ISBN: 978-981-18-3959-7
2022 the 12th International Workshop on Computer Science and Engineering (WCSE 2022)
doi: 10.18178/wcse.2022.06.048

# Design and Implementation of WAF Bypass Web Vulnerability Detection System

Zengyu Cai[1,3], Mengya Zhang[2,3], Zi'an Wang[1], Jianwei Zhang[2,3+] , Yuan Feng[1,3], and Nan Jiang[1]

[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

[2] Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou, 450002, China

[3] Henan Key Laboratory of Food Safety Data Intelligence, Zhengzhou, 450002, China

**Abstract.** Nowadays, with more and more web applications, the phenomenon of bypassing WAF attacks on networks has only increased, which has made the situation of Internet security more and more serious. This article first provides a background on common web vulnerabilities. Secondly, the overall architecture and overall functional design of the system are proposed, and then the modules of the functional design are introduced, mainly including the design of information collection, vulnerability detection, and the design of generating reports. Next, the key technologies used by the system are introduced. Finally, the system is tested, and the test results show that the system can successfully bypass WAF to detect web vulnerabilities.

**Keywords:** web vulnerability detection, bypassing WAF, internet security, fuzzing technology.

## 1. Introduction

With the advent of big data and the 5G era, there are more and more web applications, web security issues hinder the further expansion of web application technologies. In order to deal with hackers' attacks on the web layer, many scholars have successively proposed technologies related to web vulnerability detection [1-6]. Huang Gang proposed to apply fuzzing technology to web vulnerability technology detection, using crawler technology to crawl links that may have vulnerabilities, sending links to detection blocks for detection, and using automated technology to fuzz test the site with high coverage [1]. A method for detecting second-order security vulnerabilities in the network was proposed by Liu, which completes the first-order detection by crawling the URL in the website and sending the anchor point, and the second-order vulnerability detection by crawling the URL in the anchor [2]. Zuo Dandan proposed a method to find vulnerability injection points based on changes in the state of the DOM [3]. Askar T. proposes a vulnerability scanning method that combines vulnerability scanning with information collection and continuously expands the scanning scope for vulnerability detection by collecting a large amount of relevant information [4]. Dalai proposes a method to prevent SQL injection attacks by monitoring the information entered by querying users and checking whether malicious parameters have been added to prevent SQL injection attacks [5]. Li proposes a method based on a machine-learning algorithm to detect XSS vulnerabilities, model by analyzing the operation sequence of the source code and using the model to obtain information related to XSS attacks for vulnerability detection [6].

All of the above is technical research on web vulnerability detection, and although various studies are endless, there is less research on vulnerability detection bypassing WAF. In view of the above problems, this paper has developed a Web vulnerability detection system that bypasses WAF. On the basis of the firewall's defense, the system can bypass the firewall to detect vulnerabilities and eliminate the "invisible" risks of the firewall.

+ Corresponding author. Tel.: +86-13603829696.
*E-mail address*: mailzjw@163.com.

# 2. Background Knowledge

## 2.1. Common Web Vulnerabilities

(1) **SQL Injection Vulnerability.** SQL injection attack is to mix malicious SQL statements into the SQL statements set by the system, and the program will execute this statement as a normal SQL statement. Because the user's input is also a part of the SQL statement, the attacker uses the controllable content input by the user to inject their own defined statement, change the logic of the original SQL statement, and let the database or program execute the input content. By controlling some SQL statements, attackers can add, delete, modify and check any data they need in the database, and can also directly obtain the system permissions of the database server, which can even lead to the embedding of malicious code and the implantation of back-end programs [7].

(2) **Command Execution Vulnerability.** when inputting a command, the attacker mixes some malicious code with the nature of the attack into the user's command. If the security inspection of the web is not rigorous enough, it will cause the execution of malicious code. The attacker can carry out destructive operations in the web application and execute system commands, resulting in the transfer of the permissions of the web server program. The attacker can use the permissions to read and write system files and execute system commands, Serious will lead to data leakage and system collapse, and more serious is that the system is completely controlled by the attacker [8].

