Research Into the Security Threat of Web Application

Yanling Zhang* and Ting Zhang

School of Information Engineering, Jiaozuo University, China
E-mail: jzdxzyl2019@163.com
**Corresponding Author*

Received 27 May 2022; Accepted 28 June 2022;
Publication 27 August 2022

Abstract

In order to effectively analyze the security threat of web application, the security threat model of web application is established. Firstly, the main problems with web application are summarized. Secondly, the main security threat of web application are analyzed, and the corresponding optimization model is constructed. An algorithm based on the improved Butterfly Optimization Algorithm (BOA) security threat optimization model is designed. Finally, a SQL injection loophole is selected for example research, and the security threat path of web application is obtained. The results show IBOA has the advantages of high optimization accuracy, global optimization and stable solution, and average accuracy rate is 99.1% and the average recall rate is 99.1%, which shows that the model has better classification effect, therefore it has the best performance.

Keywords: Security threat, web application, improved butterfly algorithm, optimization model, SQL injection loophole.

Journal of Web Engineering, Vol. 21_5, 1707–1726.
doi: 10.13052/jwe1540-9589.21514

1 Introduction

With the acceleration of China's economic development and social informatization process, the Internet has become an indispensable part of people's work and life. In order to adapt to social development, establish their own good image, expand social influence and improve work efficiency, more and more government agencies, banks, enterprises and institutions have established their own portal websites. However, because websites are usually in a relatively open environment, such as the Internet, the complexity and diversity of various web application systems lead to a large amount of vulnerabilities to the system. Trojan horse and other viruses and malicious code are rampant on the Internet. Hackers invade and tamper with website security from time to time, and even some acts of tampering with websites directly escalate into political events, seriously endangering national security and people's interests. Therefore, network security threats are becoming more and more serious. Therefore, it is necessary to ponder over how to protect network security and how to ensure the security of web application programs and provide users with fast and stable services [1, 2]. There are many factors affecting network security, mainly including the impact of viruses and network systems, the limitations of firewalls and the impact of users' daily operations.

Web application is a program that can be accessed through the web. The biggest advantage of the program is that users can easily access it. What users need is to have a browser, and there is no need to install other software. Compared with the traditional software engineering practice, the system based on Web application pays more attention to users, which makes it difficult to model web application with interactive behaviors. Therefore, in order to ensure the correctness and security of the system, web application development needs system modeling. System security specifically refers to the correctness, mutual exclusion and deadlock free of the system, which is used to explain that "bad things will never happen". Due to the interaction between web application program and browser, it is very important to ensure its reliability and security through verification and validation technology [3–5]. In the web browser cache, users can not only interact with the web page, but also interact with the web browser itself through the buttons in the web browser, such as back, forward, refresh or rewrite the URL address. They can also interact with Web databases through browsers. The user will not be prompted if the page loses its position. User's operation on the browser and web page will affect the establishment of the whole web page navigation model. This interaction behavior may have a negative impact on the security

of web application programs. A complete web application programming can correctly provide the functions users want in terms of requirements. However, when the web application program actually runs in the actual environment, it may be far from the needs of users. From the perspective of software verification, in order to ensure the correctness and security of system functions, system modeling should include the interaction details between users and web system, web and browser as a whole, and be specified in the design specification [6, 7]. The improved butterfly optimization algorithm is designed to optimize the security threat model of web application, which can obtain the high analysis precision and efficiency.

2 Current Problems with Web Application Program

2.1 Difficult Isolation of Web Running Environment

In the past, it was unlikely that text files would steal e-mail in the traditional mode of the personal computer era, but it was common on the web. In the browser world, documents and code are intertwined in the same HTML file, and completely unrelated applications can only be partially isolated at most. In addition to following a few flexible browser level security control frameworks, various interactions between different websites are implicitly acquiesced. So in fact, all web applications have paid a heavy price for uninvited malicious cross domain access. Finally, they can only reluctantly separate the code and displayed data by some clumsy methods. All web applications have failed in this matter, but sooner or later. Many content related security problems, such as cross site scripting or cross domain request forgery, are common in the web field, but they are rarely encountered in the architecture of dedicated clients.

2.2 Lack of Unified Security Mechanism in Browser

The web does not have a general holistic security model at all. In the browser field, “homology policy” mechanism can be regarded as the core security paradigm of this kind, but in fact, this mechanism with many problems is only a small subset of cross domain interaction. As a result, there are many such piecemeal adjustments, but no one can shoulder the responsibility of browser security. Due to the lack of correctness, it is impossible to judge when a single application ends and when a new application starts. In such a dilemma, what exactly is an attack? It is difficult for us to control whether we need to load or cancel permission or complete a security related task [8, 9].

