

**Web Security – IE2062**

**Assignment**

**Journal Book**

****

**Table of Contents**

[Acknowledgement 3](#_Toc136181182)

[OWASP top 10 vulnerabilities 4](#_Toc136181183)

[Scanning Tools 6](#_Toc136181184)

[Vulnerabilities 12](#_Toc136181185)

[**What is a vulnerability?** 12](#_Toc136181186)

[Challenges 15](#_Toc136181187)

[References 16](#_Toc136181188)

# Acknowledgement

Before I start writing my assignment, I'd like to sincerely thank Dr. Lakmal Rupasinghe, the lecture incharge, for his guidance and help in getting our entire class started on and complete our assignments.

I also want to convey my sincere gratitude to Ms. Chethana Liyanapathirana for the guidance and constant support she gave me and the rest of our class as we worked to complete this task.

# ****OWASP top 10 vulnerabilities****

**The Open Web Application Security Project (OWASP) has compiled a list of the 10 most important web application security concerns. Successful Bug Bounty searching depends on being aware of these dangers and taking steps to mitigate them. The list consists of:**

* Injection:

**occurs when an attacker delivers erroneous data to a system interpreter, giving them access to illegal data or the ability to run unwanted instructions. Examples include LDAP injection, OS, and SQL.**

* Broken Authentication:

**include authentication system issues that might be used by attackers to pose as legitimate users, steal session tokens, or take advantage of other security holes.**

* Sensitive Data Exposure:

**occurs when a program fails to provide proper security for sensitive data, such as financial or personal information, allowing hackers to steal, alter, or intercept it.**

* XML External Entities (XXE):

**exploits weak XML processors that read user input, potentially giving attackers access to private information or the ability to run remote malware.**

* Broken Access Control:

**occurs when an application does not effectively enforce access constraints, giving attackers the ability to carry out unauthorized actions like editing or deleting data.**

* Security Misconfiguration:

**include unprotected cloud storage, insufficient setups, or insecure default configurations that can leave an application vulnerable to a variety of assaults.**

* Cross-Site Scripting (XSS):

**occurs when a hacker introduces harmful scripts into a weak application, giving them the possibility to hijack user sessions or alter webpages.**

* Insecure Deserialization:

**assaults that take use of flaws in the deserialization process with the potential to execute code remotely, get elevated privileges, or use other methods.**

* Using Components with Known Vulnerabilities:

**involves utilizing software or libraries from third parties that have known security issues, leaving an application open to exploitation.**

* Insufficient Logging and Monitoring:

**occurs when a system for monitoring and recording an application is inadequate, giving attackers the ability to remain undetected and retain persistence** [1]**.**

# ****Scanning Tools****

* Sublist3r

A python application called Sublister leverages OSINT to enumerate the subdomains of websites. It aids pen-testers in obtaining and compiling subdomains for a target domain. Sublilster uses a variety of search engines, including Google, Yahoo, etc., as well as programs like Netcraft, Virustotal, etc., to retrieve reliable results [2].

* Recon-ng

A reconnaissance and OSINT program with a user interface akin to Metasploit is called Recon-ng. Running recon-ng from the command line expedites recon by automating the collection of data from public sources. Recon-ng offers a wide range of configuration, recon, and output options for various report formats. Our Recon-NG Tutorial's OSINT Contextual help and command completion are only a couple of the useful features offered by the interactive interface [3].

* Subfinder

A Go programming tool for subdomain enumeration is called subfinder. Using digital sources like Censys, Chaos, Recon.dev, Shodan, Spyse, Virustotal, and many other passive online sources, Subfinder is used to detect passive subdomains of websites. Ethical hackers and bug bounty hunters frequently employ Subfinder during the information collecting phase, commonly referred to as reconnaissance [4].

* Crt.sh

You may find all the SSL or TLS certificates for the specific targeted domain on the website crt.sh. Additionally, the website that tracks certificates is open-source. The website has a GUI interface, making it incredibly simple to acquire information, and its goal is to maintain certificate records as clear as possible. The certificate algorithms are available to you in ciphertext form. "Certificates.Saint Helena" is what the crt.sh stands for [5].”

* Httprobe

Using the tool httprobe, you can quickly scan for live http and https servers. Using this tool, you may rapidly determine which subdomains in a list are active. Httprobe was developed by Tom Hudson and is available on Github [6].

* theHarvester

You may rapidly search for active http and https servers with the utility httprobe. You may quickly ascertain which subdomains in a list are active by using this tool. Tom Hudson created Httprobe, which is accessible on Github [7].

