Cyber Security

Lab 1 - Web Application Vulnerabilities

Fullname: Amangeldi Zhanserik
ID: 22B030301
E-mail: zha_amangeldi@kbtu.kz
Date of submission: February 27, 2025
Class time: Thursday, 15:00–18:00

School of Information System and Engineering
Kazakh-British Technical University
Academic Year 2024-2025

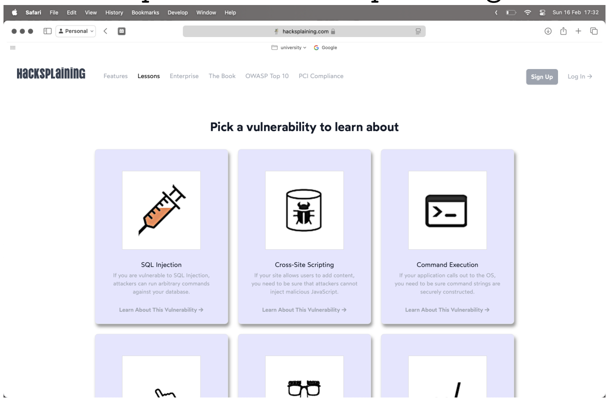
1 Introduction

This lab aims to provide hands-on experience with common web application vulnerabilities. We will be using two resources: Hacksplaining for interactive security training and OWASP Top 10 for understanding industry-standard security practices.

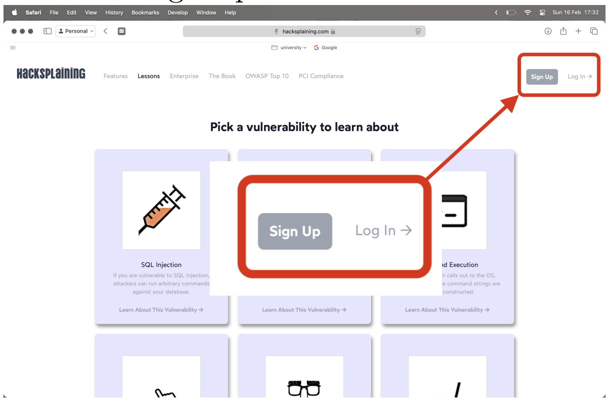
2 Part 1: Hacksplaining Security Training for Developers

2.1 Sign Up for Hacksplaining

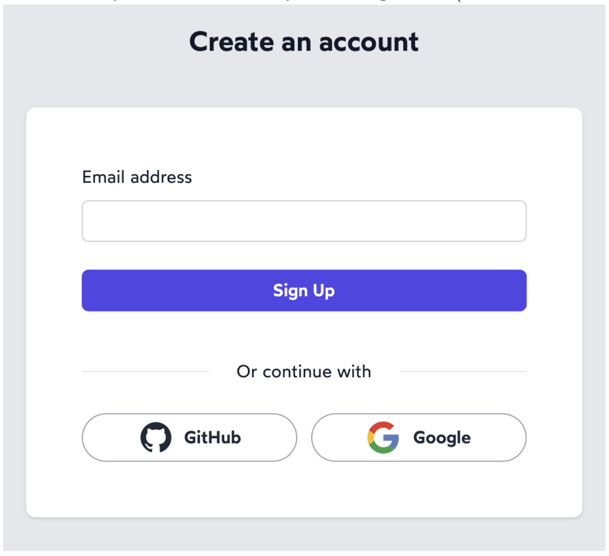
1. Go to `https://www.hacksplaining.com/`.



