

Web Application Penetration Testing

1st Dhanesh Kumar
Patel

2nd Naveen

3rd Gorav Saini

4th Sourav Vatsyaayan

CSE-IS

CSE-IS

CSE-IS

CSE-IS

Chandigarh University

Chandigarh University

Chandigarh University

Chandigarh University

Mohali

Mohali

Mohali

Mohali

21BCS4263@cuchd.in

21BCS3572@cuchd.in

21BCS4344@cuchd.in

21BCS3541@cuchd.in

Abstract— In the current scenario, internet usage is huge and increasing day by day. Almost all business activities use network equipment, and people are becoming more and more dependent on it. As their dependence on the Internet increases, people are worried about the security of information. Because most of the business involves e-commerce, communication, payment, etc. is done over the internet. Therefore security is the most important thing for any website. Basically, security concerns are higher in organizations, offices, and finance. This article aims to solve the most important weakness problem and how they can be overcome. This article covers the most popular vulnerabilities (top 10 according to OWASP) and explains the precautions you should take when dealing with them. This article provides better understanding in a simple and easy way. As the whole world adopts new technology and everything moves online, the need for security increases. People must ensure the security of their websites and the security and privacy of their end users. So, when the world needs new technology, the need for security measures also increases. Any app or website is recommended only if it is secure and can only be done by web testers. This article explores the negative side of reality.

Keywords: Web pentesting, Website Hacking, OWASP testing guide, web vulnerability scanning, bug hunting

Introduction

Knowledge is wealth. In this digital world, every document has value. All of this information is stored in the form of Internet files. There are two types of information; public information and private information. Public information Resources published on the Internet. For example: Data generated from Google queries. Personal information is a resource hidden behind an authentication wall. For example: your email data. The email is protected by an authentication wall and requires your username and password to complete. But what if someone can read your email without authentication? What if someone could read your email without your knowledge by obtaining your credentials? This situation creates the need for secure web applications. Everything is web-based now. Most software also has its own web application version. But all web applications are vulnerable to hackers. This is why, Web Application Penetration emerge as need of the hour. Website need a defence in depth approach to mitigate against the security flaws [1]. It is essential to Penetration test every web application before it goes online and gets hacked by a Black Hat cyber warrior out there. Hackers constantly hunt for web app vulnerabilities[5]. The best way to mitigate against the hacker attacks is to learn their methodologies[2]. Here, we discuss the most common entrance tests that must be completed before the application goes live and explain the process of how to take these tests.

CLASSIFICATION OF WEB ATTACKS

A. Client Side Attacks

As the name suggests, user-side attacks are used by

attackers against users of a particular website to steal their information. The most common client-side attacks include cross-site request access (CSRF), cross-reference scripting (CORS), cross-site scripting (XSS), clickjacking, HTML injection, etc. takes place.

B. Server Side Attacks

In contrast, server-side attacks are carried out against web servers. In a server-side attack, the attacker targets the vulnerable end of the web application and sends malicious code to the server. Once the payload is successfully processed on the server, it responds to the attacker with the confidential information that the attacker requested from the payload. Classified information; server information, information about the services running on the server and their models, user information, passwords, etc. Contains.

PENETRATION TESTING

Penetration testing and vulnerability testing are two different terms. The latter will discover vulnerabilities and report them to the security team, while the latter will participate in the use of objections and attempt to delete data or increase compliance or engage in other targeted acts of violence.

Penetration testing helps developers discover vulnerabilities in their applications and protect the security of their applications. Website[3] real-time monitoring has been proven to improve website security. In order to avoid any risks, the entrance test must be performed regularly after the online application. Since new zero-day vulnerabilities are discovered every day, developers' primary responsibility is to carefully monitor the third-party services they rely on. Penetration testing is not limited to web applications but can be used on IoT devices, networks, computers, mobile applications, etc.

It is important to follow the coding process to ensure security, guide users of web application developers, and use security controls when testing access. Therefore, organizations can be better prepared to prevent malicious attacks. Additionally, security measures should be developed to prevent criminals from using these vulnerabilities during penetration tests. In the case of , although specific security measures such as encryption and authentication methods are installed, they are not fully implemented. Additionally, other measures may be taken to increase the security of the website, such as the use of firewalls and intrusion detection systems. This security system will provide strong protection against attacks and provide the company with information about all activities

through its website. Secure coding practices and regular security audits are also recommended to ensure new discoveries are quickly identified and patched.