(3) **The file contains vulnerabilities.** It mainly writes the frequently used functions into the file and introduces the file through the corresponding functions of PHP. The verification is bypassed because the incoming file name has not been reasonably verified or due to some omissions. At this time, if the attacker mixes malicious files, it will lead to accidental file disclosure and even malicious code injection [9].

# 3. Design of Web Vulnerability Detection System Bypassing WAF

## 3.1. System Architecture

The design of a web vulnerability detection system bypassing WAF is mainly used to detect potential web vulnerabilities. Although WAF is constantly developing, WAF will still be bypassed by attackers through various means. The system needs to bypass WAF to detect Web vulnerabilities by collecting a large amount of information. The overall architecture of the entire vulnerability detection system is shown in Figure 1.

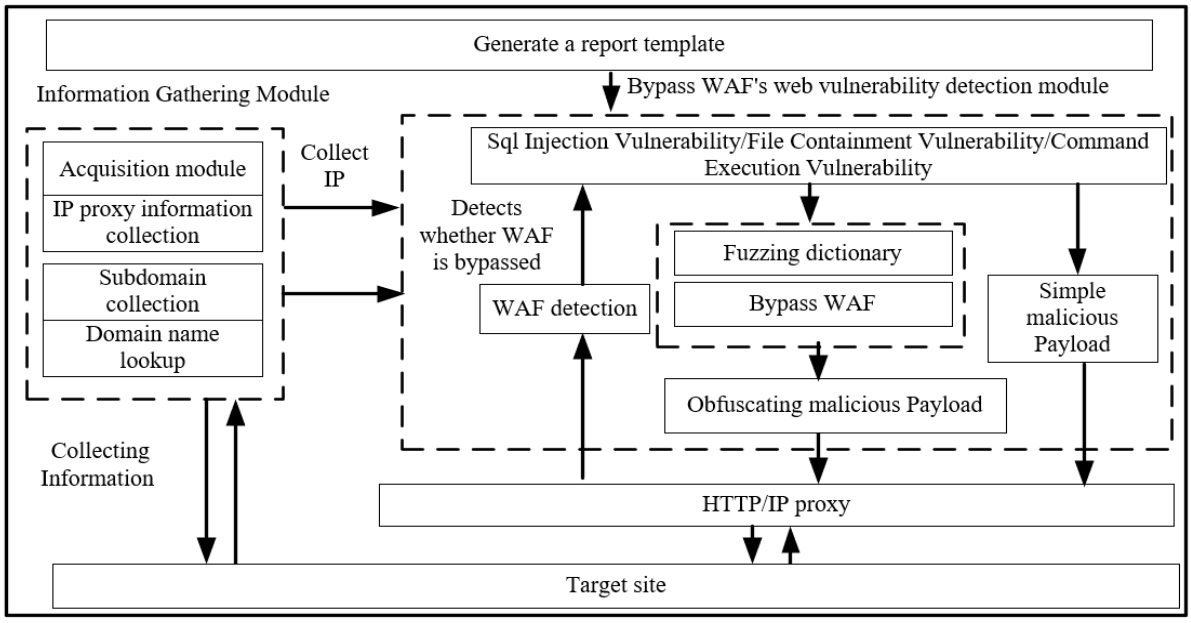


Fig. 1: Overall system architecture diagram

First, the information module collects domain name information and sub-domain name information for the target site, actively requests to collect URL information such as the interface in the response, and the IP

proxy module collects the publicly available IP proxy information on the Internet. Then, the vulnerability detection module generates a simple malicious payload to detect the website. The response information will first enter the WAF detection module. If there is a WAF, it will start to determine whether there is a bypass scheme in the WAF, which will be used first. Secondly, use fuzzing to obfuscate the payload, such as special character replacement, special encoding, etc, to perform circular fuzzing. If it is still unsuccessful, record it, and judge that the vulnerability does not exist, and enter the next vulnerability detection stage. If successful, it means that there is a vulnerability here, and the vulnerability type will be recorded. Finally, when all vulnerability detections are over, a vulnerability report will be output to complete the vulnerability detection.

