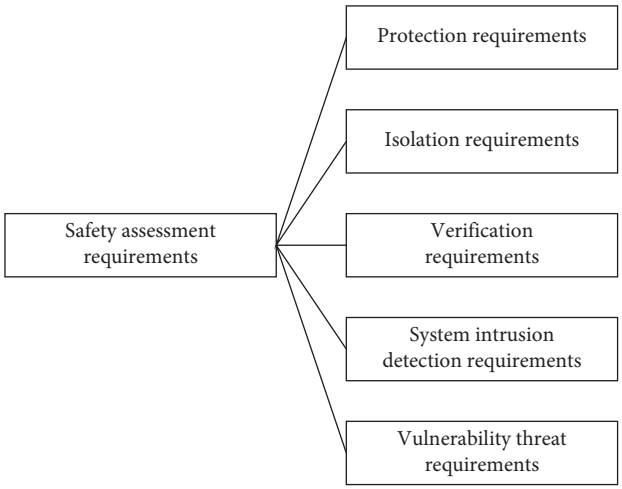

Figure 5: Safety assessment requirements.

resulting in attack vulnerabilities. Irregular vulnerability scanning and security evaluation can effectively deal with this, find and repair the imperfections of code design and system design, and can effectively prevent network attacks.

3. Penetration Test Method Design

In the actual operation, the penetration test [22] will be divided into three parts and stages. Different angles will make the differentiation methods different. For example, it is divided into preparation stage, penetration test stage, and overall comparison and evaluation stage on the theoretical basis, while in the technical operation, it is divided into three stages: detection, attack penetration, and target permission acquisition. The specific implementation of penetration test needs to consider the following factors: the scale of the client, the distribution of the network, and the composition of the system. In the content of the scheme, it is necessary to specify the test purpose and scope, time, place, personnel information, risk avoidance means and methods, overall plan, process, etc.

The overall test includes three parts: information collection and detection, vulnerability scanning, and penetration attack (as shown in Figure 6). In the detection stage, it is mainly to collect information and investigate. First, it is necessary to use the single dimension method to test.

(1) *Detection phase*. We use the main software and tools in the market to test two different websites, compare the results, and then design the detection module in the design of the automatic penetration test integration system.

(2) *Vulnerability scanning phase*. It mainly uses scanning tools to scan the website, obtain vulnerability information, summarize the scanned vulnerabilities, and summarize the advantages and disadvantages of each scanning tool, which can be used as a reference for the design of scanning module in automatic penetration test integration system.

(3) *Infiltration stage*. We use terminal operation and graphical interface operation to penetrate the website, find the security problems existing in the website, analyze the test results in detail, repair the existing security problems and vulnerabilities, and design the penetration attack module of the automatic integrated test system.

(4) *Automated penetration testing phase*. We design and implement the automatic penetration test system and use the system to conduct automatic penetration test on the website. By comparing the test results with the previous single dimension method, we draw some conclusions and the advantages and disadvantages of the two methods so as to provide experience for website security maintenance in the future.

4. Safety Assessment Method Design

Security risks need to be quantified, which is not only conducive to the quantitative calculation of risk value but also allows the client to feel the security of the website and system more intuitively [23, 24]. The flow chart of the safety assessment is shown in Figure 7.