Evaluating web applications is a necessity nowadays. Web penetration testing is generally more targeted and detailed than other types of penetration testing. The main purpose of such tests is to identify vulnerabilities and cybersecurity risks in websites, databases, codes and backend networks. To protect your website from security breaches, you need to be proactive. Access to all web applications is critical before they go online and become vulnerable to attack by black hat cyber warriors. Hackers are constantly looking for vulnerabilities in web applications, so it is important to understand methods to mitigate their attacks. There are many tools on the market that can be used to achieve the goals of web pen testing, with differences in performance and providing fast, easy results. Therefore, individuals and organizations must determine which tools are most useful for network penetration testing. The number of cyber attacks online continues to increase and requires the creation of new technologies to ensure a safe environment. In this article, we will review the proposed solutions and discuss security issues in web applications. The purpose of this article is to provide an overview of penetration testing and tools for web applications, as well as to review previous literature in this field and analyze the advantages and limitations of each solution. It also provides advice on how to select appropriate tools for network penetration testing and offers suggestions for future research in this field. This article is important from both scientific and industrial perspectives. For entry testers, looking at the results presented can help them make better decisions. Future researchers in this field will also benefit from a clear understanding of the field's limitations and future directions.

MANUAL TESTING VS AUTOMATED TOOLS

Manual penetration testing requires expertise in handling HTTP requests and responses. Penetration testers can fuzz HTTP requests to understand potential attacks that may be executed on a particular endpoint. The biggest disadvantage of using electronic devices is that they are not good. The automation tool works based on the data used by the developers. Every developer has their own testing process. Some of these may work, some may not. Therefore, not all automation tools will be

successful. It's best to follow your own strategy when taking the entrance exam. But automation tools play an important role in content discovery and search and help you save a lot of time. In a few years, all penetration testing methods will be used together[5].

1. Reconnaissance
2. Scanning
3. Exploitation
4. Maintaining Access and Privilege Escalation
5. Clearing Tracks and Reporting

Web App Vulnerabilities

OWASP creates a famous list of the top 10 vulnerabilities in web applications and provides advice on how to combat these failures. Figure 1 lists the top 10 OWASP vulnerabilities.



Figure 1

A. Broken Access Control

Some websites require verifying access level before providing functionality to users. But for everything to be usable, the program must pass the same check as the server. When assertions are not accepted, attackers can access the work without proper authorization.

Below are some examples of attacks that can use vulnerability management. In a local archive injection attack, the attacker attempts to find a page that is considered an entry in the archive path to be included in the calling page. Additionally, a remote data attack is similar to a local data attack, except that it does not involve data on the same server, the attacker controls the user input to include remote data .

B. Cryptography Failure

Cryptography refers to methods and techniques used to maintain confidentiality, non-repudiation,

integrity, and authenticity. Encryption failure is a major issue in online application security, resulting in application data exposure due to weak or lack of encryption methods. This information; may include passwords, patient health information, company secrets, credit card information, email addresses and other sensitive user information. Today's online applications manage data at rest and in transit and require effective security measures to minimize threats.

Below are some examples of attacks that can exploit encryption flaws. Some transmissions use weak encryption algorithms that can be broken in a reasonable amount of time. Encryption errors include sending confidential information in plaintext, using outdated or insecure methods, and using external or misleading information. Insufficient randomness of the encryption function and the presence of sensitive information in the arrangement are the most common causes of failure [17].

C. Injection

The translator can receive or send suspicious messages from the attacker. The attacker may provide false information to deceive the interpreter and render the instructions illegal. Three types of injection are most common: SQL injection, code injection, and XPath injection.

Below are some examples of attacks that can use injection. The first type of attack is called SQL injection; This attack involves introducing SQL commands into the input form or query to access data or change its content (such as deleting or modifying database information). The second type of attack, called code injection, involves injecting code that an application understands and runs to exploit malicious code. The third type of attack, called XPATH injection, occurs when a web application uses user input to generate XPath queries for XML documents .