* Infoga

Using the haveibeenpwned.com API, Infoga, a free and open-source program, can determine whether emails have been compromised. Infoga is used to scan email addresses using various websites, information-gathering search engines, and information about information that has been disclosed on websites and web apps. It is one of the simplest and most practical tools for web app and website reconnaissance for email analysis. There is a Linux version of the Infoga tool as well. This program has the ability to record data like IP address, email country, and hostname. This program gathers data from a variety of open sources, including webpages and search engines. Think of Google, Shodan, etc. The early stages of penetration testing are greatly aided by this tool for security researchers [8].

* Sherlock

Available on GitHub, Sherlock is a free and open-source program. You can use this program for nothing by downloading it for free from Github. Sherlock is skilled at locating usernames on 300 different social media platforms. As you may already be aware, many individuals create accounts on social media sites using their real names. Let's say you locate someone on a social networking platform like Facebook or Instagram. We need to visit various websites and conduct repeated, individual searches. Sherlock makes it simple for us to locate someone's social media presence online. Sherlock looks up usernames across 300 social media websites and then delivers the relevant link for that social media platform. Python is used to write Sherlock [9].

Characteristics of Sherlock:

* A free and open-source tool is Sherlock.
* Python is used to write Sherlock.
* To find usernames, people turn to Sherlock.
* More than 300 social media sites are searched by Sherlock.
* Sherlock searches 300 websites for usernames using a Python script.
* Sherlock requests a username before looking into the user's online activity on other social sites.
* Biultwoth.com

The technologies used to create a website are also visible in the source code through meta tags. The vendor who made the website's template or the name of the WordPress theme that the page is using are frequently mentioned in meta tags. A webpage's source code demonstrates that JavaScript and CSS were utilized to create the website [10].

* Webtech

Internet technology stack lookup. Identify vulnerable and old technology versions. Make a list of the websites that employ a particular web technology [11].

* Whatweb & Wappalyzer sacan

A website can be fingerprinted using the online versions of the WhatWeb and Wappalyzer programs to find applications, web servers, and other technologies. To identify the technologies being used, the tools look at the HTTP Headers on the web server and the HTML source of a web page. A typical attack vector is security flaws in well-known web applications and technologies. Maintaining current client side scripts and web apps can dramatically lower your risk of being hacked [12].

* W3techs

Discover the technologies that a specific website is employing, including the CMS, programming language, web server, and hosting company [13].

* Nmap

The Linux command-line utility Nmap is used for network analysis and security testing. Hackers, cybersecurity enthusiasts, even network and system administrators, typically use this program. It serves the following functions:

* Information about a network in real time
* a complete list of all the IPs that are active on your network
* the quantity of open ports in a network
* Give a list of the live hosts
* scanning of the host, OS, and ports [14].
* DNSEnum

Using a hands-on approach, we will learn about DNS enumeration and how it works. A website's name is translated into an IP address using a program called the Domain Name System (DNS), and vice versa. As an illustration, the DNS will intercept a user's request for a domain and retrieve the appropriate IP address before connecting the user to that IP address [15].

* DNSRecon

On GitHub, you can download the free and open-source script or program known as DNSRecon. One of the often used scripts in the security field for domain reconnaissance is dnsrecon. The language used to create this script is Python. The script cannot be used unless Python is installed on your Kali Linux operating system. A security researcher can use this script to examine all the DNS entries for AXFR and perform DNS enumeration on all different kinds of records, including SOA, NS, TXT, SVR, SPF, and others. For retrieving Googlebot-indexed subdomains, this script additionally made use of Google Dorks [16].

* Censys

This tool uses the Certificate Transparency logs kept by Censys to list subdomains. Any subdomain that has ever received an SSL certificate from a public CA should be returned [17].

* Way Back Machine

The Internet Archive, a nonprofit organization with headquarters in San Francisco, California, established The Wayback Machine as a digital archive of the World Wide Web. It was developed in 1996 and made available to the general public in 2001. It enables users to "rewind time" and view websites as they appeared in the past [18].

* Waybackurls

online crawling, which is the process of indexing data on online pages using automated scripts or crawling programs, is a crucial component of security testing. Web crawler, spider, spider bot, and a crawler are terms for these scripts or programs that perform crawling. Another Golang-based script or tool called Waybackurls is used to explore sites on stdin and get known URLs from Wayback Machines. You need to have a Golang environment installed on your PC because Waybackurls is a Golang language-based app [19].