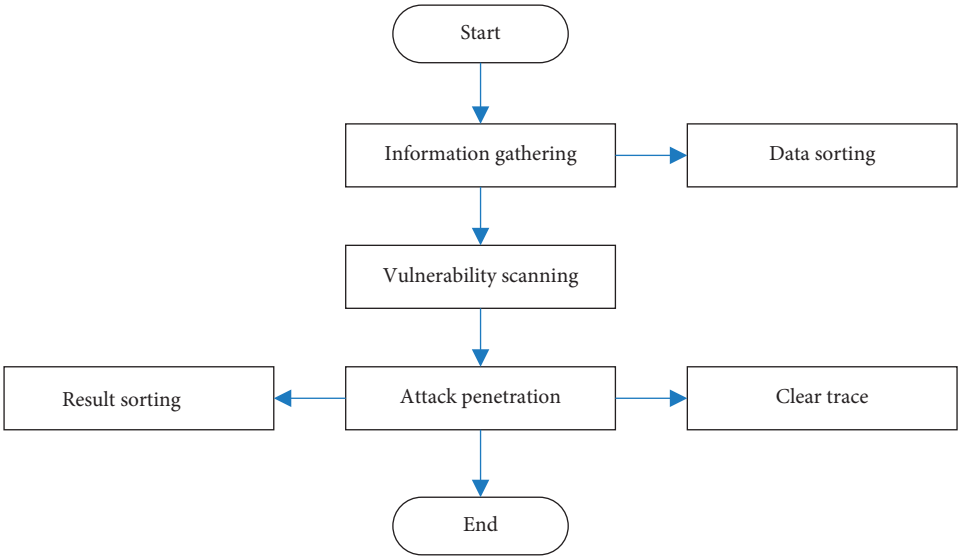

Figure 6: Penetration test flow.

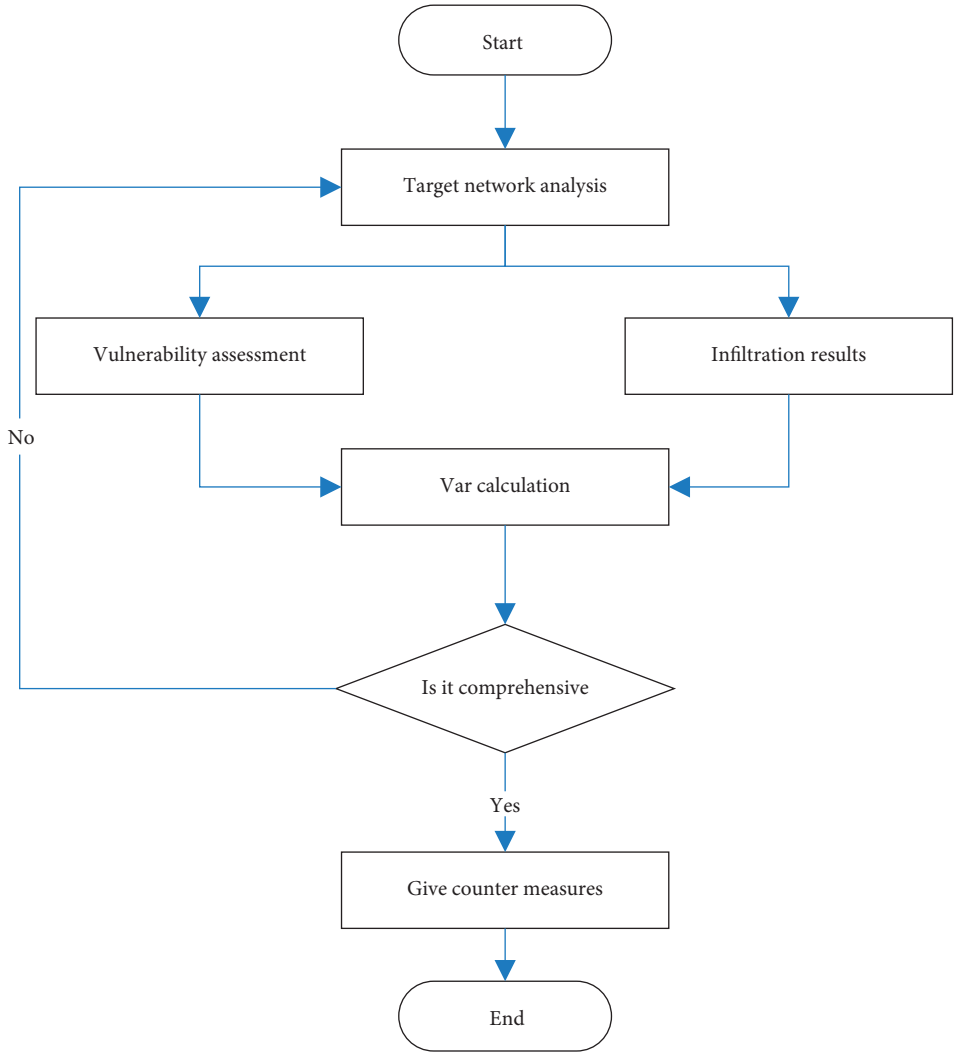

Figure 7: Safety assessment flow.