2. Click on "Sign Up".



3. Choose your own way to register(Email, Github or Google). I chose Google.



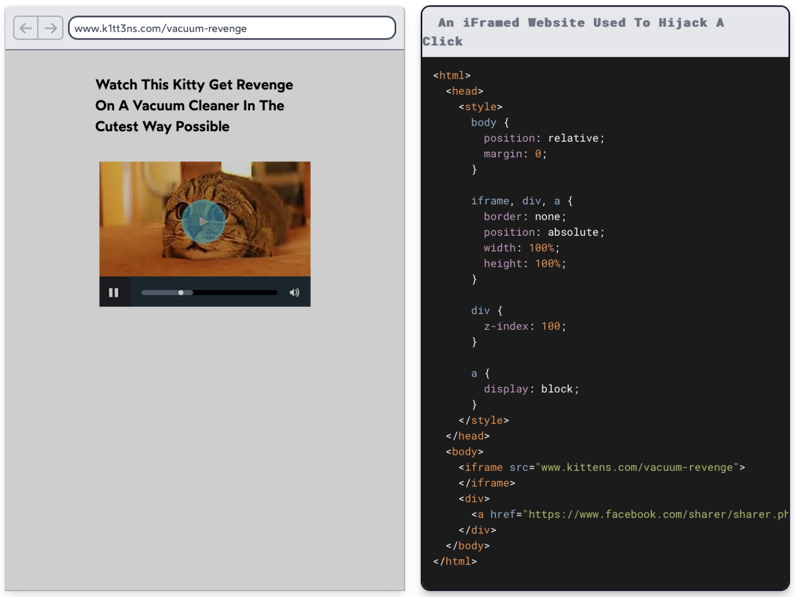
2.2 Explore Web Application Vulnerabilities

Once logged in, I explored the following vulnerabilities:

- Clickjacking
- Cross-Site Request Forgery
- Cross-Site Scripting
- SQL Injection
- Directory Traversal
- Command Execution
- Denial of Service Attacks

2.2.1 Clickjacking

Clickjacking is a malicious technique to trick a user into clicking on something different from what the user perceives they are clicking on. Clickjacking has been used to:


www.k1tt3ns.com/vacuum-revenge
Watch This Kitty Get Revenge On A Vacuum Cleaner In The Cutest Way Possible

An iFramed Website Used To Hijack A Click

```
<html>
  <head>
    <style>
      body {
        position: relative;
        margin: 0;
      }

      iframe, div, a {
        border: none;
        position: absolute;
        width: 100%;
        height: 100%;
      }

      div {
        z-index: 100;
      }

      a {
        display: block;
      }
    </style>
  </head>
  <body>
    <iframe src="www.kittens.com/vacuum-revenge">
    </iframe>
    <div>
      <a href="https://www.facebook.com/sharer/sharer.ph
    </div>
  </body>
</html>
```

- Harvest login credentials
- Trick users into turning on their web-cam or microphone
- Spread worms on social media sites
- Promote online scams
- Spread malware

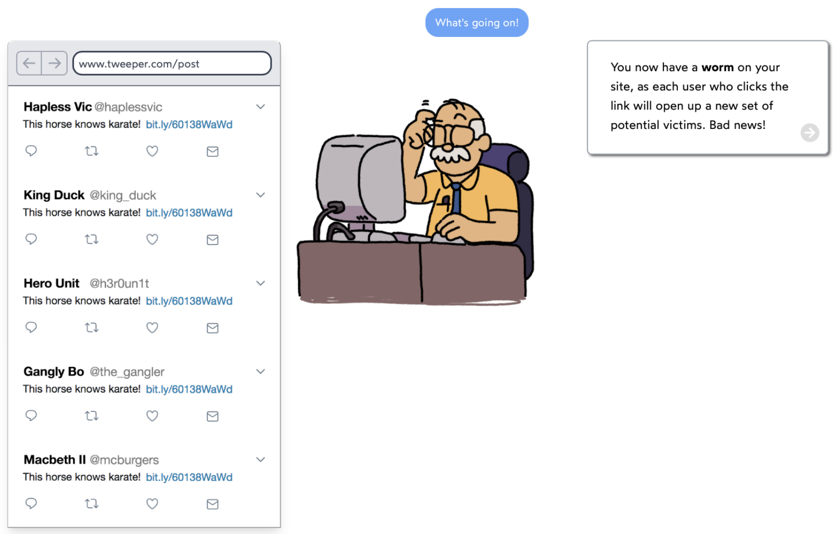
There are several ways to protect from Clickjacking:

- Content Security Policy (CSP)
- X-Frame-Options
- Frame-Killing Code

Note: Clickjacking won't affect your site directly, but it could potentially affect your users. And only you can protect them!