2.3 There Are Problems in Collaboration of Cross Browser Interaction

When multiple browsers attempt to interact with each other, there are a series of very serious vulnerabilities that are difficult to blame on which specific piece of code. It's impossible to find out which specific products are the culprits: they're just doing their job responsibly [10, 11]. The only problem is that they don't have a public specification that all browsers should follow. Another closely related problem is that even though the security mechanisms of browsers appear to be very similar, they are actually incompatible, which rarely happened before the advent of the web. If the security models of different browsers are different, a web application development specification may be reasonable for one browser, but it may not be applicable to the other and will be misleading. Program developers often don't realize these problems unless they happen to use the affected browser. Even so, they often don't realize it until they step on a mine [12, 13].

2.4 The Boundary Between Client and Server is Blurred

The origin of the web fully conforms to the conventional "client-server" framework, however the functional boundary between user and server feedback is quickly blurred. The key problem applies java script tool that represents execution of the application logic of the HTTP server in the browser (that is, the "client"). But the security problem needs the client to be responsible, which is evidently impractical. In conventional "user-server" system, APIs are generally distinct. It is very easy to assess performance of server without considering the client, and vice versa. In addition, in each component, it is easy to isolate a small functional interval and determine what operations will be performed in this interval. However, the new model of the web and the common application APIs on the web are both vague and temporary, so it is completely impossible to rationally infer the security of a system through the previous analysis methods.

Web application system is composed of operating system and web application program. Many programmers do not know how to develop secure applications. They are not trained in secure coding. Their experience may be developing stand-alone applications or enterprise web applications that do not take into account the catastrophic consequences that can occur when security vulnerabilities are exploited.

First, there are three hidden dangers in providing security services to the public in most web applications: first, the security services should not

be provided to the public. Second, the server puts the data that should be private into the area of public access, resulting in the disclosure of sensitive information. Third, the server trusted the data from untrusted data sources, resulting in being attacked [14].

Many web server administrators have never looked at their servers from another perspective and have not checked the security risks of the servers, such as using port scanners for system risk analysis. If they had done so, they wouldn't have run so many services on their own systems, which didn't need to run on the machines that officially provide web services, or they didn't need to be open to the public. In addition, they did not modify the banner information of the application program providing services to the outside, so that the attacker can easily obtain the relevant version information of the application program provided by the web server, and find the corresponding attack method and attack program according to the information. Many web application programs are vulnerable to attacks through server, programme, and inner code. The attacks directly bypass the security measures of the surrounding firewall. There are some problems in web application security, such as improper entry, unavailable estimate regulate, unavailable description and risk management, buffer overflow, injection attack, exception error handling, denial of service attack, unsafe allocation management and so on. The four main attacks are as follows:

2.4.1 Injection attack

At present, there are more and more network applications based on database. At the same time, software used to search SQL injection points can be seen everywhere on the network. Attackers only require some theory basis to use professional software to search attacked objects. The increase of attack targets and attackers makes the SQL injection attack expand in recent years.

(1) Dos and DDOS attack DOS is the abbreviation of "denial of service", which can intentionally attack weakness of network protocol or directly consume resource of attacked object based on barbaric method, in order to ensure computer or network be not to offer general services, or avoid collapse of system. Primitive DoS attacks must be achieved based on a lot of bandwidth resources; however individual "intruder" generally does not meet these requirements. However in the later stage, the attacker invented a distributed attack method, i.e., applied professional software to obtain some network bandwidth to launch a lot of attack needs on same object simultaneously, which is DDoS (distributed denial of service) attack. In short, DDoS

attacks are a collection of DoS attacks centrally controlled and launched by "intruders", which has difficulty in being oposed. Dos and DDoS can attack network, the server will be paralysis and availability of system will be decline.

2.4.2 Cross site attack (XSS attack)

If the browser does not receive a script, it will send it to the application for cross validation. If it does not receive a script, it will send it to the application for cross validation. The attacker uses the website program to filter the user's input insufficiently, and inputs the HTML code that can be displayed on the page and affect other users, so as to steal the user's data, use the user's identity to make some action, or carry out virus infringement on the visitor, or turn the user to a malicious website.

2.4.3 Trojan horse on the website

It refers to uploading a Trojan horse program to a website, then generating an online horse with a Trojan horse generator, uploading it to the space, and adding code to make the Trojan horse run when opening the web page! As the disseminator of the virus, its purpose is to download the Trojan horse to the user's local area and further execute it. When the Trojan horse is executed, it means that more Trojan horses will be downloaded and further executed, entering a vicious cycle, so that the user's computer will be attacked and controlled.