Before formulating the algorithm of security risk calculation, the selection of variables is very key. Most of the previous evaluation algorithms calculate with one or two variables. The advantage of such calculation is that it is convenient to get the result and will not make mistakes, but the disadvantage is that there are too few reference factors. There are often many indicators that affect the security of a system. When scoring, they need to be taken into account, it is relatively feasible to focus on several main factors and calculate the weight with other factors as a reference. Vulnerability risk quantification is one of the most important factors in the whole security assessment process. The overall security of the system depends on the weakest part. Which risk quantification value of the detected vulnerability is greater will determine the weakness of the whole system. The quantitative value needs to be calculated by the set evaluation algorithm, and the quantitative evaluation of system security can be effectively obtained according to the calculated results.

Most quantitative value algorithms pay more attention to the quantification of vulnerability threats. By quantifying the threats of existing vulnerabilities and detected vulnerabilities, the security factor is evaluated according to the quantitative value, and a vulnerability threat quantitative table calculated by the algorithm is obtained. Through this table, the threat quantitative value of each vulnerability can be seen intuitively so as to use this value as a reference to obtain the website security evaluation coefficient. However, there are still some deficiencies in these evaluation methods. One of them is also the improved part of the algorithm proposed in this paper, that is, it is necessary to consider the test evaluators and website maintainers. As the attacker and defender, they also need a quantitative value for the judgment of system setting and network structure, and these factors cannot be considered after the test.

Many security attacks and risks are sudden, and there is no time to make a temporary response. Countermeasures need to be given in advance, and these countermeasures need to estimate the possible results of the two sides in the process of playing chess in advance. The security evaluation algorithm based on the prediction of attack and defense game results proposed in this paper adds the evaluation of system maintenance personnel on various structural factors of the network and system and the predictive evaluation of the whole website by using the existing knowledge system through the prior understanding of the website system. After the final test, based on the test results, we make an overall evaluation with reference to the quantitative value of vulnerabilities. Combined with the previous evaluation of maintenance personnel, we get the final score through the algorithm. The relevant variables are described below.

Evaluation of website by website maintenance personnel.

Taking each page and structure of the website as variables, a score of X_{i} is obtained, where i ranges from 1 to N, X_{i} ranges from 0 to 10, and the score results are expressed as A. That is

$$A = \sum_{i=1}^{n} X_i. \tag{1}$$

Here, A is the sum of the scores given by the maintenance personnel on each page of the website, and then, we calculate the average to obtain the score B of a single maintenance personnel.

$$B = \frac{A}{n}. \tag{2}$$

The score of each page and structure constitutes a vector.

$$\overrightarrow{Z}_i = (Z_1, Z_2, \ldots, Z_n). \tag{3}$$

Here, n is still the number of website pages and structures, and each page also needs to have a proportion. It also needs to be evaluated by maintenance personnel. It is defined as $V_i V_{\mathrm{i}}$. C represents the whole weight vector, then

$$\overrightarrow{C} = (V_1, V_2, \ldots, V_n). \tag{4}$$

Here, $V_1 + V_2 + \cdots + V_n = 1$.

Therefore, the score of the whole website is the inner product of two vectors. Through the calculation of the inner product, the results can be obtained more accurately. D is defined as the importance score of each module of the website, then

$$\overrightarrow{D_i} = \overrightarrow{Z_i} * C. \tag{5}$$

The first is the comprehensive scoring algorithm of the maintenance personnel for the website, and the next is the evaluation algorithm of the tester. Since the tester needs to simulate the attacker's destruction of the website to the greatest extent, we need to consider the strength of penetration testing and focus on the algorithm for some serious vulnerabilities detected.

Common Vulnerability Scoring System (CVSS) is a standard for vulnerability assessment, which intuitively reflects vulnerability risk by using numbers ranging from 0 to 10. The whole evaluation system of CVSS [25, 26] is composed of basic score, temporary score, and environmental score. The whole evaluation process is to integrate the scores of these three factors. First, calculate the values of each part according to the formula and then get the final score according to the summarized formula. A high score represents a high threat, and a low score represents a low threat. The overall evaluation process is shown in Figure 8.