2.2.2 Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they are currently authenticated. CSRF attacks have been used to:



www.tweeper.com/post

Hapless Vic @haplessvic
This horse knows karate! bit.ly/60138WaWd

King Duck @king_duck
This horse knows karate! bit.ly/60138WaWd

Hero Unit @h3r0un1t
This horse knows karate! bit.ly/60138WaWd

Gangly Bo @the_gangler
This horse knows karate! bit.ly/60138WaWd

Macbeth II @mcburgers
This horse knows karate! bit.ly/60138WaWd

You now have a **worm** on your site, as each user who clicks the link will open up a new set of potential victims. Bad news!

- Steal Confidential Information.
- Spread warms on social media.
- Install malware on mobile phones

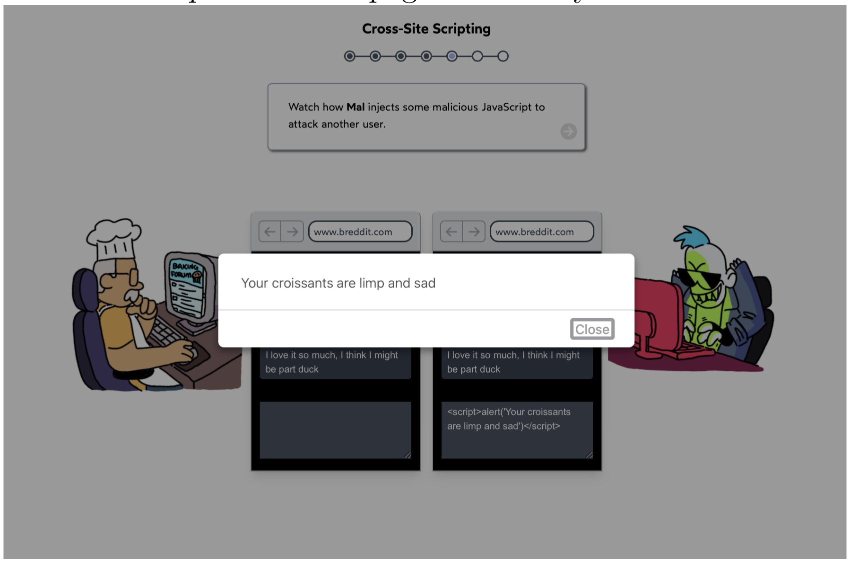
There are several ways to protect from CSRF:

- Anti-CSRF Tokens.
- SameSite Cookie Attribute.
- Include Addition Authentication for Sensitive Actions.
- Representation State Transfer (REST) APIs.

Note: CSRF attacks can be very damaging, especially if the victim is an administrative user. Therefore, it is crucial to implement proper CSRF protection mechanisms.

2.2.3 Cross-Site Scripting

Cross-Site Scripting (XSS) is a security vulnerability that allows an attacker to inject malicious scripts into web pages viewed by other users. XSS attacks have been used to:



Watch how **Mal** injects some malicious JavaScript to attack another user.

Your croissants are limp and sad

I love it so much, I think I might be part duck

I love it so much, I think I might be part duck

```
<script>alert('Your croissants are limp and sad')</script>
```

- Spreading worms on social media.
- Session Hijacking.
- Identity Theft.
- Denial of service attacks and website vandalism.
- Theft of sensitive data.
- Financial fraud.