The Trojan horse virus causes the user image to be damaged: the attacker inserts a piece of code into the normal page (usually the home page of the website). When the surfer opens the page, the code is executed, and then downloads and runs the server-side program of a Trojan horse, so as to control the host of the surfer.

In a word, they are difficult to be discovered or even verified by other users, because they are difficult to be attacked by other users. Web application attack can bypass the protection of firewall and intrusion detection products, and enterprise users cannot find the existing web security problems.

3 Web Vulnerability Threat Model

The main purpose of penetration test is to evaluate the security of the website. It can use a special vulnerability scanning tool to check the security status of the website. Professional technicians with rich experience in network attack can also be employed to play the role of "hacker". It can simulate attacks on websites from intranet, extranet and different network segments, so as

to simulate illegal operations within the enterprise and external attacks that know nothing about the internal status of the enterprise. However, the effect of this security testing method depends too much on the ability of tools and technicians, and is usually distributed throughout the enterprise in a discrete status to penetrate the website, which cannot fully simulate the process of hacker attack. A vulnerability may have certain preconditions before it can be found. Therefore, this "separate" status is easy to unilaterally estimate the vulnerability threat, thus hiding the existence of the vulnerability. Therefore, we need to model various types of Web vulnerabilities to find the greatest threat to the website [15].

The impact of attack graph as a model of attack conditions, processes and results in penetration testing This paper will design Vulnerability Threats Testing Model ($VTTM$) according to the basic idea of attack graph, and find the biggest threat of the website comprehensively and accurately by solving the model. The attack graph takes the form of directed graph. The vertex represents the security status of the website. The conversion from edge to edge contains various conditions. According to the basic idea of attack graph, this paper will establish a $VTTM$ model for the web vulnerability testing process of the website, which will be represented by weighted directed graph. Generally, the steps of establishing deletion model are: establishing status conversion diagram $\rightarrow$ combine $\rightarrow$ optimize $\rightarrow$ determine the optimal function and constraints.

The $VTTM$ model has five basic elements: website security status S, initial status S_0, purpose status S_t, security status conversion γ and value cost ratio σ. $VTTM$ model can be expressed by quintet (S_0, S,S_t, γ, σ), and its basic thought S is to use a large number of test samples to test each security status, and after the conversion condition is met, conduct status conversion. If the target status can be reached in the end, it indicates that the website has vulnerabilities corresponding to S_0 and S_t.

Each status in s contains the following elements:

(1) User permission: current user level ID. It is usually represented by enumeration, including guest (anonymous), IUSR-xxxx (Internet access user), normal (public), administrator, etc.
(2) Target: specify the target to be tested, which can be single or multiple servers. Status conversion does not involve target migration. This element can be omitted.
(3) Vulnerability: exploitable vulnerability obtained through status conversion.

(4) Ability: the harm that can be done to the website or the attack that can be carried out under the current status, such as session spoofing, password eavesdropping, Trojan horse implantation and SQL injection.

γ_{ij} means to convert the website security status S_j to after applying the test sample to S_i, and such conversion is expressed by $< S_i, S_j >$, where S_i refers to the test condition, and S_j refers to the test result. Test cases should be atomic. That is, the smallest test unit in the test link can obtain a single effect by a single means. In model $VTTM$, γ_{ij} refers to directed edge, which can assign value to each edge. Here, the weight can be expressed by the binary (C, V), where$C$ refers to the cost of implementing the test sample. It is the comprehensive value of execution cycle, the possibility of being recorded by IDs or IIS and overcoming system resources: V refers to the cost of implementing the test sample. The above two elements are set according to the tester's own needs and experience.

The process of vulnerability testing can be regarded as a status conversion process TP (called test path in this paper) based on test conditions, which is represented by linear status sequence [16]:

$$TP = \{S_0, S_1, \ldots, S_n, \ldots, S_m \mid S_n \in S_0, s_1 \in S, s_1 \in S_m, 0 \leq i \leq |S|\} \tag{1}$$

The key factor to reach the target status in $VTTM$ is to reasonably and fully design the test sample that can cover all the function modules in the website program.