## 3.2. Function Design

The functional design of the vulnerability detection system program is divided into three functional modules: an information collection module, a vulnerability detection module, and a report generation module, the overall function design diagram of the system is shown in Figure 2. When the user enters the corresponding instructions in the program interface, the program automatically searches for information and gives the user detailed usage feedback, and then detects the vulnerability based on the information. The specific functions are as follows.

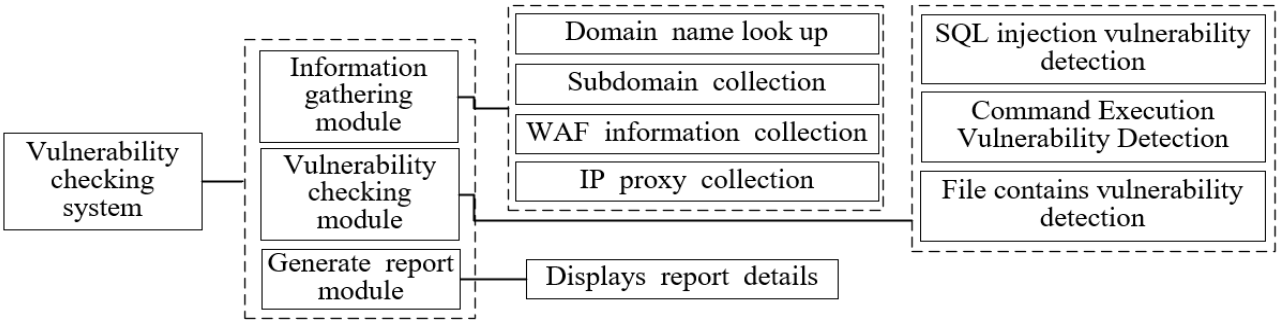


Fig. 2: Overall Function Design Diagram of Vulnerability Detection System

(1) **Domain name query:** resolve the domain name of the target site and the related affiliated domain names collected for user detection. (2) **Subdomain name collection:** when searching for a subdomain name, output real-time search results, display the status of its search, and clarify the progress. (3) **WAF information collection:** domain name detection process may encounter WAF obstruction, collection of WAF information will help programs better detect. (4) **Collection of IP proxies:** obtain IP proxies on relevant websites to deal with websites with WAF firewalls and detect as many websites as possible. (5) **SQL injection vulnerability detection:** perform SQL injection vulnerability detection on websites based on the information collected and payload keywords. (6) **Command execution vulnerability detection:** perform command execution vulnerability detection on websites based on command connectors and common functions collected. (7) **File inclusion vulnerability detection:** perform file inclusion vulnerability detection on websites based on functions that frequently cause file inclusion vulnerabilities. (8) **Show details of the report:** when a vulnerability is detected on a website, detailed information such as the site where the vulnerability occurred and the vulnerability type will be output and displayed.

## 3.3. Design of Information Collection Module

Invulnerability detection, the ability of WAF firewalls to restrict malicious access to IP can result in a large amount of IP consumption. Therefore, the system needs to collect a large number of IP proxies to cope with IP consumption when bypassing WAF in order to achieve full vulnerability detection. The main functions of the information collection module are domain name resolution, subdomain name collection, WAF information collection, and IP proxy collection.

(1) **Domain name query.** Domain Name System (DNS) is a distributed network directory service, which is mainly used for the mutual exchange of domain names and IP addresses [10]. The system uses a domain name system to manage the corresponding relationship between name and IP. Use the socket library function terminal gethostbyname () to get the IP value of the domain name.