* Dirsearch

Dirsearch is a command-line-only tool that is built on the Python programming language. When it comes to recursive scanning, Dirsearch is a lightning fast program, thus for each directory it finds, it will go back and crawl the directory in search of some further directories. An powerful command-line tool called Dirsearch is made specifically for brute-forcing folders and files on web servers or web path scanners. As an advanced tool, Dirsearch enables hackers to carry out a difficult online directories discovery with amazing performance, speed, high accuracy, advanced rectification, and contemporary brute-force approaches with appropriate outputs [20].

* UrlBuster

On the target server, there may be secret files and directories that hold sensitive and important data about the target. The security of the application may also be jeopardized by this information being made public. We need an automated method to locate these directories and files because manual testing would be difficult for the tester. UrlBuster is an automated program written in the Python programming language that searches the target server's hidden directories and files. When brute-forcing folders and files, UrlBuster supports utilizing a customized word list. In order to remain anonymous or conceal your identity, it supports the No TLS feature. The use of the UrlBuster program is free and open-source [21].

* WhatWeb

A free and open-source program called Whatweb is accessible on GitHub. Whatweb is a scanner that was created in Ruby. The target website's web technologies can all be found and recognized with this tool. This program can recognize all JavaScript libraries, content management systems, and blogging platforms used by websites. There are more than 180 modules in Whatweb. Each module is in charge of extracting a certain piece of data from the target website. Whatweb functions as a tool for information collecting and can recognize all email addresses, SQL issues, and website technologies [22].

* Wafw00f

Similar to IPS and IDS, a firewall is a security tool that keeps an eye on both incoming and outgoing traffic. A firewall can be either hardware or software. The firewall is a security tool that keeps an eye on and filters incoming traffic as well as prevents outsiders from accessing your company's or organization's internal systems without authorization. The firewall not only prevents illegal traffic from entering the system, but it also aids in preventing malicious software and files from infecting the system. It occasionally acts like an antivirus. However, it isn't an antivirus. The WAF program defends against all attacks, including SQLi and XSS. This is a free and open-source program that can determine whether or not a website has a firewall. Even this program will provide you with all the details on the firewalls that are present on the website. Like a typical firewall, the WAFW00F has the ability to filter out requests [23].

# Vulnerabilities

## **What is a vulnerability?**

A vulnerability is a gap or a weak point in the application—it could be an implementation error or a design flaw—that allows an attacker to harm the application's stakeholders. The owner of the application, application users, and other organizations that rely on the application are stakeholders.

Examples of vulnerabilities-:

* Lack of user input validation Lack of adequate logging mechanisms
* handling fail-open errors
* not properly terminating the database connection

Check out the OWASP Top Ten Project for a fantastic summary. You can download a report that goes into great depth about the major vulnerabilities and read about them online. The Top Ten are used by numerous organizations and agencies to raise awareness of application security.

List of Vulnerabilities

* [Allowing Domains or Accounts to Expire](https://owasp.org/www-community/vulnerabilities/Allowing_Domains_or_Accounts_to_Expire)

An administrator may overlook to renew a domain name or email account. Email addresses utilizing free services like Yahoo may expire after several months of inactivity, and domains have a protracted grace period before expiration.

* [Buffer Overflow](https://owasp.org/www-community/vulnerabilities/Buffer_Overflow)

Buffer overflow is the most well-known type of software security issue. It is difficult to find and difficult to exploit, but attackers have discovered it in a variety of goods and parts. In a traditional buffer overflow exploit, data on the call stack is replaced, including the return pointer for the function. There are numerous additional sorts of buffer overflows, such as Heap buffer overflow and Off-by-one Error. Format string attack is a different family of bugs that are extremely similar.

Buffer overflow vulnerabilities typically involve the breaking of a programmer's presumptions at the code level. Memory-manipulation routines in C and C++ do not perform bounds checks, making it simple for them to overwrite the buffers' allocated bounds. Most buffer overflows are caused by a confluence of memory operations and incorrect presumptions regarding the amount or composition of a piece of data.

* [Business logic vulnerability](https://owasp.org/www-community/vulnerabilities/Business_logic_vulnerability)

Business logic vulnerabilities are distinct from standard security vulnerabilities in that they involve leveraging an application's authorized processing flow in a way that has unfavorable effects for the organization. They are distinct from issues with authentication and every other category, and there are numerous significant business logic flaws, although they are less prevalent than the OWASP Top Ten. To detect errors in business logic, it is important to understand the business in its entirety.

* [CRLF Injection](https://owasp.org/www-community/vulnerabilities/CRLF_Injection)

CRLF refers to Carriage Return and Line Feed, which are used to note the end of a line in different Operating Systems. In the HTTP protocol, the CR-LF sequence is always used to terminate a line. A CRLF Injection attack occurs when a user manages to submit a CRLF into an application.