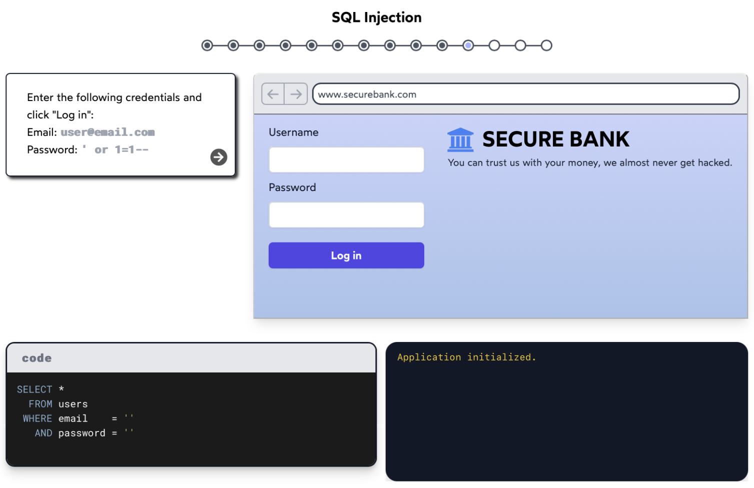
There are several ways to protect from XSS:

- Escape dynamic content.
- Allowlist Values.
- Implement a Content Security Policy (CSP).
- Sanitize HTML

Note: Avoid reading or manipulating cookies in client-side JavaScript. Mark cookies as HTTP-only to ensure they are handled by the browser and not accessible via JavaScript.

2.2.4 SQL Injection

SQL Injection is a code injection technique that might destroy your database. SQL Injection attacks have been used to:



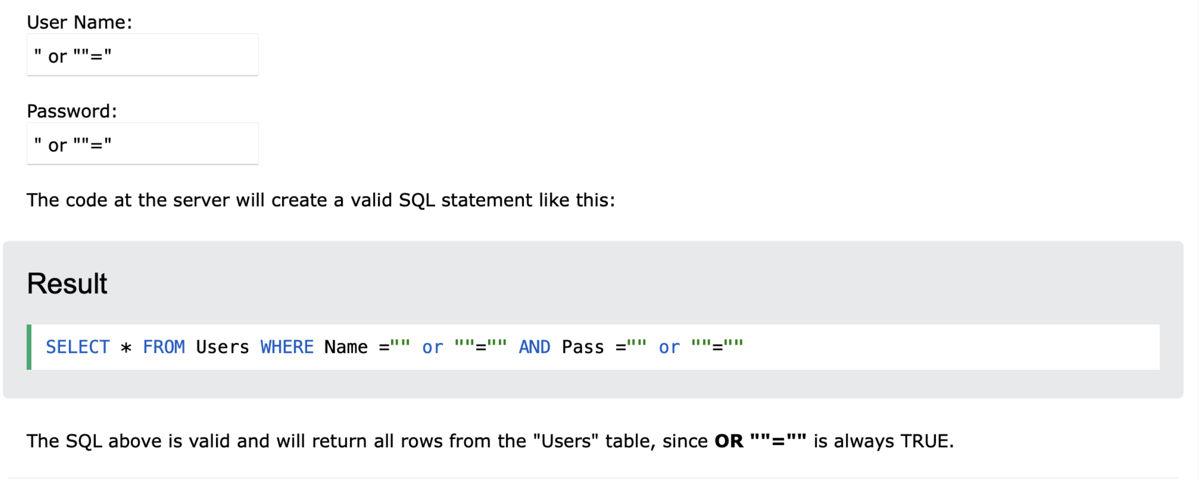
Enter the following credentials and click "Log in":
Email: user@email.com
Password: ' or 1=1--

```
code
SELECT *
  FROM users
 WHERE email    = ''
   AND password = ''
```

```
Application initialized.
```

- Extract sensitive information.
- Enumerate the authentication details of users registered on a website.
- Delete data or drop tables.
- Inject further malicious code.

Simple example of SQL Injection:



User Name:

" or ""="

Password:

" or ""="

The code at the server will create a valid SQL statement like this:

Result

```
SELECT * FROM Users WHERE Name ="" or ""="" AND Pass ="" or ""=""
```

