www.ijecs.in
International Journal of Engineering and Computer Science
Volume 14 Issue 02 February 2025, Page No. 26814-26818
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i02.4978

Analysis of Security Threats to Web Applications and Company Websites

Denys Malakhov [1*]

[1] CEO, expert in creating websites based on WordPress DM LLC WEB STUDIO, Florida

Abstract

The article analyzes the security threats of web applications and company websites. Two-tier and three-tier architectures of web applications are considered. An algorithm for assessing risks and searching for vulnerabilities in the web application infrastructure is presented, and the possibility of automating the testing process is noted. The most common threats to the security of web applications are identified. The main means of ensuring the security of web resources are provided.

Keywords: *web applications, websites, information security, security threats, security tools.*

1. Introduction

According to Storm Wall, the number of DDoS attacks worldwide increased by 63% in 2023 [1]. An increase in their power, the frequency of attacks aimed at the DNS protocol, and multi-vector attacks were recorded. There has been an increase in the number of attacks on the entertainment and e-commerce industries, which are actively digitizing, making them a target for attackers looking to exploit security system flaws.

Currently, the number of different web applications is rapidly increasing, some of which are created using automated tools [2]. At the same time, insufficient attention is paid to the quality and security of program code, leading to the emergence of vulnerabilities of varying degrees of criticality. These circumstances actualize the importance of ensuring complex security of companies' web resources.

The work goal is to analyze security threats to web applications and company websites. To achieve this, an analysis and synthesis of materials on the information security of web resources was carried out, and a system-structural approach was applied to consider the key aspects of the research problem.

Web applications are additional software designed to automatically perform certain manipulations on web servers and use web browsers as user interfaces [3]. They are executed on a server, which is accessed by sending and receiving packets via the HTTP protocol. Web applications have a client-server architecture. In systems where a server is used to directly and completely respond to user requests, a double-linked architecture is implemented, in which three basic components are distributed between two links (Fig. 1) [4].

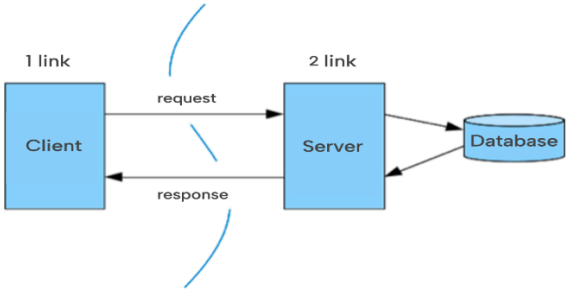

Figure 1. double-linked architecture of a web application

In the case when the application is divided into two or more parts that can be executed on separate devices, a triple-linked architecture is implemented (Fig. 2). Parts of a web application communicate by exchanging messages in a prescribed format. Data presentations are carried out on the client side, and resource management is carried out on the database server side. The third link of the system is the application server.

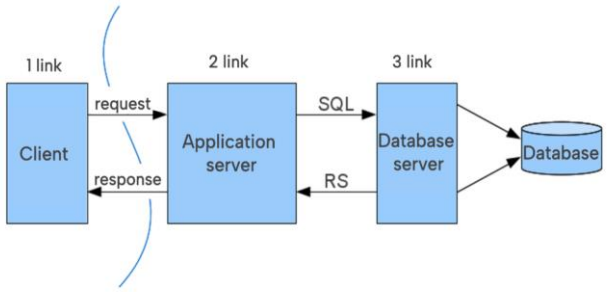

Figure 2. triple linked architecture of a web application

Specific protection means and measures depend on the infrastructure and priority risks of a particular web resource [5]. To assess risks, it is necessary to conduct an audit of the web infrastructure: map domains, study DNS history, check the domain for spam and malware, find duplicate domains, analyze the code and TLS certificates. Next, a risk assessment and a search for infrastructure vulnerabilities are carried out, which require:

- identify web applications, attacks on which can cause the greatest losses;
- search for known vulnerabilities using IoT search engines to discover services, IP addresses and other potentially risky aspects of web applications;
- scan the infrastructure to detect vulnerabilities on all levels;
- when developing software, yourself - scan application code for security.

