LAB15 WEB vulnerability analysis

Student ID	
Name	
Email address	
Class	
Browser	

① Select one vulnerability example among SQL injection, XSS, HTTP header injection, programming err message

https://www.netsparker.com/support/vulnerability-severity-levels-netsparker/

SQL injection, a prevalent cybersecurity threat, exploits vulnerabilities in web applications to inject malicious SQL code into their database queries. Attackers leverage this technique to manipulate the application's interaction with the database, enabling them to gain unauthorized access to sensitive data, modify or delete crucial information, or disrupt the application's functionality

There are three main kinds of SQL injection attacks:

- In-band SQL injection is the most common type of SQL injection attack. The attacker

uses the same communication channel to launch their attack and to gather their results. This type of attack is relatively easy to exploit because the attacker can see the results of their attack directly.

- Inferential (blind) SQL injection is more difficult to detect because the attacker does

not receive any direct feedback from the database. Instead, the attacker must infer the results of their attack by observing the application's behavior. This type of attack is more difficult to exploit because the attacker must be more careful about the SQL code that they inject.

- Out-of-band SQL injection is the most difficult to defend against because the attacker does not need to communicate directly with the database. Instead, the attacker communicates with a server that they control and the server then communicates with the database. This type of attack is very difficult to detect because the attacker's communication with the database is hidden from the application.

SQL injection attacks pose a serious threat to database security, as they can enable attackers to gain unauthorized access to sensitive data, destroy or modify data, escalate privileges, or even render the database server unavailable. These attacks can have devastating consequences for businesses and organizations, compromising confidential information and disrupting operations.

② Explain CWE, CVSS or CVE each

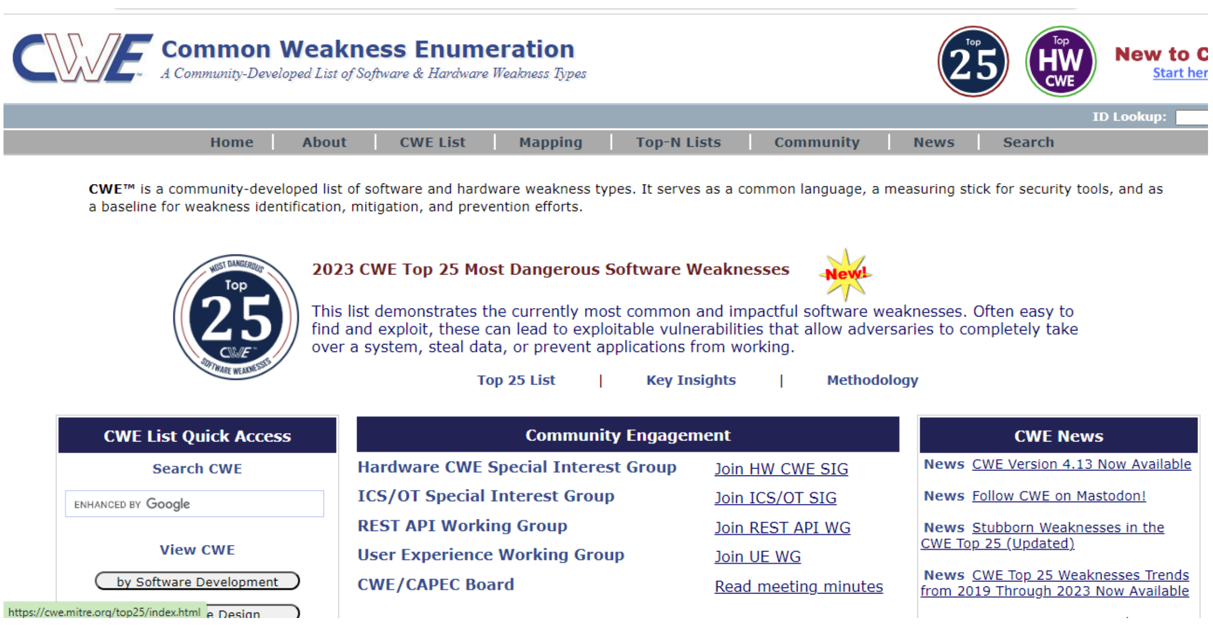
CWE (Common Weakness Enumeration) is a list of common software weaknesses that can lead to vulnerabilities. It is a taxonomy of software weaknesses that helps prioritize

remediation efforts and improve software security. Each CWE weakness is assigned a unique identifier and is described in detail, including its causes, consequences, and examples.

CVSS (Common Vulnerability Scoring System) is a standard for assessing the severity of computer security vulnerabilities. It assigns a score to each vulnerability based on three factors: the impact of the vulnerability, the exploitability of the vulnerability, and the scope of the vulnerability. The higher the CVSS score, the more severe the vulnerability.

CVE (Common Vulnerabilities and Exposures) is a list of publicly known cybersecurity vulnerabilities. Each vulnerability is assigned a unique identifier and is described in detail, including its name, description, affected products, and mitigation information. The CVE list is maintained by the National Vulnerability Database (NVD).

③ Search vulnerability level of example from CWE, CVSS or CVE and explain the level



Common Weakness Enumeration

A Community-Developed List of Software & Hardware Weakness Types

New to C
Start her

ID Lookup:

Home | About | CWE List | Mapping | Top-N Lists | Community | News | Search

CWE™ is a community-developed list of software and hardware weakness types. It serves as a common language, a measuring stick for security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.