D. Insecure Design

To prevent insecure design, developers are advised to use security design, threat design and patterns when creating applications.

Below are some bad examples where unsafe designs can be used. Poor design quality occurs when designers and quality assurance and/or safety teams fail to identify and identify hazards during the design process. These vulnerabilities are also caused by not following security best practices when developing applications. As the threat landscape changes, mitigating vulnerabilities requires ongoing threat modeling to protect against known attacks. Without security by design, it is difficult to detect and fix architectural errors such as unprotected

credentials, breach of trust boundaries, generation of error messages containing incorrect information, and misclassification or segmentation .

E. Security Misconfiguration

occurs when one or more components of a system (such as applications, frameworks, application servers, web servers, database servers, network routers, and platforms) are configured with invalid security settings. Security facilities must be developed, implemented and maintained.

Below are some examples of attacks that can be used to illegally exploit security. Default settings are often the source of such situations . An attacker can exploit this issue to launch multiple attacks. The strength of the attack is determined by the scope and location of the configuration.

F. Vulnerable and Outdated Components

Software components (such as modules, packages, or APIs) are parts of a system or application that continue to function.

Below are some examples of attacks that can exploit vulnerabilities in bad and legacy products. Component-based vulnerabilities occur when a software component is unsupported, outdated, or vulnerable to attack. Improper use of malware in a production environment can break web applications. For example, a company may download and use software such as OpenSSL, but neglect to update or fix it when vulnerabilities are discovered. Since many software products share the same permissions as web applications, any flaws or incompatibilities in the products can put the application at risk. Use devices with confidential information that expose applications to attacks that may target part of the application stack. For example, the following conditions may cause a target to be known as vulnerable: virus injection, unauthorized command injection, XSS, and malicious objects and objects. In the following scenario, an attacker uses an unpatched system to execute malicious code on the server. Attackers gained access to the company's internal network and then used scanning tools to detect internal systems containing faulty or outdated products. The attacker then uses a vulnerability in the previous product to upload malicious code to the application server.

G. Identification and Authentication Failures

Hackers use this vulnerability to exploit authentication errors, as the name suggests. Hackers can cause security risks by accessing user information, passwords, credentials, and other access information . Below are examples of attacks that can use validation and error validation. The

evidence is considered as a coercive claim-objection.

H. Software and Data Integrity Failures

Software and data integrity failures are caused by code and infrastructure that do not protect against legal violations.

Below are some examples of attacks that may use illegal software and data. A good example of this is when an application relies on plugins, libraries, or modules downloaded from untrusted sites, repositories, or content distribution. Insecure CI/CD pipelines can expose systems to intrusions, malicious code, and system risks. Additionally, many applications now allow automatic updates, which do not require sufficient authentication to download and apply updates to previously trusted applications. The attacker can push and distribute updates to any configuration .

I. Security Logging and Monitoring Failures

If there is no access, suspicious events and events will remain unattended for a long time, leading to security actions being illegal for longer periods of time before better access is detected. Website hackers can cause a lot of damage, but if the website application owner does not monitor the behavior of suspicious activities, hacking will be more difficult. In this case, auditing will be very useful. Below are examples of attacks that can exploit security access and tracking errors. If there is no decision-making and monitoring process, cyber attacks can have an impact and lead to limited understanding of what is happening to the system .

J. Server Side Request Forgery (SSRF)

An attacker can use this vulnerability to make a request to an unintended target in the server side application. Below are some examples of attacks that can exploit SSRF vulnerabilities. In a typical SSRF attack, the attacker may also use SSRF to simply connect the server to an internal service within the organization's infrastructure. They can force the server to connect to an external network, revealing credentials and sensitive information .

Overview of Penetration Testing Tools

This section introduces four business and open source testing tools. These are Netsparker, Acunetix, Vega, OWASP ZAP. Each tool has unique features and benefits that can be used to analyze various security aspects of the website.

A. Netsparker

Netsparker is an online security testing tool. It can detect and uncover application-level security flaws

on any website. Netsparker is available in two versions: desktop and cloud. We can use the cloud version to crawl hundreds of websites or web applications simultaneously. The desktop version is a simple tool that can be used on a single website, while the cloud version allows users to crawl multiple websites simultaneously, making it a very powerful tool for webmasters and developers.