After establishment, traverse the status conversion diagram to obtain all test paths from the initial status to the target status. Generally, there are depth first traversal and width first traversal. In model $VTTM$, use deep traverse first. And during the traverse, the path from s_0 to s_1 is the linear status order. Then determine the optimal sequence according to the needs, and exploit the vulnerabilities according to the optimal sequence will achieve the maximum benefits. The optimal standard is determined by the needs of the tester. For example, without considering the cost, only considering the value, the optimal function is:

$$\max \sum v_{ij} \tag{2}$$

If the value is not considered and only the cost is considered, the optimal function is:

$$\min \sum c_{ij} \tag{3}$$

If both are considered, the value cost ratio $P_{ij} = \frac{V_{ij}}{C_{ij}}$ can be introduced, and the optimal function is:

$$\min \sum P_{ij} \tag{4}$$

The conversion conditions of each status in the model can be regarded as constraints, which are mainly reflected in test condition S's internal elements of "vulnerability" and "capability".

In the establishment of the $VTTM$ model of Web loophole, all types of vulnerability threats need to be considered. Determine the initial status and target status respectively. Create their own sub status conversion diagrams, and then merge them into a global status conversion diagram. In order to reduce the complexity of calculating the optimal test sequence, the global status conversion diagram must be optimized. Optimization follows the following principles:

(1) When merging status nodes, each element of the node must be consistent. Otherwise, it cannot be merged.
(2) There is status node with conditional dependency, i.e., the two nodes S_i and S_j that can be expressed by ordered pair $< S_i, S_j >$ cannot be merged.
(3) According to the actual situation of the website, check whether the conversion conditions of each status are true. If not, remove the conversion.
(4) Add necessary paths to different sub status conversion diagrams, because the test process should be comprehensive and dynamic, and will not be confined to a specific link. Testers may find the conditions for this vulnerability through other defects of the web program, so that the security status conversion can be transferred with preconditions.
(5) If there is a loop in the optimized status conversion diagram, the status conversion in the loop may be atomic, so it is necessary to re analyze and add new status nodes to avoid generating a loop.

4 Solution Algorithm of the Web Application Threat Model

In order to solve the threat model of web application, we need to choose an effective optimization algorithm. Butterfly optimization algorithm (BOA) is a heuristic global optimization algorithm formally proposed by Sankalap Arora and Satvir Singh in 2019 to simulate butterfly foraging behavior. BOA

algorithm has the advantages of simple implementation, few parameters and novel ideas. It is suitable for solving high-dimensional function optimization problems. Compared with other active optimization algorithms proposed in recent years, the optimization performance is better and less affected by the change of dimension, so it has great research potential However, there are some problems such as slow convergence speed and low optimization accuracy for complex functions.

Aiming at the shortcomings of BOA algorithm, this paper proposes a Butterfly Optimization Algorithm based on Differential Mutation and Adaptive weight, (DMABOA). Firstly, in the global search stage, the nonlinear inertia weight is introduced to adaptively adjust the detection range and search granularity of the algorithm in different evolutionary periods. At the same time, the global formula is further improved by adding F-distribution adaptive random mutation, which improves the activity of the population and expands the global search range of the algorithm. In the local search stage of the algorithm, the TM strategy has a strong ability to judge the local number part of the jump and the extreme disturbance value, while the TM strategy accelerates the convergence speed of the algorithm [17, 18].

In the BOA algorithm, each butterfly will emit a certain concentration of fragrance. At the same time, they can also sense the fragrance of other butterflies and move towards the butterfly that emits a stronger fragrance. Butterfly's fragrance concentration formula is as follows [19]:

$$f = c \cdot I^a \tag{5}$$

Where, f refers to the fragrance concentration, I refers to the stimulation degree, which is related to the adaptability; a is the power exponent, $0 \leq a \leq 1$, c is the sensing status, $0 \leq c \leq 1$. In the BOA algorithm, $a = 0.1$, and the initial value of c is 0.01.

The process of the basic BOA algorithm is as shown below:

Step 1: Initialize the parameters: the group size N, conversion probability p, dimensional dim, the Butterfly's fragrant sensing status c and power exponent a, iteration times N_{iter} and the initial position of Butterfly $x = (x_1, x_2, \ldots, x_{\dim})$.

Step 2: Calculate the fitness value of the objective function of each butterfly and find the current optimal value. The stimulation intensity of each butterfly is determined by the fitness value, and the fragrance concentration emitted by butterfly is calculated.

Step 3: Generate a uniformly distributed random number Rand, which is used to decide whether butterfly performs global search or local search.

Step 4: When rand $<$ p, conduct global search, butterfly individual moves to the global optimal solution, and the global search formula is [20]:

$$X_i^{t+1} = X_i^t + (r^2 \cdot g^* - X_i^t) \cdot f \tag{6}$$

In the formula, X_i^{t+1} refers to the solution vector of Butterfly i at iteration t, and r is a random number evenly distributed among [0,1], g^* refers to the global optimal solution at present.