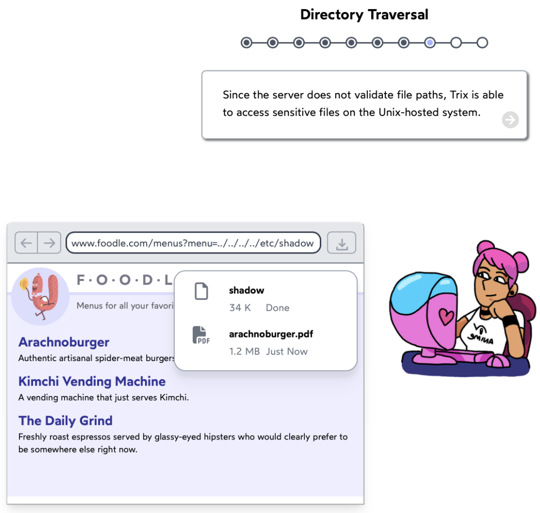
The SQL above is valid and will return all rows from the "Users" table, since **OR ""=""** is always TRUE.

Protection from SQL Injection:

- Parameterized Queries.
- Using Object Relational Mapping (ORM).
- Escaping some symbol characters.
- Sanitize user input.

2.2.5 Directory Traversal

Directory traversal vulnerabilities allow attackers to access arbitrary files on your system by manipulating URL paths. If an attacker discovers a directory traversal vulnerability, they can access sensitive files, such as configuration files, source code, and more. Protection from Directory Traversal:



Since the server does not validate file paths, Trix is able to access sensitive files on the Unix-hosted system.

www.foodle.com/menus?menu=../../../../etc/shadow

shadow
34 K Done

arachnoburger.pdf
1.2 MB Just Now

Arachnoburger
Authentic artisanal spider-meat burgers

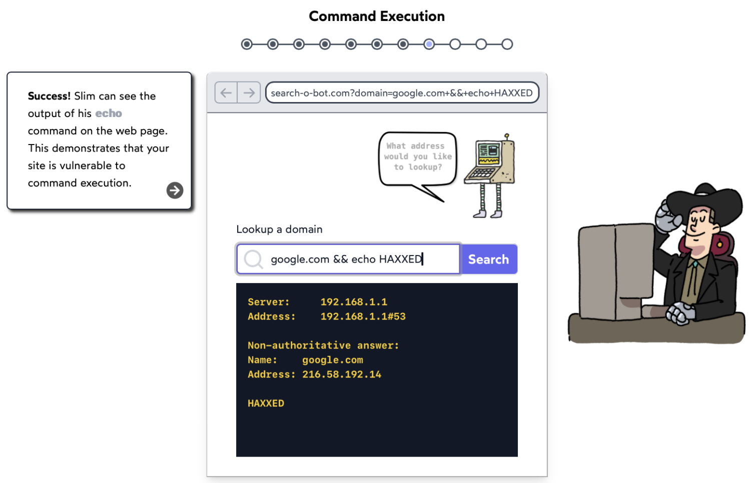
Kimchi Vending Machine
A vending machine that just serves Kimchi.

The Daily Grind
Freshly roast espressos served by glassy-eyed hipsters who would clearly prefer to be somewhere else right now.

- Use a content management system (CMS) that has built-in security features.
- Use Indirection. Each time a file is uploaded, construct a ”friendly” name for this on your site, and when the file is accessed, perform a lookup in your data-store to discover the actual file path.
- Segregate Your Documents
- Sanitize Filename Parameters. The safest approach is to restrict filenames to a list of known good characters, and ensure that any references to files use only those characters.
- Run server processes with Restricted Permissions.

2.2.6 Command Execution

Command Execution is a major security lapse that allows an attacker to execute arbitrary commands on your server. Protection ways:



Command Execution

Success! Slim can see the output of his **echo** command on the web page. This demonstrates that your site is vulnerable to command execution.

search-o-bot.com?domain=google.com+&&+echo+HAXXED

What address would you like to lookup?