We combine the accurate score based on the CVSS scoring standard with the importance score of each module of the website to obtain:

$$\alpha = \frac{B + D/n}{2} \times 0.4 + C \times 0.6. \tag{6}$$

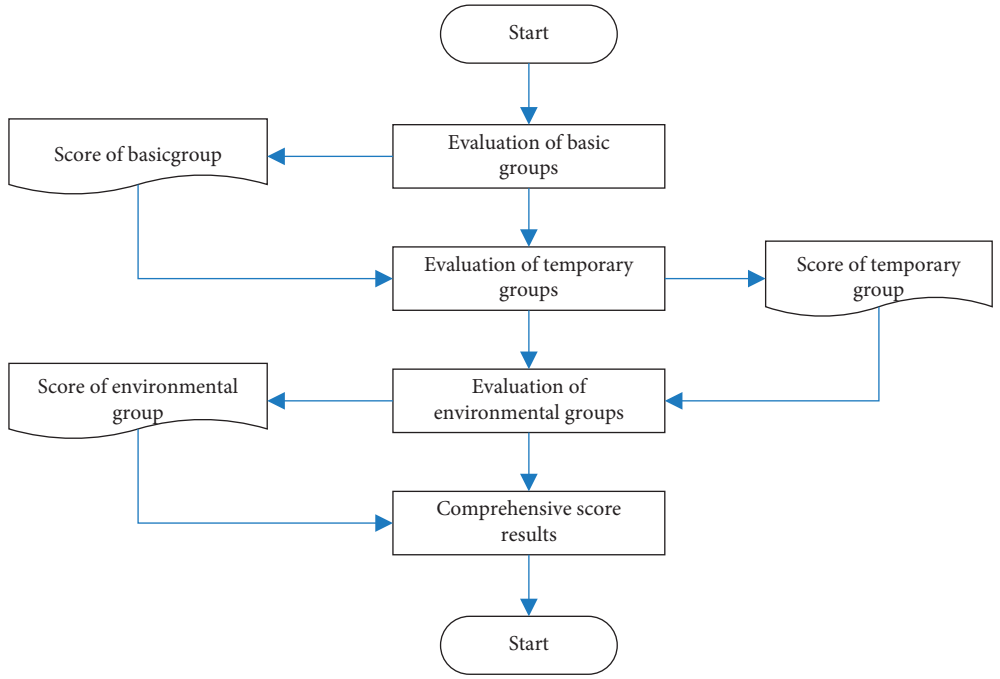

Figure 8: The overall evaluation process of CVSS.

Here, C is the score after CVSS optimization calculation. Since the system maintenance personnel may lack knowledge of system security vulnerabilities and security-related knowledge, they are given a base of 0.4 in weight distribution. Although the security testers are very good at security attacks, they are unable to have a detailed understanding of the overall structure, page blocks, and logic of the system in a short time, so the base is given as 0.6.

5. Design of Web Security Penetration Testing Tool

5.1. Overall Framework. We crawl for the target web application, obtain a large number of URLs and web page content, modify the parameters in the URL or construct abnormal HTTP requests to imitate the operation of malicious users, realize malicious injection, determine whether there are vulnerabilities in the web application by analyzing the response, and generate the corresponding report.

The classic MVC architecture is used to separate the view layer, business logic layer, and data layer and separate the functions of this tool. Using the modular design theory, the user interface module and vulnerability report module are mainly designed in the view layer, and the control engine module, crawl module, injection module, and analysis module are designed in the business logic layer. Due to the huge number of crawls and the need to use multithreading, the thread pool module in the common component is designed. The injection process needs to construct abnormal HTTP requests and analyze the response, so the HTTP proxy module is designed. Because there is some global processing information in the whole process, in order to better solve the migration and reuse of code, the configuration file module needs to be used. In the data layer, it mainly designs URL database, injection database, and analysis database. These independent modules only complete their own functions and do not need to pay too much attention to the functions of other modules. They communicate through some predefined interfaces to transfer parameters, which can improve the reusability of modules, and the modification or addition of new modules of some modules will not affect the normal work of other modules.

The overall frame design is shown in Figure 9.

5.2. User Interface Layer. The user interface layer is the bridge between the security penetration tester and the tested web application. In this module, testers can set the types of attacks that the tested web application wants to test (such as XSS and SQL injection) or test all types and then submit these information to the control engine to perform subsequent tasks.

This module can also generate a vulnerability report form to clearly show all scanning results to testers.