Step 5: When rand $\geq$ p, carry out local search, butterfly carries out local walk, and the local search formula is:

$$X_i^{t+1} = X_i^t + (r^2 \cdot X_j^t - X_k^t) \cdot f \tag{7}$$

In the formula, X_i^t, X_j^t, X_k^t refers to the solution vector of the Butterfly i, j, k at generation t, among which X_j^t and X_k^t are the position vectors of two individuals randomly selected in the t generation population.

Step 6: Judge whether the algorithm operation meets the end conditions. If so, record the current optimal solution and its target value. If not, update the sensory form C of fragrance concentration according to Equation (8) and return to step 2 to continue the next round of iteration.

$$c(t+1) = c(t) + \frac{0.025}{c(t) \cdot N_{iter}} \tag{8}$$

In the formula, t is the current iteration times, and N_{iter} is the maximum iteration times.

In iterative evolutionary algorithms, inertia weight can be used to adjust and control the global survey and local mining ability of the algorithm. Aiming at the disadvantages of slow convergence speed and low optimization accuracy of the basic butterfly algorithm for complex functions, this paper introduces adaptive inertia weight in the global search stage, which decreases nonlinearly with the increase of evolutionary algebra. The inertia weight function is proposed as follows:

$$\omega = \frac{1}{(t+1)^{(\alpha+\beta)^3}} \cdot \left(\frac{\alpha \cdot t^\alpha}{N_{iter}^\alpha} + \frac{\alpha \cdot t^\beta}{N_{iter}^\beta} \right) \tag{9}$$

Where, α and β are weight coefficients, and $\alpha = 3$ and $\beta = 5$. According to the analysis of the above formula, the value of inertial weight of ω is within [0, 1]. The value decreases with the increase of the number of iterations, that is, in the early stage of evolution, the weight value is large, the global search ability of the algorithm is strong, and the detection range is large, which is conducive to the algorithm to jump out of the local optimum, and the weight value decreases with the increase of the number of iterations. At this time, Butterfly performs fine mining near the optimal value, which improves the optimization accuracy and accelerates the convergence speed of the algorithm. The decreasing trend of continuous deceleration in the later stage also improves the mining ability, balances a certain search ability, and maintains appropriate population activity and the ability to jump away from local extreme points. The improved global location update formula is [21]:

$$X_i^{t+1} = \omega \cdot X_i^t + (r^2 \cdot g^* - X_i^t) \cdot f \tag{10}$$

The key factor in the global optimization algorithm is the activity of basic BOA. In order to improve the activity of butterfly population, adaptive random variation is introduced into the global formula. In the global search of the algorithm, the update of the next generation butterfly location is not only determined by the current butterfly location and the global optimal butterfly location, but also participated by the random butterfly location in the contemporary population, which increases the diversity of the butterfly population and improves the search ability of the algorithm in the global stage. The adaptive random variation formula is as follows:

$$Ra = \varepsilon \cdot (X_j^t - X_k^t) \tag{11}$$

$$\varepsilon = \varepsilon_0 + fpdf() \cdot (1 - \varepsilon_0) \cdot \left(\frac{N_{iter} - t}{\pi \cdot N_{iter}} \right) \tag{12}$$

In formula (11), X_j^t and X_k^t are the random solutions within the current group. ε is the mutation factor, and its calculation formula is as shown in formula (12), among which ε_0 is the initial mutation factor. After a large number of tests, when ε_0 is 0.1, the algorithm has the best improvement effect. $fpdf()$ is a random number that obeys the F distribution. The F distribution function presents an asymmetric peak shape, which first increases rapidly and then decreases slowly, which helps butterfly individuals adaptively converge to the position of the global optimal solution with the increase of the number of iterations. The formula combining the number of iterations and F distribution

is as follows [22]:

$$F = \frac{\Gamma\frac{\mu+\lambda}{2}}{\Gamma\frac{\mu}{2}\Gamma\frac{\lambda}{2}} \cdot \frac{\mu}{\lambda} \cdot \left(\frac{\mu}{\lambda} \cdot \left(1 + \frac{t}{N_{iter}}\right)\right)^{\left(\frac{\mu+1}{2}-1\right)} \cdot \left(1 + \frac{\mu}{\lambda} \cdot \left(1 + \frac{t}{N_{iter}}\right)\right)^{\left(-\frac{\mu+1}{2}\right)} \tag{13}$$

In the formula, λ and μ are the degrees of freedom with non-interchangeable positions. After repeated tests, when $\lambda = 3$ and $\mu = 4$, the algorithm has the strongest ability in global search and the fastest convergence speed. The global search formula is as follows:

$$X_i^{t+1} = \omega \cdot X_i^t + (r^2 \cdot g^* - X_i^t + Ra) \cdot f \tag{14}$$

Targeted Mutation (TM) strategy is an improvement of differential mutation strategy. Introducing TM tool into local optimal stage of BOA can well adjust the balance between randomness and certainty of difference vector, therefore problems of low optimal efficiency and slow running speed due to large randomness in local search stage can be prevented.