(2) **Subdomain name collection.** When the system performs vulnerability detection on the target domain name, it also detects subdomains of the domain name. The method of searching for domain names and detecting vulnerabilities at the same time solves the problem of inconvenient input caused by too many domain names and also realizes rapid and large-scale vulnerability detection. This system introduces the asyncio module, which solves the shortcoming that python is difficult to handle asynchronously, and significantly improves the efficiency of python in IO-intensive programming. Aiodns is an accelerated DNS resolution library that can scan subdomains asynchronously to speed up the detection process.

(3) **WAF information collection.** When performing vulnerability detection, it will encounter the obstruction of WAF firewall, use simple malicious payload for detection, simulate browser to send an HTTP request, actively trigger WAF interception, and extract WAF response. In addition, a huge number of WAF signatures and bypass schemes are collected for the program to use.

(4) **IP proxy collection.** IP proxy is an important security function. The proxy server can work in the dialogue layer of the open system interconnection model and then act as a firewall. Obtain a sufficient number of proxy IPs by enumerating IP proxy websites and crawling proxy IPs, and conduct vulnerability detection through sufficient attempts. The system uses nine IP agents for vulnerability detection. Here, regular expressions are used to read the relevant information of the IP agent website.

## 3.4. Vulnerability Detection Module Design

The design principle of the vulnerability detection module is to detect the target website one by one using the vulnerability corresponding payload detection rules. If the current detection rules meet the vulnerability characteristics, the vulnerability exists. If not, the next vulnerability detection will be performed until all vulnerability detection is completed. The main functions of the vulnerability detection module are: detecting SQL injection vulnerabilities, detecting command execution vulnerabilities, and detecting file inclusion vulnerabilities. The detection process is shown in Figure 3. SQL injection vulnerability detection: use payloads of different SQL injection types to test, and determine the injection type according to the results. The command execution and file contain vulnerability detection, the program will traverse the payload keyword and queue it to test the target website.

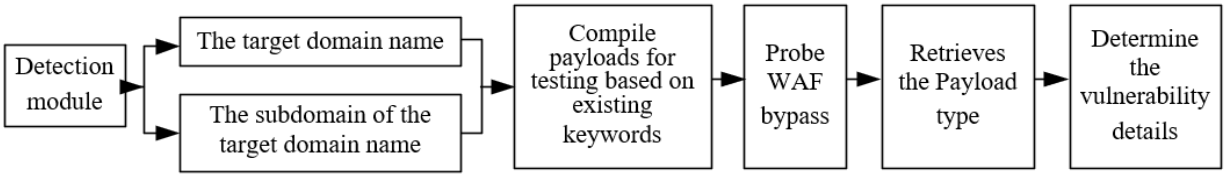


Fig. 3: vulnerability detection process

(1) **SQL injection vulnerability section.** The detection of SQL injection vulnerabilities is mainly based on the type and characteristics of servers and databases, and then detects vulnerabilities. Because of its wide range of types, this program divides vulnerability detection into three types according to the execution effect. First, error-based, using echo error injection, SQL injection based on error echo is to make the data be echoed back to the page through the contradiction of SQL statements. Second, union query injection injects the delete function when the union can be used, and delete the data that is outdated and needs to change the storage location. Because union injection can directly return injection information, not a boolean value, injection is relatively easy. Last, boolean-based, that is the injection that determines the authenticity of the condition according to the return page.

(2) **Command execution vulnerability section.** The attacker splices malicious system commands into normal commands by calling the function that executes system commands. If the website has these three situations, a vulnerability can be formed: user input is spliced into normal commands as system command parameters, and user input is not filtered or not filtered strictly.

(3) **File contains a vulnerability section.** The file inclusion vulnerability is to write a frequently used function into a file and import the file through the corresponding function of PHP. Because the system does not detect the file name, it is possible to attack the website by mixing malicious code in

the file. In this paper, the detection purpose is achieved by constructing the HTTP request parameter value and using the server to remotely load files with obvious characteristics on the local or web server.