Lookup a domain

google.com && echo HAXXED

```
Server:     192.168.1.1
Address:    192.168.1.1#53

Non-authoritative answer:
Name:    google.com
Address: 216.58.192.14

HAXXED
```

- Try to Avoid Command Line Calls Altogether. Use APIs whenever possible.
- Escape Inputs Correctly
- Restrict the Permitted Commands
- Perform Thorough Code Reviews
- Run server processes with Restricted Permissions

2.2.7 Denial of Service Attacks

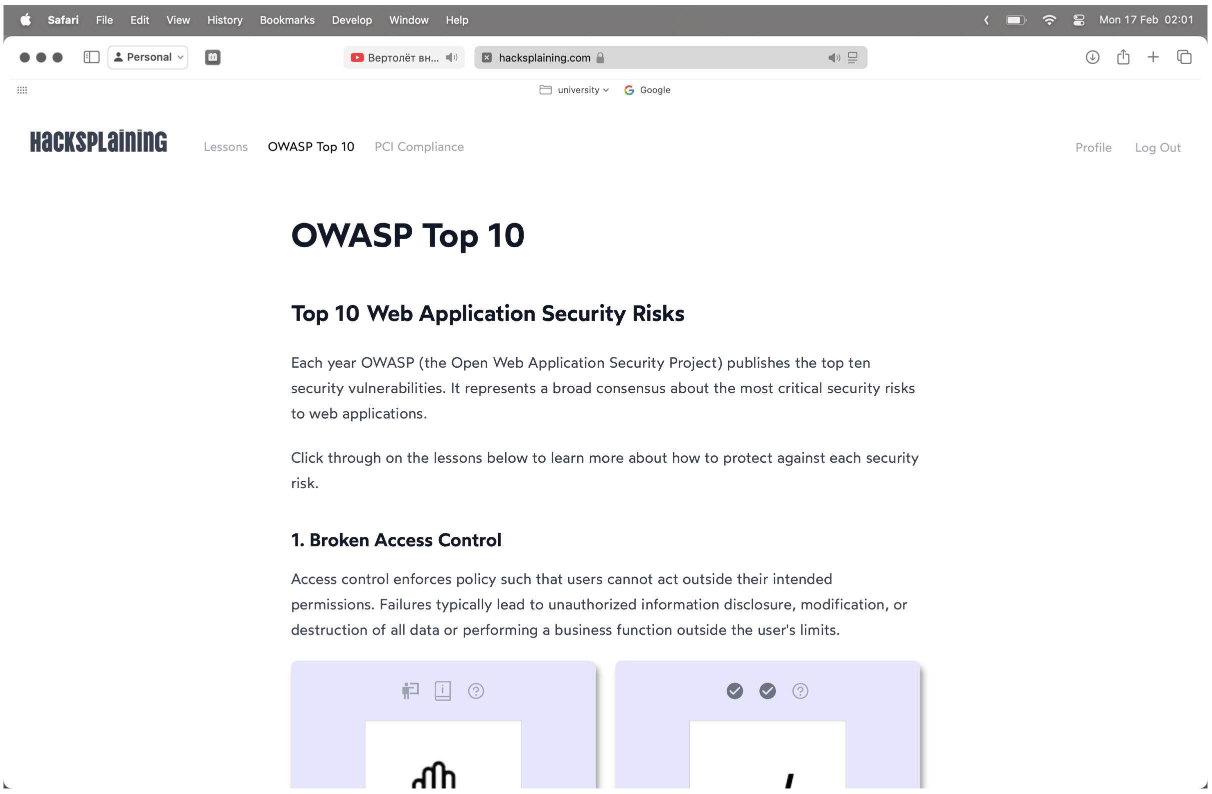
Denial-of-service attacks aim to make a site unavailable to users. They can be politically motivated, for extortion, or to cause disruption. Attackers often use distributed applications to flood a site from multiple IP addresses, making it hard to filter out all sources.

To protect against denial-of-service attacks, consider using commercial tools and services. Many cloud providers offer basic protection and alerting for free, with advanced options available at additional cost.

3 Part 2: Understanding OWASP Top 10

3.1 Open OWASP Top 10

I visited the OWASP Top 10 webpage to explore the top 10 web application security risks.



OWASP Top 10

Top 10 Web Application Security Risks

Each year OWASP (the Open Web Application Security Project) publishes the top ten security vulnerabilities. It represents a broad consensus about the most critical security risks to web applications.

Click through on the lessons below to learn more about how to protect against each security risk.