The key for BOA to obtain high-quality global optimal solution is whether the algorithm can get rid of local extremum. If only TM strategy is added, it will not only accelerate the convergence speed, but also increase the possibility of the algorithm falling into local optimal solution. To reduce this risk, the improved BOA algorithm uses the decision coefficient based on the directional difference strategy ξ to choose between the current butterfly location and the global optimal butterfly location of the current generation. If the decision random number is greater than the decision coefficient, the random walk is implemented according to current butterfly position; If the decision random number is less than or equal to the coefficient, the random walk is carried out based on the global optimal butterfly position of the current generation. TM policy model is expressed by:

If $rand > \xi$

$$X_i^{t+1} = X_i^t + (r^2 \cdot X_j^t - X_k^t) \cdot f \tag{15}$$

Otherwise,

$$X_i^{t+1} = \eta \cdot g^* + \theta \cdot (X_j^t - X_k^t) \cdot f \tag{16}$$

Where θ is the golden ratio coefficient, r is a random number distributed evenly in [0, 1], g^* is the global optimal solution at present. It can make

the new solution move towards the optimal solution and accelerate the convergence speed. It is verified from a large number of experiments that when $\xi = 0.9$, the algorithm has the strongest local search ability and is easier to jump out of local extremum; η is the disturbing factor. It is to prevent the algorithm from falling into local extremum due to the introduction of the current optimal value, further improve the optimization performance of the algorithm and find the global optimal solution. The calculation formula of η is as follows:

$$\eta = 1 + \mathrm{g}amrnd() \cdot \tan(\pi \cdot (rand - 0.5)) \tag{17}$$

In the formula, $\mathrm{g}amrnd()$ is Gamma random number, and *rand* is the jump of random number value $\mathrm{g}amrnd()$ helps the algorithm avoid local optimal value, and makes the value of disturbance factor η more flexible and diverse.

5 Example Analysis

To validate effectiveness of constructed web application threat model, a $VVTM$ model is established with SQL injection vulnerability as the research object. The target status is to get the webshell of the website. The five elements of the model are described above, and the corresponding status conversion diagram is illustrated in Figure 1.

$d(i, j, k)$ refers to the $\max \sum P_{ij}$ corresponding to the path from node S_i to node S_j, and the maximum node on this path is S_k. Thus, it can be defined that:

$$d(i,j,k) = \begin{cases} 0, i = j, k = 0 \\ -\infty, < s_i, s_j > \in \gamma \\ \max\{d(i,j,k-1), d(i,k,k-1) \\ \qquad +d(i,j,k-1), < s_i, s_j > \in \gamma, k \geq 0 \end{cases} \tag{18}$$

294506 pieces of data were used in the experiment, including XSS, SQL injection, LFI, command execution, directory traversal, RFI and normal traffic sample data. The data distribution is shown in Figure 2.

The parameter setting of IBOA is as follows: $N = 45$, $\dim = 10/50/1000$ and $N_{iter} = 50$. In order to verify the effectiveness of this algorithm, the same model is solved by using PSA and GA. The analysis results are shown in Table 1.

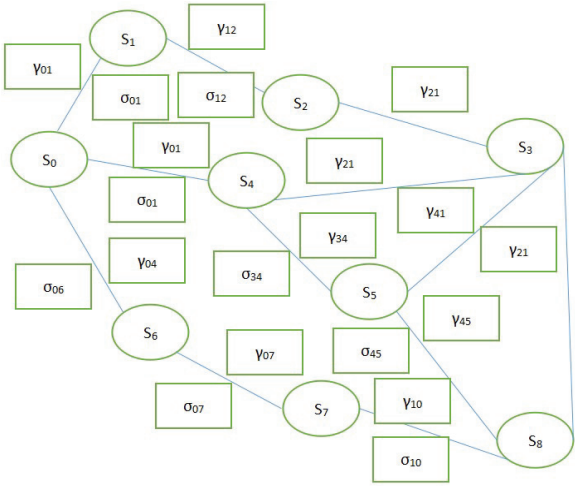

Figure 1 Status conversion diagram of SQL injection vulnerability threat.