B. Acunetix

Acunetix is an online security assessment that monitors and controls websites specifically using HTML and JavaScript. The software development lifecycle includes project management or bug management systems and includes compliance reporting. It works independently of the operating system using a web browser [48]. Just enter your target website's URL; It will provide all the necessary features. Acunetix is an excellent tool for monitoring and managing websites, especially those that use HTML and JavaScript.

C. Vega

Vega is a free and open-source online security test used to detect problems in network applications. Its graphical user interface (GUI) is developed in Java. Vega has two senses: scanner and agent. For debugging purposes, the Vega interactive web application provides a block monitor. Vega's attack modules are written in JavaScript, and since the modules are open source, they can be developed and modified by users via the JavaScript API [49]. Vega is a very powerful web application debugging tool as it can detect vulnerabilities hidden by the user. It also provides great flexibility by allowing users to customize their attack situations by adding new attacks and updating existing attacks.

D. OWASP ZAP

OWASP is a multinational, not-for-profit organization dedicated to the advancement of software security. ZAP is a simple, open-source, integrated penetration testing tool for detecting vulnerabilities in online applications. OWASP provides information, methods, documents and tools related to web security issues [50]. Using security tools such as ZAP provided by OWASP and following security codes is important for organizations that develop or maintain applications. In addition to providing security tools such as ZAP, OWASP also provides training to people involved in software development or management.

Clearing track and reporting

All requests to access the web server will be stored in log files. If an attacker gains superuser privileges on the web server, he can delete log files without leaving a trace. But getting superuser rights is not

easy; it depends on the key version and other malware used by the server. Therefore, instead of removing the logs, it would be better to use the domain name to perform access testing on the website. Advertising is the final step of the entrance exam. Write a detailed report of, including

- Name of the vulnerability
- Vulnerable Endpoint
- Technical Description of the vulnerability
- Business Impact and severity
- Proof of Concept (PoC) video or image

Results Analysis

A. Footprinting

Footprinting is a criminal technique used to gather as much information as possible to identify access opportunities to target computers, infrastructure and networks. This is one of the best ways to find defects.

Subdomain finding












Scan date: 2022-09-12 19:04:35 Domain Country: India (IN)  Subdomains found: 12 Most used IP: 52.74.41.140 (2x)		
Whois Check Check Status Copy to clipboard Download CSV Download JSON		
Subdomain	IP	Cloudflare
news.cuchd.in	142.250.186.51	
alumni.cuchd.in	52.74.41.140	
www.alumni.cuchd.in	52.74.41.140	
blog.cuchd.in	142.250.184.211	
✓ lms.cuchd.in	3.6.55.179	
cuchd.in	104.255.32.116	
✓ www.cuchd.in	23.186.192.187	
payments.cuchd.in	23.186.192.181	
✓ uims.cuchd.in	112.196.7.181	
url.cuchd.in	none	
studio.cuchd.in	none	
✓ preview.cuchd.in	none	

Figure 2

B . OpenVAS Scan

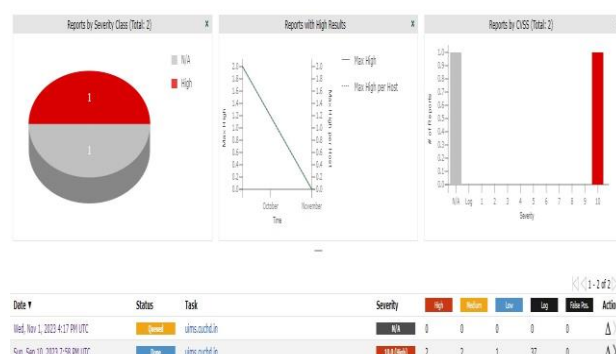


Figure 3

C.Vulnerability Detection

Vulnerability	Severity
MS15-034 HTTP.sys Remote Code Execution Vulnerability (Active Check)	10.0 (High)
SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	7.5 (High)
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	4.3 (Medium)
SSL/TLS: Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability	4.0 (Medium)
TCP Timestamps Information Disclosure	2.6 (Low)