(4) **Generate report module design**. when the user checks the website domain name for vulnerabilities, the program will generate an intuitive detection report and output the results of the vulnerability detection. The report content includes detection information such as subdomains.

# 4. Key Technologies for Web Vulnerability Detection System Implementation Bypassing WAF

There are three key technologies mainly used in the implementation of the system, mainly aiodns, asyncio asynchronous search, Boolean-based SQL injection, and bypassing WAF based on Fuzzing Technology.

## 4.1. Asyncio Asynchronous Search

When searching for subdomains, it is often blocked due to a huge amount of data, or the entire queue is blocked due to a request being blocked. In response to this problem, this system uses the asyncio module, which uses coroutines and multiplexing I/O to access other resources, so that the process does not need to wait for the result, and can directly continue subsequent operations, which greatly saves time. The subdomain collection process designed according to the asynchronous function is as follows: (1) First use async def to declare the function as an asynchronous function, then use the task name and create a total of 100 tasks. (2) When await is triggered, the blocked asynchronous call interface is suspended, waiting for the file to be loaded. (3) Traverse the tasks one by one, issue a request and wait for a response. (4) During this period, the connection receives several responses, triggers await to suspend the blocked asynchronous call interface, and waits to collect the responses into a list one by one through asyncio. gather(*tasks), which can finally save the results locally or print them out of the request. Boolean-based SQL injection vulnerability detection method.

## 4.2. Bypass WAF Based on Fuzzing Technology

This system uses fuzzing in the process of bypassing WAF. Fuzzing is an effective way to detect WAF filtering rule flaws and attempt to bypass WAF's technique. The principle of fuzzing bypassing WAF is mainly (using SQL injection as an example here) because the defense mechanism of WAF is to parse HTTP requests and match the content of each parameter in it according to the rule base. Then fill in the corresponding parameters, send them in batches, and judge whether the WAF intercepts this obfuscation method according to the returned content. If intercepted, it is judged that the obfuscation method is not desirable. If it is not intercepted, this obfuscation method can be used to inject all SQL statements. Wrap it up to bypass WAF. The test process of Fuzzing is as follows:(1) Set cookies first. (2) Set the protocol header headers again. (3) Fuzzing blasting using for loop.

# 5. System Test Results

Table 1: System module tests and test results

| The owning module | Test the process | Test results |
|---|---|---|
| Information Gathering Module. | Enter the target website xxx.xxx.com, run the program, and collect subdomains. | Outputs the collected subdomain and website address information. |
| Vulnerability detection module. | xxx.xxx.com the target website with WAF, run the program and perform vulnerability detection. | Through detection, multiple SQL injection vulnerabilities, command execution vulnerabilities, and file inclusion vulnerabilities were discovered |

The web vulnerability detection system that bypasses WAF is designed to be concise and is mainly divided into three modules, namely: information collection module, vulnerability detection module, and

report generation module. This article tests each of these three modules, and the test results are shown in Table 1.

For the test of the information collection module, enter the target URL on the system, then run the program to search for subdomains, and the system will output the collected information such as subdomains and websites. To test the vulnerability detection module, enter the target website with WAF on the system and run the program. If the target website has vulnerabilities, it will output the payload, URL and detected vulnerability types and other information. The test results show that the system can successfully bypass WAF, and detect multiple SQL injection vulnerabilities, command execution commands and file inclusion vulnerabilities, and successfully output vulnerability detection reports. Screenshots of the SQL injection vulnerability, command execution vulnerability, and actual test effect of file inclusion vulnerability are shown in Figure 4.