* [CSV Injection](https://owasp.org/www-community/attacks/CSV_Injection)

Websites may insert questionable input into CSV files, a practice known as CSV Injection or Formula Injection. When a CSV file is opened in a spreadsheet program like Microsoft Excel or LibreOffice Calc, all cells that begin with = will be recognized by the program as formulas.

Remember that simply making sure that user input from untrusted sources does not begin with these characters is insufficient. Additionally, you must be careful with the field separator and quotes as attackers may use these to begin a new cell and then insert the potentially harmful character in the middle of the user input.

* [Catch NullPointerException](https://owasp.org/www-community/vulnerabilities/Catch_NullPointerException)

Generally speaking, catching NullPointerException is bad practice. NullPointerException is often caught by programmers in one of three situations:

1. There is a null pointer dereference in the application. It was simpler to catch the subsequent exception than to address the underlying issue.
2. To indicate a problem, the application explicitly throws a NullPointerException.
3. The program is a component of a test harness that gives classes that are being tested unexpected input.

Only the last of these three situations is appropriate.

* Doubly freeing memory

When free() is used more than once with the same memory address as an input, double free errors occur. This can cause a memory leak and corrupt the memory management data structures of a program. An attacker can mislead a program into running code of their choosing by overwriting specific registers or memory locations, leading to an interactive shell with elevated rights. A linked list of free buffers is read when a buffer is free()'d to combine and rearrange the memory space. An attacker can overwrite valuable registers by unlinking an unneeded buffer and then calling shellcode from its own buffer.

* Expression language injection

When data under the control of the attacker enters an EL interpreter, expression language injection occurs.

Prior to EL version 2.2, an attacker might recover private server-side data accessible through implicit objects. Model objects, beans, session scope, application scope, etc. are included in this. Method invocation is permitted by the EL 2.2 specification, which enables an attacker to run arbitrary code in the context of the application. This creates a risk of server compromise because it can alter application functionality, reveal private information, and eventually gain access to system code.

In a certain pattern, Spring JSP tags will double resolve EL in some versions of the Spring Framework. This capability cannot be disabled in versions earlier than 3.0.6, hence the pattern must be avoided [24].

# Challenges

* A clear framework.
* It is a very big task and needs to be given more attention.
* Need more time.
* Too much time spent using scanning tools.
* Not being able to figure out how to start the report.
* Lack of proper understanding of the matters to be included.
* Taking a lot of time to choose suitable domains.
* It took time to find the tools because they were not working correctly.

# References

|  |  |
| --- | --- |
| [1] | "OWSAP 10 vulnerabilities," *https://owasp.org/www-project-top-ten/.* |
| [2] | "sublist3r," *https://github.com/aboul3la/Sublist3r.* |
| [3] | "recon-ng," *https://github.com/lanmaster53/recon-ng.* |
| [4] | "subfinder," *"subfinder," https://github.com/projectdiscovery/subfinder.* |
| [5] | "crt.sh," *https://github.com/vavarachen/crtsh\_scanner..* |
| [6] | "httprobe," *https://github.com/tomnomnom/httprobe.* |
| [7] | "theHarvester," *https://github.com/laramies/theHarvester.* |
| [8] | "infoga," *https://github.com/m4ll0k/Infoga.* |
| [9] | "sherlock," *https://github.com/sherlock-project/sherlock.* |
| [10] | "builtwith," *https://pro.builtwith.com/.* |
| [11] | "webtech," *https://webtechsurvey.com/.* |
| [12] | "WhatWeb & Wappalyzer Scan," *https://hackertarget.com/whatweb-scan/.* |
| [13] | "w3techs," *https://w3techs.com/sites.* |
| [14] | "nmap," *https://nmap.org/.* |
| [15] | "DNSEnum," *https://github.com/fwaeytens/dnsenum.* |
| [16] | "DNSRecon," *https://github.com/darkoperator/dnsrecon.* |
| [17] | "censys," *https://search.censys.io/api.* |
| [18] | "way back machine," *https://archive.org/web/.* |
| [19] | "waybackurls," *https://github.com/tomnomnom/waybackurls.* |
| [20] | "dirsearch," *https://github.com/maurosoria/dirsearch.* |
| [21] | "urlbuster," *https://github.com/cytopia/urlbuster.* |
| [22] | "what web," *https://github.com/urbanadventurer/WhatWeb.* |
| [23] | "wafw00f," *https://github.com/EnableSecurity/wafw00f.* |
| [24] | "vulnerabilities," *https://owasp.org/.* |