IP	Name
95 % 112.196.7.181	80/tcp Sun, Sep 10, 2023 8:27 PM UTC
98 % 112.196.7.181	443/tcp Sun, Sep 10, 2023 8:12 PM UTC
98 % 112.196.7.181	443/tcp Sun, Sep 10, 2023 8:12 PM UTC
80 % 112.196.7.181	443/tcp Sun, Sep 10, 2023 8:12 PM UTC
80 % 112.196.7.181	general/tcp Sun, Sep 10, 2023 8:09 PM UTC

Figure 4

D.Cross-Site Scripting (XSS)

```

+J Vulnerable component: bootstrap v3.3.7
+J Component location: https://uims.cuchd.in/js/bootstrap.min.js
+J Total vulnerabilities: 4
+J Summary: XSS in data-template, data-content and data-title properties of tooltip/popover
+J Severity: high
+J CVE: CVE-2019-8331
+J Summary: XSS in data-target property of scrollspy
+J Severity: medium
+J CVE: CVE-2018-14041
+J Summary: XSS in data-container property of tooltip
+J Severity: medium
+J CVE: CVE-2018-14042
+J Summary: XSS in collapse data-parent attribute
+J Severity: medium
+J CVE: CVE-2018-14040

```



```

if ( key === "closeText" ) {
    this.uiDialogTitlebarClose.button({
        // Ensure that we always pass a string
        label: "" + value
    });
}

```

Figure 5

Conclusions

The current study focuses on research on the topic of penetration testing, mainly web penetration testing. Since manual access measurement is

Conference On Computing For Sustainable Global Development (Indiacom), New Delhi, 2016, Pp. 2159-2164.

3. Jose Fonseca, Marco Vieira, And Henrique Madeira, "Evaluation Of Web Security Mechanisms Using Vulnerability & Attack Injection", Dependable And Secure Computing, Ieee Transactions (Volume:11, Issue: 5)

4. [HTTPS://SIMPLYSECURE.BLOG/2017/07/05/FIVE-](https://SIMPLYSECURE.BLOG/2017/07/05/FIVE-)

ineffective in terms of time, money and effort, automatic measurement is reviewed. Network scanners are used to perform network penetration testing. This article begins by explaining how to take the test and how to identify the differences between manual and automated tests. It then examines the components of network penetration testing and the methods involved. Having introduced the most common web modifications and strategies to mitigate or prevent attacks, most of the current conflicts in the web environment are related to attack tools that can be used to perform penetration tests to detect these vulnerabilities. Additionally, this article reviews and compares some of the available penetration testing tools. According to reports examining various browsers, Netsparker Web Vulnerability Scanner, Acunetix, Vega, Wapiti, OWASP ZAP, IronWASP and W3af are more important than others. In the final stage, seven tests are introduced: technology, platform, interface, online/offline, vulnerability, usability and cost. A study found that not all web crawling tools have the same capabilities and that integration can provide detailed information about a website's vulnerabilities. Moreover, all these tools have their own advantages and disadvantages, so the choice depends on the needs of the organization or individual. However, we recommend that they consider features such as vulnerability detection, detailed reporting, performance and materials when choosing a device. The purpose of this article is to help network users choose the technology that best suits their needs. This research can help individuals and organizations determine the best tools to perform network penetration testing. It may be helpful for access testers to review the results provided to make better decisions.

REFERENCE

1. M. Howard And D.E. Leblanc, Writing Secure Code, Micro- Soft Press, 2002.

2. M. Khari, Sonam, Vaishali And M. Kumar, "Comprehensive Study Of Web Application Attacks And Classification," 2016 3rd International

PHASES-OF PENETRATION-TESTING/

5. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities-The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore, India, 2018, Pp. 58-62.

6. Padmaja K, "A Study On Web Application And Protection Against Vulnerability", In International Journal Of Engineering Research And Application, (Ijera), 2012, Pp.001-006.

7. "Security Code Review-Identifying Web Vulnerabilities", By Kiran Maraju.

8. M.Khari And N.Kumar, "User Authentication Method Against Sql Injection Attack", International

Journal Of Scientific And Engineering Research,2013, Pp. 1649-1653.

9.
[HTTP://WWW.THESPANNER.CO.UK/2014/05/06](http://WWW.THESPANNER.CO.UK/2014/05/06)