2023 CWE Top 25 Most Dangerous Software Weaknesses New!

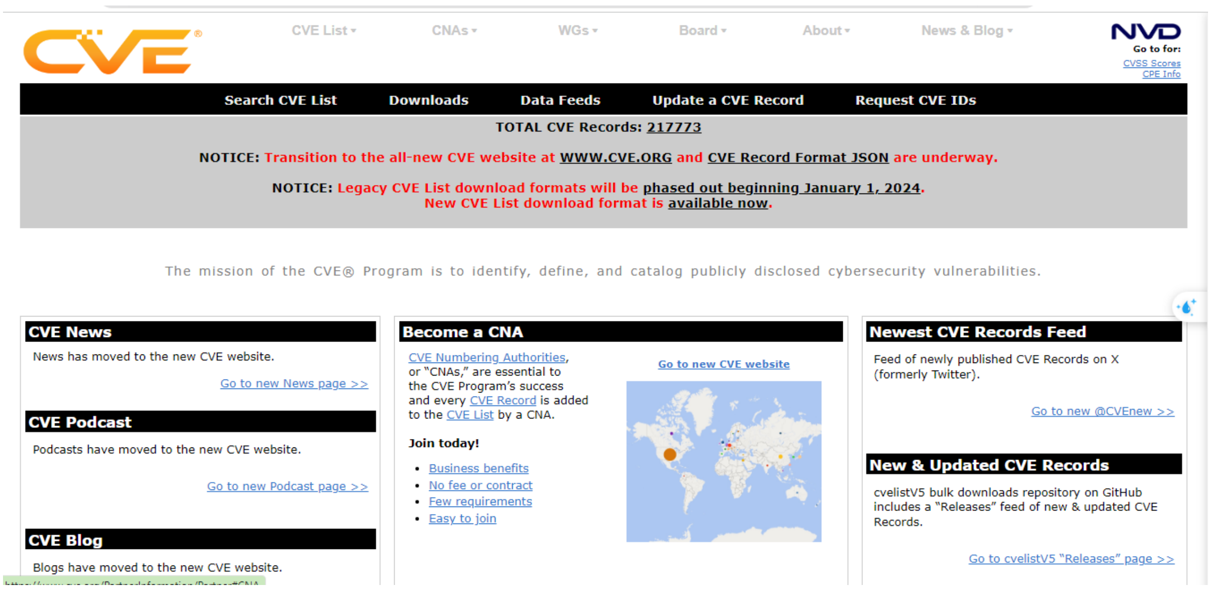
This list demonstrates the currently most common and impactful software weaknesses. Often easy to find and exploit, these can lead to exploitable vulnerabilities that allow adversaries to completely take over a system, steal data, or prevent applications from working.

Top 25 List | Key Insights | Methodology

CWE List Quick Access	Community Engagement		CWE News
Search CWE	Hardware CWE Special Interest Group	Join HW CWE SIG	News CWE Version 4.13 Now Available
ENHANCED BY Google	ICS/OT Special Interest Group	Join ICS/OT SIG	News Follow CWE on Mastodon!
View CWE	REST API Working Group	Join REST API WG	News Stubborn Weaknesses in the CWE Top 25 (Updated)
by Software Development	User Experience Working Group	Join UE WG	News CWE Top 25 Weaknesses Trends from 2019 Through 2023 Now Available
e Design	CWE/CAPEC Board	Read meeting minutes	

https://cwe.mitre.org/top25/index.html

CVE



CVE List ▾ CNAs ▾ WGs ▾ Board ▾ About ▾ News & Blog ▾

NVD Go to for: CVSS Scores CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 217773

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Legacy CVE List download formats will be phased out beginning January 1, 2024.
New CVE List download format is available now.

The mission of the CVE® Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

CVE News

News has moved to the new CVE website.

Go to new News page >>

CVE Podcast

Podcasts have moved to the new CVE website.

Go to new Podcast page >>

CVE Blog

Blogs have moved to the new CVE website.

Become a CNA

CVE Numbering Authorities, or "CNAs," are essential to the CVE Program's success and every CVE Record is added to the CVE List by a CNA.

Go to new CVE website

Join today!

- Business benefits
- No fee or contract
- Few requirements
- Easy to join

Newest CVE Records Feed

Feed of newly published CVE Records on X (formerly Twitter).

Go to new @CVEnew >>

New & Updated CVE Records

cvelistV5 bulk downloads repository on GitHub includes a "Releases" feed of new & updated CVE Records.

Go to cvelistV5 "Releases" page >>

④ Design your own processof preventing or protecting web vulnerability

Desgin your own process of preventing or protecting web vulnerability

SQL injection is a common type of web application vulnerability that allows attackers to

inject malicious SQL code into a web page. This code can then be used to steal data,

modify data, or even take control of the entire server. There are a number of ways to prevent SQL injection, including:

- Using prepared statements: Prepared statements are a way of sending SQL queries to the database that prevents malicious code from being injected. The query is first sent to the database server, which then parses it and prepares it for execution. Any user-supplied data is inserted into the query after it has been parsed, which prevents it from being interpreted as SQL code.
- Using parameterized queries: Parameterized queries are similar to prepared statements, but they allow you to specify the data types of the parameters. This can help to prevent SQL injection attacks that exploit type conversion vulnerabilities.
- Escaping user input: Escaping user input is a way of encoding special characters so that they are not interpreted as SQL code. This can be done using a variety of methods, such as using backslashes to escape single quotes.
- Validating user input: Validating user input is a way of checking to make sure that the data is in the correct format before it is sent to the database. This can help to prevent SQL injection attacks that rely on invalid data to trigger errors.
- Using a web application firewall (WAF): A WAF is a security appliance that can be used to filter out malicious traffic, including SQL injection attacks. WAFs can be configured to block specific types of SQL injection attacks, or they can be configured to use machine learning to detect and block new types of attacks.

References:

https://portswigger.net/web-security/cross-site-scripting/preventing

https://cwe.mitre.org/data/definitions/79.html

https://stackoverflow.com/questions/1996122/how-to-prevent-xss-with-html-php