5.3. Logic Control Layer

5.3.1. Design of Web Crawler Module. The main function of the crawler module [27–29] is to crawl the web page, obtain the URL contained in the page through web page analysis, format and filter the obtained URL, and save it to the database. Of course, in the design process of the crawler module, considering the huge number of crawls and the long crawling time of using a single thread, multithreading is introduced, and the thread pool is mainly used here. In order to detect as many vulnerabilities as possible, we need to crawl the web page in sufficient detail. The crawl component we

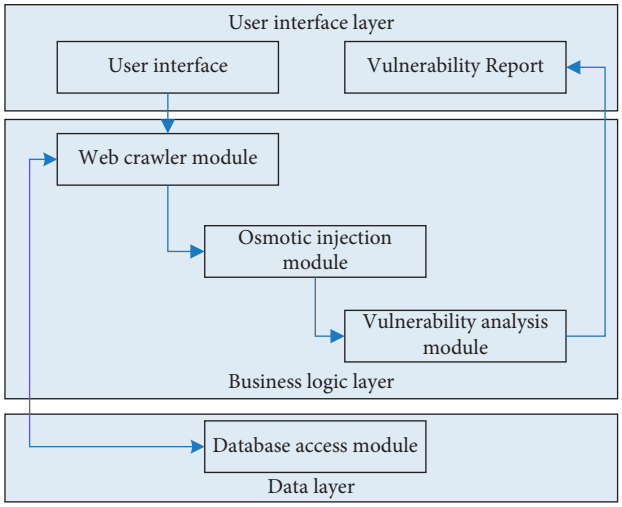

Figure 9: The overall frame design of web security penetration testing tool.

designed adopts the breadth-first strategy. The so-called breadth first is layer crawling, which indexes, processes, and grabs web pages according to the distribution and layout of layers. Because the web pages of many websites contain a large number of files and it is time-consuming to use a single thread, the general crawling components need to be set as multithreaded, which greatly reduces the crawling time and improves the crawling efficiency.

The flow chart of the web crawler module is shown in Figure 10.

5.3.2. Design of Penetration Injection Module. Penetration attack is to simulate the operation of malicious users and send malformed HTTP requests to the server. There are two methods for the HTTP request sent by the web application and the server. One is the GET method, and the other is the POST method. Therefore, the injection point is also divided into two kinds. For GET method, it only needs to modify the obtained URL, send the request to the server, and determine whether there is a vulnerability in the web application through the server-side response. For the POST method, it needs to use HTTP proxy to intercept the post request, modify the parameters, construct abnormal requests, and detect the vulnerabilities on the web application server.

Security penetration injection is mainly to send the modified HTTP request to the target web application. A large number of URLs can be obtained in the crawler module, and these URLs are stored in the database. We need to take out these URLs, find the injection attack point, and call the vulnerability detection plug-in to simulate the attack. The structure diagram is shown in Figure 11.

The penetration attack module mainly designs a control engine and several vulnerability attack plug-ins. The control module is the main thread of the program, controls the underlying vulnerability attack plug-ins, and contacts with the web crawler module. The main thread obtains the