1. Broken Access Control

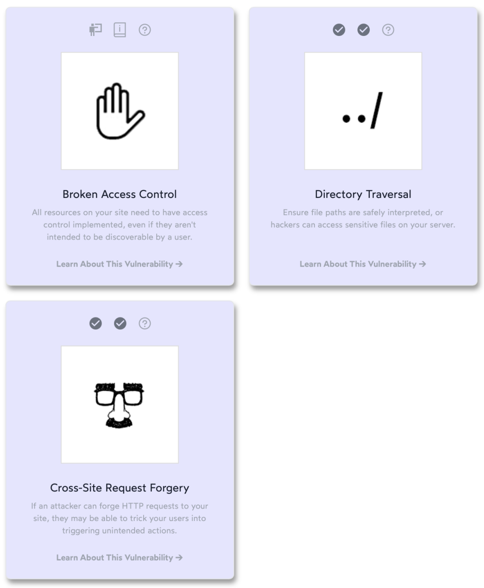
Access control enforces policy such that users cannot act outside their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.

3.2 Explore What OWASP Is

OWASP stands for Open Web Application Security Project, a nonprofit foundation that works to improve the security of software. It provides open-source projects, tools, and methodologies for securing applications. The OWASP Top 10 is a standard awareness document that highlights the most critical security risks to web applications.

3.2.1 Broken Access Control

Access control ensures users act within their permissions. Failures can lead to unauthorized data access, modification, or destruction.



Broken Access Control

All resources on your site need to have access control implemented, even if they aren't intended to be discoverable by a user.

Learn About This Vulnerability →

Directory Traversal

Ensure file paths are safely interpreted, or hackers can access sensitive files on your server.

Learn About This Vulnerability →

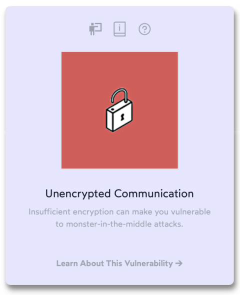
Cross-Site Request Forgery

If an attacker can forge HTTP requests to your site, they may be able to trick your users into triggering unintended actions.

Learn About This Vulnerability →

3.2.2 Cryptographic Failures

Cryptographic failures occur when sensitive data is not properly protected with strong encryption. This can lead to data theft, fraud, and other crimes. Protection methods include encrypting data at rest and in transit using modern encryption algorithms.



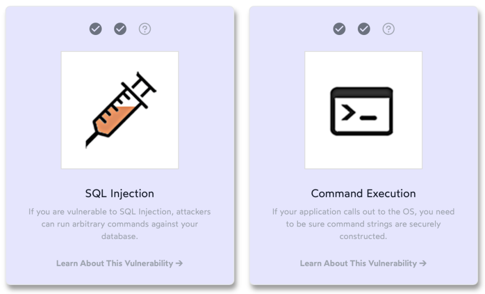
Unencrypted Communication

Insufficient encryption can make you vulnerable to monster-in-the-middle attacks.

Learn About This Vulnerability →

3.2.3 Injection

Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. This can lead to unauthorized data access or execution of unintended commands. Protection methods include using parameterized queries and input validation.



SQL Injection

If you are vulnerable to SQL Injection, attackers can run arbitrary commands against your database.

Learn About This Vulnerability →

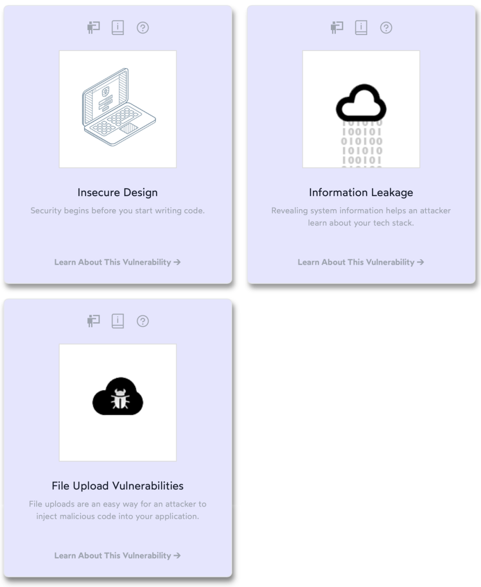
Command Execution

If your application calls out to the OS, you need to be sure command strings are securely constructed.