The process of searching for vulnerabilities can be automated [6]. The generally accepted methodology for testing web resources is the OWASP method, which is an iterative process, the task of each stage of which is to collect additional information about the object under study, which will help in subsequent analysis (Fig. 3). The choice of tools for searching and exploiting vulnerabilities depends on the available set of knowledge about the information system being studied. The current list of system data can be considered as the state of the system, and the use of various utilities as an action. Since the process of searching for vulnerabilities is similar to Markov decision process, and the task of automated searching for vulnerabilities is characterized by a variety of input and output data and the lack of correct application-vulnerability pairs, a reinforced learning model can be used to develop an intelligent system for searching vulnerabilities.

Figure 3. Web application infiltration testing process

To organize the protection of web applications and mitigate risks, it is necessary to consider the most common security threats [7-9]:

1. SQL injections. An attacker injects invalid data into a web application to force it to perform an action for which the application was not designed.
2. Cross-Site Scripting, XSS. The attacker injects arbitrary JavaScript code into pages viewed by other users. If there is an XSS vulnerability in a web application, then attackers can damage the site's content, intercept user sessions, and perform other actions on behalf of the resource. The most common types of XSS are:

- Reflected: the attacker creates and sends the user a link with malicious code, after

which the server reflects and executes this code on the client side;

- DOM-based: using DOM manipulation, the virus code is executed in the user's browser;
- Stored: Malicious code is stored on the web server and executed when requests are made to the page.

3. Cross-Site Request Forgery (CSRF). The vulnerability allows attackers to issue requests on behalf of a web application, forcing authenticated users to perform unwanted actions such as changing their password, making a transaction, publishing or deleting content.
4. Clickjacking. An attack in which an attacker disguises malicious links as normal elements of a web page. When clicking on them, the user commits a malicious action.
5. Cross-site script inclusion (JSON / JavaScript Hijacking). The attacker uses cross-domain queries to extract private JSON or JavaScript data.
6. Buffer Overflow. The attacker enters more data into the application than it can accept, resulting in memory overwriting beyond the allocated buffer limits and a software crash. As a result, the attacker achieves a short-term or long-term denial of service and can gain complete control of the web server.

Most attacks on web servers are carried out through network firewalls, http or https ports [10]. To ensure the security of web applications at the application level, scanners and firewalls are used. Web Application Scanning (WAS) is a program that automatically checks web applications for specific security vulnerabilities using a filtering algorithm based on negative logic. This filtering, based on signatures of known attacks, gives security systems the ability to prevent requests that could match attack signatures from reaching protected servers.

Web application firewall (WAF) is a software or device that filters and monitors HTTP traffic between web servers and clients to identify and block potentially dangerous requests (Fig. 4) [11]. WAF analyzes incoming and outgoing HTTP requests, rejecting matching rules or vulnerability signatures through negative, positive, or a combination of logic filtering. Despite the spread of WAFs, a separate problem is their bypass by attackers, which requires additional countermeasures.

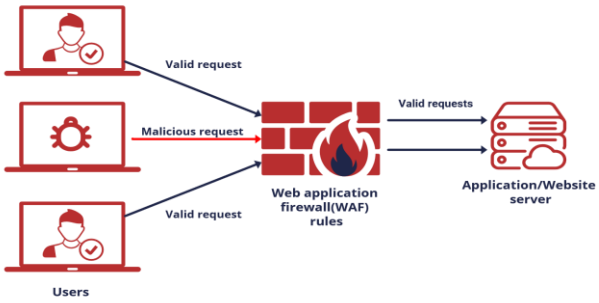

Figure 4. WAF function schematic

In addition to WAS and WAF, the following tools are used to ensure the security of web resources [12, 13]:

1. Load balancers for web applications. Allows you to distribute the load between several network devices - routers and firewalls.
2. Protection against DDoS attacks. Includes servers, the use of anti DDoS scripts and specialized equipment.
3. Captchas and software checks. Allows you to filter parasitic traffic - separate bots from real users.