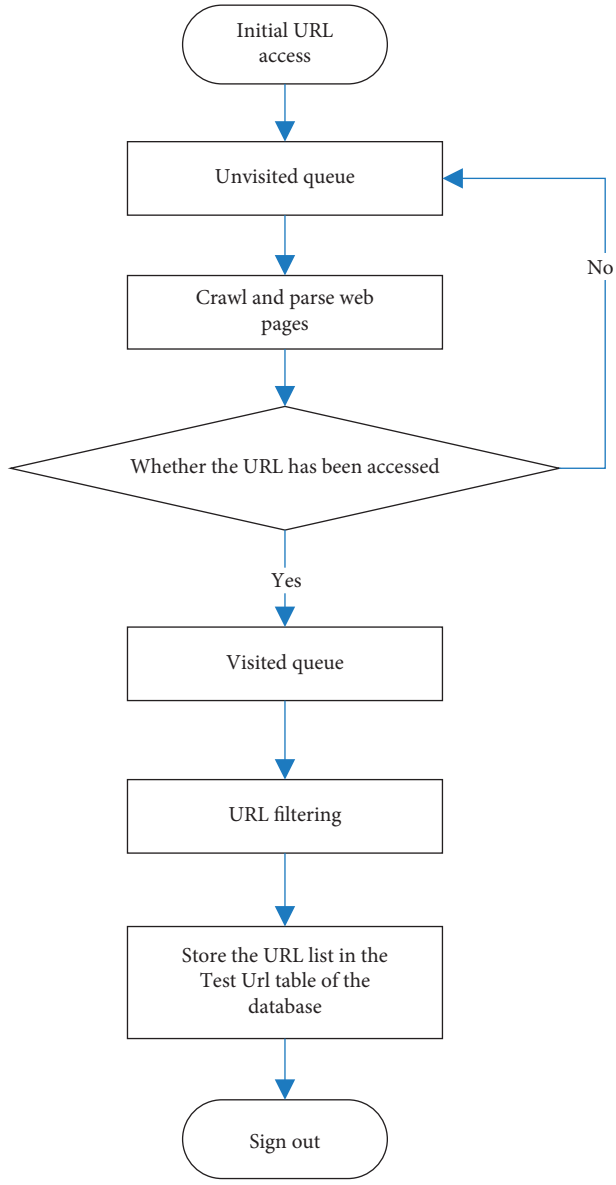

Figure 10: The flow chart of web crawler module.

application data through the web crawler module and gives it to the control module, which is responsible for transmitting the obtained data to each subthread. After receiving the data from the main thread, each subthread waits for the main thread to call the subthread and perform different types of vulnerability injection, respectively. Specifically, the tester can select a URL or all URLs for testing. After receiving the instruction, the control engine starts to call each vulnerability plug-in, construct a malicious attack connection, and complete the penetration injection process.

The design of the control engine is to separate the control logic from the specific execution module and achieve the decoupling effect. In this way, even if a new module is added or the original module is modified in the future, there is no need to modify the overall architecture of the tool, just add a new one in the interface class.

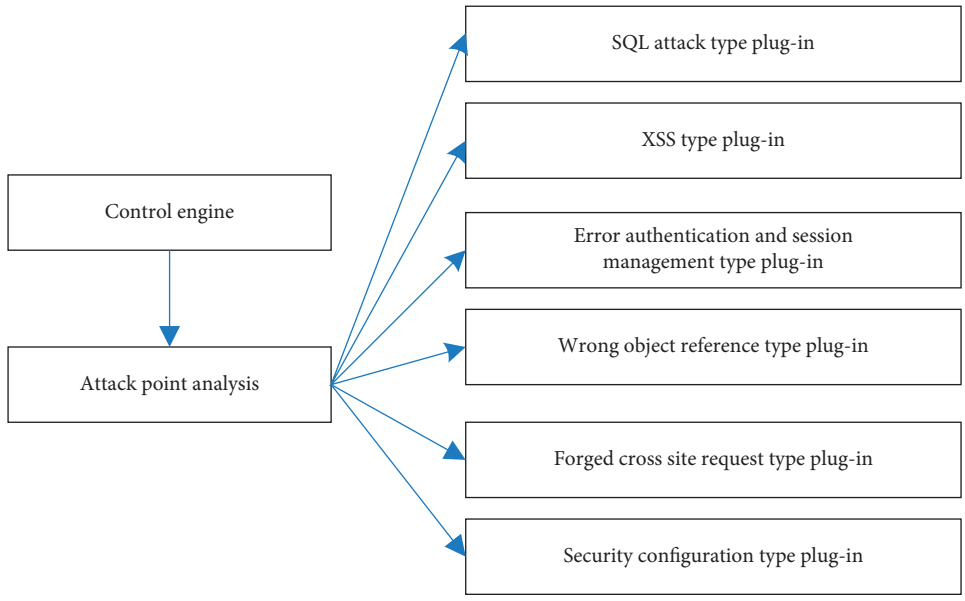

Figure 11: The structure of security penetration injection.

Table 1: Spider URL data table.

Field name	Field type	Is it nonempty	Record content	Remarks
ID	Int	Yes	Primary key	Self-increasing
URL	Varchar (25)		Requested URL	
URL_UNVISITED	Varchar (20)		The URL extracted from the crawled web page	
URL_VISITED	Varchar (20)		URL that has been crawled	

Table 2: Test URL data table.