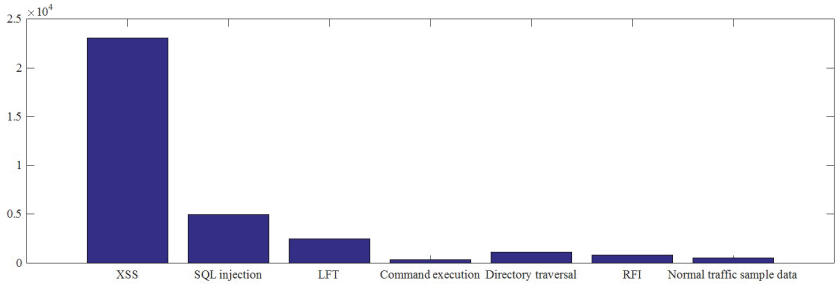

Figure 2 Simulation analysis data distribution.

Table 1 Comparison of optimization result

	dim = 10			dim = 50			dim = 50		
Algorithm	Bad Solution	Optimal Solution	Average Value	Bad Solution	Optimal Solution	Average Value	Bad Solution	Optimal Solution	Average Value
IBOA	6.56	7.12	6.84	7.22	8.15	7.69	7.43	8.57	8.00
PSA	5.24	6.33	5.79	6.43	7.03	6.73	6.65	7.24	6.95
GA	4.88	5.05	4.97	5.47	6.27	5.87	5.53	6.46	5.99

It can be seen from Table 1 that for different dimensions, IBOA obtains the largest optimal value among the three algorithms, so IBOA can obtain the optimal path. As the dimension decreases, the optimal values obtained by different methods tend to converge. Therefore, IBOA has the advantages of high optimization accuracy, global optimization and stable solution.

Table 2 Classification results of optimization model

Times	Accuracy Rate/%	Recall Rate/%
1	99.2	99.1
2	99.4	99.2
3	99.2	99.2
4	99.2	99.4
5	99.3	99.2
Average	99.2	99.2

The final optimization model is obtained through training. Finally, when the model parameters are fixed, 20% of data is used as test collection and 80% of data is used as the training collection. Experimental results are obtained by repeated execution for many times, which are illustrated in Table 2.

Through the observation of the results of many experiments, the average accuracy rate is 99.1% and the average recall rate is 99.1%, which shows that the model has better classification effect. The proposed model can obtain the better analysis results, and the correct analysis results can provide the effective measures for ensuring security of Web application.

6 Conclusion

Anomaly detection of web application is a new direction of anomaly detection. At present, the research on anomaly detection of web application is in its infancy. This paper discusses the method of vulnerability threat model based on Web. According to the basic idea of attack graph, the $VVTM$ model is established by taking sol injection attack as an example. The dynamic programming algorithm is used to solve this problem, and the safety evaluation is carried out. Compared with the traditional anomaly detection, it also has the problem of too high false alarm, as well as the challenges of effective utilization of load information, protection of users' privacy, response to data encryption and so on. This research constructs an improved BOA algorithm, which avoids the ability of individuals to jump out of local optimal solution, speeds up the convergence speed, improves the optimization accuracy, has good optimization stability when the dimension changes, and improves the optimization level of web application threat model. Through performance comparison analysis between IBOA and PSA and GA, the better performance of IBOA has been verified.

References

[1] Zohreh S. Gatmiry, Ashkan Hafezalkotob, Morteza Khakzar bafruei, Roya Soltani, Food web conservation vs. strategic threats: A security game approach, Ecological Modelling, 442, 2021, 109426.

[2] Simon Applebaum, Tarek Gaber, Ali Ahme, Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey, Procedia Computer Science, 189, 2021, 359–367.

[3] Giuseppe Cascavilla, Damian A. Tamburri, Willem-Jan Van Den Heuvel, Cybercrime threat intelligence: A systematic multi-vocal literature review, Computers & Security, 105, 2021, 102258.

[4] Andrea Tundis, Samuel Ruppert, Max Mühlhäuser, A Feature-driven Method for Automating the Assessment of OSINT Cyber Threat Sources, Computers & Security, 113, 2022, 102576.

[5] Massimiliano Rak, Giovanni Salzillo, Daniele Granata, ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems, Computers and Electrical Engineering, 99, 2022, 107721.

[6] Bin Zhao, Yi Ren, Diankui Gao, Lizhi Xu, Yuanyuan Zhang, Energy utilization efficiency evaluation model of refining unit Based on Contourlet neural network optimized by improved grey optimization algorithm, Energy, 185, 2019, 1032–1044.