To protect against risk, it is necessary to organize secure publication of data. To do this, you need to configure rules in the WAF, connecting a firewall with built-in IDS and IPS modules to detect and prevent intrusions; restrict access and enable multi-factor authentication; restrict access to administrator consoles.

Thus, ensuring the security of web resources is one of the key aspects in the field of web development. The wide functionality of modern web applications makes them easy to use and at the same time potentially vulnerable to various types of cyber-attacks. Attackers can exploit a few

possible vulnerabilities in web resources, so developers and information security specialists need to apply a set of methods to detect and prevent attacks at all stages of product creation and maintenance in accordance with the principle of continuity.

References

1. 2023: StormWall annual report on DDoS attacks [Electronic resource] // StormWall. – 2024. – URL: https://stormwall.pro/ddos-ataki-2023-otchet-za-god (date of access: 10.08.2024).
2. Pulko T.A., Drzhevetsky N.A., Romeiko M.Yu. Web Vulnerability Scanner «TechnoScan» // Endless light in science. – 2023. – № 9. – C. 212-218.
3. Shutko N.A. Theoretical concepts of protecting web applications from vulnerabilities // Bulletin of Science – 2022. – T. 4, № 11 (56). – C. 253-269.
4. Software testing. Lecture 6: client-server architecture [Electronic resource] // GitBook. – URL: https://sergeygavaga.gitbooks.io/kurs-lektsii-testirovanie-programnogo-obespecheni/content/lektsiya-6-ch1-arhitektura-klient-server.html (date of access: 10.08.2024).
5. Web application protection: from analysis to countering attacks. Part 1 [Electronic resource] // NUBES. – 2024. – URL: https://nubes.ru/blog/articles/web-application-security-part1 (date of access: 10.08.2024).
6. Vybornova O.N., Ryzhikov A.N. Automated search for web application vulnerabilities based on machine learning with reinforcement // Caspian Journal: Management and High Technologies. – 2021. – № 1 (53). – C. 91-97. – DOI: 10.21672/2074-1707.2021.53.1.091-097
7. Putyato M.M. Analysis of typical vulnerabilities when building web applications / M.M. Putyato, A.S. Makaryan, V.V. Leshchenko, V.O. Nemchinova // Bulletin of the Adygea State University. Series: Natural, mathematical and technical sciences. – 2022. – № 3 (306). – C. 77-85. – DOI: 10.53598/2410-3225-2022-3-306-77-85
8. Loginova N.V. Methods for detecting and preventing vulnerabilities in front-end development // Bulletin of Science and Education. – 2024. – № 2-1 (145). – C. 17-26.
9. Web application security: analysis of methods of protection against attacks at the Backend level [Electronic resource] // Habr. – 2024. – URL: https://habr.com/ru/articles/800017/ (date of access: 10.08.2024).
10. Ozhiganova M.I., Kurtametov E.S. Application of machine learning in protecting web applications // NBI technologies. – 2020. – T. 14, № 2. – C. 16-20. – DOI: 10.15688/NBIT.jvolsu.2020.2.3
11. Krylov I.D. Determination of information security risks such as bypassing the Web Application Firewall / I.D. Krylov, I.V. Kicha, D.P. Yakovlev, A.A. Zhdanov, D.K. Shulga, I.O. Elfimov, G.V. Belikov, V.A. Selishchev // News of Tula State University. Engineering Sciences. – 2023. – № 8. – C. 305-309. – DOI: 10.24412/2071-6168-2023-8-305-306
12. 12. Dovgal V.A., Sheredko D.I. Ensuring information security of a website in conditions of import substitution // Bulletin of the Adygea State University. Series: Natural, mathematical and technical sciences. – 2022. – № 2 (301). – C. 67-77. – DOI: 10.53598/2410-3225-2022-2-301-67-77
13. Web application protection: from analysis to countering attacks. Part 2 [Electronic resource]// NUBES. – 2024. – URL: https://nubes.ru/blog/articles/web-application-security-part2 (Date of access: 10.08.2024).