Field name	Field type	Is it not empty	Record content	Remarks
ID	Int	Yes	Primary key	Self-increasing
URL	Varchar (30)		Requested URL	
DEPTH	Varchar (20)		URL depth	
RESPONSE_CODE	Varchar (20)		HTTP response code	
METHOD	Varchar (10)		HTTP request method	
GET_PARAM	Varchar (256)		GET request parameters	
POST_PARAM	Varchar (256)		POST request parameters	
REQUEST_HEADER	Varchar (20)		HTTP request header	
RESPONSE_HEADER	Varchar (20)		HTTP response header	
RESPONSE_BODY	Varchar (256)		HTTP response body	

5.3.3. Design of Vulnerability Analysis Module. Vulnerability analysis and injection complement each other. After the penetration injection attack is completed, the server will give a response and determine whether there are vulnerabilities in the web application by analyzing the response. Its main principle is to match the obtained response with the preset output. If the matching is successful, it indicates that there are loopholes. If the matching is not successful, it indicates that there are no loopholes.

Since the judgment rules of each vulnerability injection type are written in the corresponding vulnerability injection plug-in, after each vulnerability injection plug-in is called and a malicious link is sent, the returned response should be dynamically matched with the plug-in to determine whether there is such a vulnerability.

5.4. Data Layer. Due to the convenience of the database, the tool in this paper uses the database as the support. When the web crawler runs, it will save some important data to the database, and these data will also provide a solid foundation for the penetration injection module. The specific design of the data table involved is shown in Tables 1 and 2.

6. Conclusions

With the rapid development of network technology and web application technology, web application has penetrated into

every bit of people's life. However, the security problem of the web application has become more and more prominent and the most important technical challenge in this information age.

Through the research on the common vulnerabilities of web applications and the penetration testing technology to detect vulnerabilities, this study first designs and implements the web crawler module. The web crawler adopts the breadth-first crawling strategy. In the process of multithreaded crawling, it obtains all the URLs of the target website through web page parsing, URL formatting, and filtering. Then, the security penetration injection module is designed and implemented, the principle of the penetration injection module is analyzed in detail, the injection points and injection parameters of get type in the request of URL as well as the injection points and injection parameters of post type are analyzed, and the constructed malicious URL is sent to the server by using the automatic injection mechanism. Finally, the analysis module is designed and implemented.

Data Availability

The data set can be accessed upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] Y. Dong, "Application of artificial intelligence software based on semantic web technology in English learning and teaching," *Journal of Internet Technology*, vol. 23, no. 1, pp. 143–152, 2022.

[2] L. Luo, "Web application software engineering technology and process," in *Proceedings of the Ieee Asia-Pacific Conference On Image Processing, Electronics And Computers, Ipec 2021*, pp. 935–937, Dalian, China, Apirl 2021.

[3] L. Li, B. Lei, and C. Mao, "Digital twin in smart manufacturing," *Journal of Industrial Information Integration*, vol. 26, no. 9, Article ID 100289, 2022.

[4] H. Lin, "Application of web 2.0 technology to cooperative learning environment system design of football teaching," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 5132618, 9 pages, 2022.

[5] R. Chen, "The design and application of college english-aided teaching system based on web," *Mobile Information Systems*, vol. 2022, Article ID 3200695, 10 pages, 2022.

[6] L. Li, T. Qu, Y. Liu et al., "Sustainability assessment of intelligent manufacturing supported by digital twin," *IEEE Access*, vol. 8, pp. 174988–175008, 2020.

[7] S. Qi, S. Li, and J. Zhang, "Designing a teaching assistant system for physical education using web technology," *Mobile Information Systems*, vol. 2021, Article ID 2301411, 11 pages, 2021.

[8] L. Li, C. Mao, H. Sun, Y. Yuan, and B. Lei, "Digital twin driven green performance evaluation methodology of intelligent manufacturing: hybrid model based on fuzzy rough-sets AHP, multistage weight synthesis, and PROMETHEE II," *Complexity*, vol. 2020, no. 6, 24 pages, Article ID 3853925, 2020.

[9] K. Zhang, "Web news data extraction technology based on text keywords," *Complexity*, vol. 2021, Article ID 5529447, 11 pages, 2021.

[10] L. Li and C. Mao, "Big data supported PSS evaluation decision in service-oriented manufacturing," *IEEE Access*, vol. 8, no. 99, pp. 154663–154670, 2020.