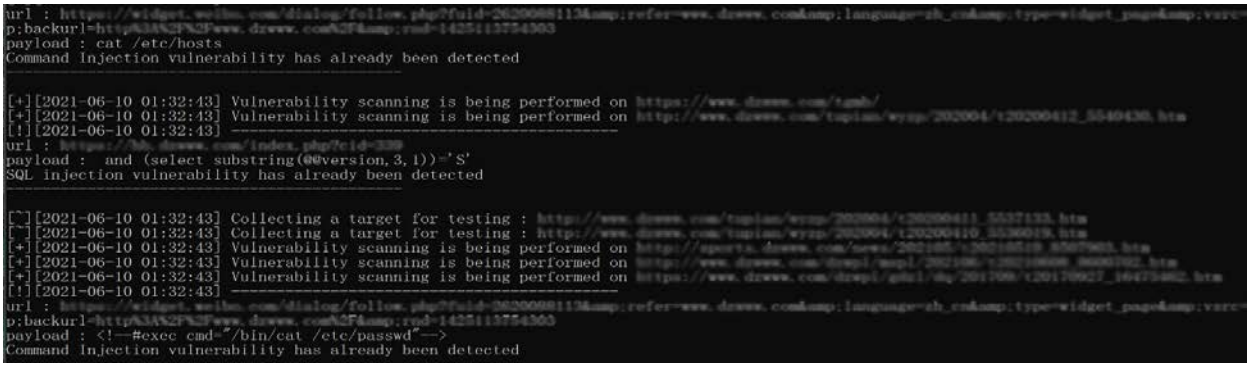


Fig. 4: Partial result graph of vulnerability detection

## 6. Conclusion

This paper designs and implements a web vulnerability detection system that bypasses WAF for the security protection of web applications in the context of the information age. The system can detect SQL injection vulnerabilities, command execution vulnerabilities, and file inclusion vulnerabilities by bypassing WAF in the working environment of WAF. The system and WAF complement each other, can see the loopholes that WAF can't see, realize the secondary protection of the Web environment, and provide an alternative solution for the protection of network security. And the system has great expansibility, but to achieve full coverage of mainstream types of Web vulnerabilities that bypass WAF, and no omissions still need further exploration, which will be the focus of the next step.

## 7. Acknowledgments

This work is supported by the National Natural Science Foundation of China (62072416), Central Plains Science and Technology Innovation Leading Talents (214200510026), Henan Science and Technology Research Project (212102210429, 222102210170, 222102210322), and China University Industry-University Research Innovation Fund - New Generation Information Technology Innovation Project (2020ITA07019).

## 8. References

[1] Huang Gang. Web Vulnerability Detection and Reinforcement System Using Fuzzing [J]. Fujian Computer, 2021, 37(03): 95-97.

[2] Liu, M., & Wang, B. (2018). A Web Second-Order Vulnerabilities Detection Method. IEEE Access, 6, 70983–70988.

[3] Zuo Dandan. Design and Implementation of XSS Vulnerability Detection Method by Wang Danfu Lihua [J]. Computer Application and Software, 2016. 33(07): 278 -281 +298.

[4] Askar T. Rakhmanov, Rustam Kh. Khamdamov, Komil F. Kerimov, et al. Automatic Vulnerability Detection Algorithm for the SQL-Injection [J]. Journal of automation and information sciences, 2019,51(7):47-54.

[5] Dalai, Asish Kumar; Jena, Sanjay Kumar (2017). Neutralizing SQL injection attack using server side code modification in web applications [J]. Security and Communication Networks, 2017(2017), 1–12.

[6] Li, C., Wang, Y., Miao, C., & Huang, C. (2020). Cross-site scripting guardian: A static XSS detector based on data stream input-output association mining [J]. Applied Sciences, 10(14), 4740.

[7] Li Bowen. On the detection and prevention of SQL injection vulnerabilities [J]. Technology and Markets, 2020, 27(11):102-103.

[8] Liu Jiadong. Common security vulnerabilities and preventive measures of PHP website [J]. Computer and Network, 2016,42(Z1):82-83.

[9] Wang Yuqiao. Research on Web Vulnerability Mining Technology Based on PHP [D]. Xidian University, 2017.

[10] Li Jianfei. Detection of unconventional domain name based on text features and DNS query characteristics [D]. Nanjing University of Posts and Telecommunications, 2019.