[7] Adem Tekerek, A novel architecture for web-based attack detection using convolutional neural network, Computers & Security, 100, 2021, 102096.

[8] Stefano Calzavara, Hugo Jonker, Benjamin Krumnow, Alvise Rabitti, Measuring Web Session Security at Scale, Computers & Security, 111, 2021, 102472

[9] Adem Tekerek, A novel architecture for web-based attack detection using convolutional neural network, Computers & Security, 100, 2021, 102096.

[10] Bin Zhao, Yi Ren, Diankui Gao, Lizhi Xu, Performance ratio prediction of photovoltaic pumping system based on grey clustering and second curvelet neural network, Energy, 171, 2019, 360–371.

[11] Göksel Uçtu, Mustafa Alkan, İbrahim Alper Doğru, Murat Dörterler, A suggested testbed to evaluate multicast network and threat prevention performance of Next Generation Firewalls, Future Generation Computer Systems, 124, 2021, 56–67.

[12] Waleed Bin Shahi, Baber Aslam Haider, Abbas Hammad AfzalSaad, Bin Khalid, A deep learning assisted personalized deception system for

countering web application attacks, Journal of Information Security and Applications, 67, 2022, 103169.

[13] Chadni Islam, M. Ali Babar, Roland Croft, Helge Janicke, SmartValidator: A framework for automatic identification and classification of cyber threat data, Journal of Network and Computer Applications, 202, 2022, 103370.

[14] Sang Min Han, Chanyoung Lee, Poong Hyun Seong, Estimating the frequency of cyber threats to nuclear power plants based on operating experience analysis, International Journal of Critical Infrastructure Protection, 37, 2022, 100523.

[15] Sang Min Han, Chanyoung Lee, Poong Hyun Seong, Estimating the frequency of cyber threats to nuclear power plants based on operating experience analysis, International Journal of Critical Infrastructure Protection, 37, 2022, 100523.

[16] Renya Nath, NHiran V Nath, Critical analysis of the layered and systematic approaches for understanding IoT security threats and challenges, Computers and Electrical Engineering, 100, 2022, 1079997.

[17] Frank L. Greitzer, James D. Lee, Justin Purl, Abbas K. Zaidi, Design and Implementation of a Comprehensive Insider Threat Ontology, Procedia Computer Science, 153, 2019, 361–369.

[18] Arnau Erol, Ioannis Agrafiotis, Michael Goldsmith, Sadie Creese, Insider-threat detection: Lessons from deploying the CITD tool in three multinational organisations, Journal of Information Security and Applications, 67, 2022, 103167.

[19] Ha Thanh Le, Lwin Khin Shar, Domenico Bianculli, Lionel Claude Briand, Cu Duy Nguyen, Automated reverse engineering of role-based access control policies of web applications, Journal of Systems and Software, 184, 2022, 111109.

[20] Wen Long, Ming Xu, Jianjun Jiao, Tiebin Wu, Mingzhu Tang, Shaohong Cai, A velocity-based butterfly optimization algorithm for high-dimensional optimization and feature selection, Expert Systems with Applications, 201, 2022, 117217.

[21] Anurag Tiwari, Amrita Chaturvedi, A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification, Expert Systems with Applications, 196, 2022, 116621.

[22] Zohre Sadeghian, Ebrahim Akbari, Hossein Nematzadeh, A hybrid feature selection method based on information theory and binary butterfly optimization algorithm, Engineering Applications of Artificial Intelligence, 97, 2021, 104079.

Biographies

Yanling Zhang was born in Jiaozuo City, Henan Province, China, she obtained a master's degree in computer application technology from the Information Engineering University of the People's Liberation Army in 2006. She is currently an associate professor at the School of Information Engineering, Jiaozuo University, Henan Province. Her research interests include big data, machine learning and other technologies is applied research in various fields of society. Since 2017, she has presided over or participated in 4 provincial and ministerial projects, presided over the completion of 16 municipal and departmental projects, and participated in 10 projects; received funding for many times; published 1 academic monograph; and completed 10 utility model patents as the first inventor 2 invention patents; 10 papers published; 13 achievement awards.

Ting Zhang was born in Jiaozuo City, Henan Province, China, She received the B.S. degree in computer science and technology from Anyang Normal University in 2015 and the M.S. degree in computer application technology from Kunming University of science and technology in 2018. She is currently a assistant at the School of Information Engineering, Jiaozuo University, China. Her research interests include Data Mining and Cloud Computing. She has published a paper, participated in writing a SCI paper and has been published, participated in three invention patents, and participated in writing two core papers.