[11] J. Li, Y. Fu, J. Xu, C. Ren, X. Xiang, and J. Guo, "Web application attack detection based on attention and gated convolution networks," *IEEE Access*, vol. 8, pp. 20717–20724, 2020.

[12] M. Babiker, E. Karaarslan, and Y. Hoscan, "Web application attack detection and forensics: a survey," in *Proceedings of the 6th International Symposium On Digital Forensic And Security*, pp. 1–6, Antalya, Turkey, March 2018.

[13] S. Ninawe and R. Wajgi, "Detection of DOM-based XSS attack on web application," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 33, pp. 633–641, 2020.

[14] K. B. Jalbani, M. Yousaf, M. S. Sarfraz, J. Oskouei, A. Hussain, and Z. Memon, "Poor coding leads to dos attack and security issues in web applications for sensors," *Security and Communication Networks*, vol. 2021, Article ID 5523806, 11 pages, 2021.

[15] X. Yu, W. Yu, S. Li, X. Yang, Y. Chen, and H. Lu, "WEB DDoS attack detection method based on semisupervised learning," *Security and Communication Networks*, vol. 2021, Article ID 9534016, 10 pages, 2021.

[16] A. K. Dalai and S. K. Jena, "Neutralizing SQL injection attack using server side code modification in web applications," *Security and Communication Networks*, vol. 2017, Article ID 3825373, 12 pages, 2017.

[17] R. R. Echeverria, J. C. Preciado, Á. Rubio-Largo, J. M. Conejero, and Á. E. Prieto, "A pattern-based development approach for interaction flow modeling language," *Scientific Programming*, vol. 2019, Article ID 7904353, 15 pages, 2019.

[18] A. K. Kayani and M. Q. Saeed, "Comparative analysis of antivirus evasion malware creator tools of kali linux, with proposed model for obfuscation," in *Proceedings of the 2021 international conference on cyber warfare and security, iccws*, pp. 24–29, Islamabad, Pakistan, November 2021.

[19] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly- and signature-based IDS for network security using hybrid inference systems," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6639714, 10 pages, 2021.

[20] X. Li, X. Jin, Q. Wang, M. Cao, and X. Chen, "SCCAF: a secure and compliant continuous assessment framework in cloud-based IoT context," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3078272, 18 pages, 2018.

[21] E. Chatzoglou, G. Kambourakis, and C. Kolias, "Your wap is at risk: a vulnerability analysis on wireless access point web-based management interfaces," *Security and Communication Networks*, vol. 2022, Article ID 1833062, 24 pages, 2022.

[22] He-J. Lu and Y. Yu, "Research on WiFi penetration testing with Kali Linux," *Complexity*, vol. 2021, Article ID 5570001, 8 pages, 2021.

[23] U. Khadam, M. M. Iqbal, M. Alruily et al., "Text data security and privacy in the Internet of things: threats, challenges, and future directions," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 7105625, 15 pages, 2020.

[24] M. Bilal, M. Asif, and A. Bashir, "Assessment of secure openid-based daaa protocol for avoiding session hijacking in web applications," *Security and Communication Networks*, vol. 2018, Article ID 6315039, 10 pages, 2019.

[25] K. Gencer and F. Başçiftçi, "The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression," *Egyptian Informatics Journal*, vol. 22, no. 2, pp. 145–153, 2021.

[26] M. Saulaiman, M. Takacs, M. Kozlovszky, and A. Csilling, "Fuzzy model for common vulnerability scoring system," in *Proceedings of the 2021 - ieee 15th International Symposium On Applied Computational Intelligence And Informatics*, pp. 419–424, Timisoara, Romania, May 2021.

[27] R. Liu, "Network crawler technology based on Python under information," in *Lecture notes in electrical engineering*, pp. 1941–1948, Springer, Berlin, Germany, 2021.

[28] M. Júnior, T. A. Rezende, M. F. Pontes, D. Assis, and G. Tavares, "Development of a focused web page crawler based on genre and content," in *Proceedings of the 20th international conferences on www/internet 2021 and applied computing 2021*, pp. 77–84, Portugal Portugal, August 2021.

[29] J. Pan, "Application of web crawler technology based on Python in big data environment," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 102, pp. 571–577, 2022.