Learn About This Vulnerability →

3.2.4 Insecure Design

Insecure design refers to the lack of security considerations during the design phase of software development. This can lead to vulnerabilities that are difficult to fix later. Protection methods include gathering security requirements, threat modeling, and regular security testing.



Insecure Design

Security begins before you start writing code.

Learn About This Vulnerability →

Information Leakage

Revealing system information helps an attacker learn about your tech stack.

Learn About This Vulnerability →

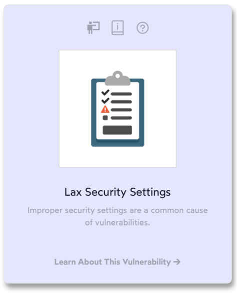
File Upload Vulnerabilities

File uploads are an easy way for an attacker to inject malicious code into your application.

Learn About This Vulnerability →

3.2.5 Security Misconfiguration

Security misconfiguration occurs when software is not securely configured, leading to vulnerabilities such as default accounts, open cloud storage, and verbose error messages. Protection methods include secure configuration standards, regular patching, and timely upgrades.



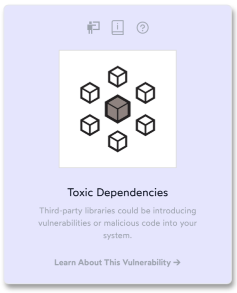
Lax Security Settings

Improper security settings are a common cause of vulnerabilities.

Learn About This Vulnerability →

3.2.6 Vulnerable and Outdated Components

Using vulnerable or outdated components, such as libraries and frameworks, can introduce security risks. Protection methods include regularly updating components and using tools to identify known vulnerabilities.



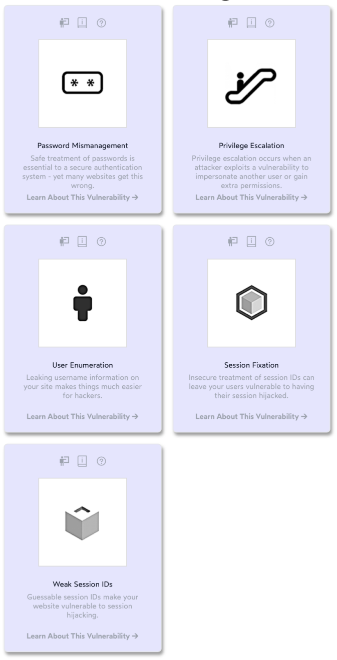
Toxic Dependencies

Third-party libraries could be introducing vulnerabilities or malicious code into your system.

Learn About This Vulnerability →

3.2.7 Identification and Authentication Failures

Failures in identification and authentication can lead to compromised passwords, session tokens, and user identities. Protection methods include strong password policies, multi-factor authentication, and secure session management.



Password Mismanagement

Safe treatment of passwords is essential to a secure authentication system - yet many websites get this wrong.

Learn About This Vulnerability →

Privilege Escalation

Privilege escalation occurs when an attacker exploits a vulnerability to impersonate another user or gain extra permissions.

Learn About This Vulnerability →

User Enumeration

Leaking username information on your site makes things much easier for hackers.

Learn About This Vulnerability →

Session Fixation

Insecure treatment of session IDs can leave your users vulnerable to having their session hijacked.

Learn About This Vulnerability →

Weak Session IDs

Guessable session IDs make your website vulnerable to session hijacking.

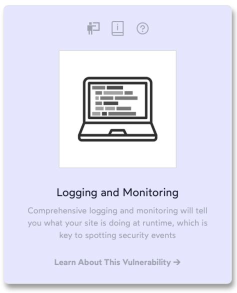
Learn About This Vulnerability →

3.2.8 Software and Data Integrity Failures

Software and data integrity failures occur when code and infrastructure do not protect against integrity violations. This can lead to unauthorized access and malicious code execution. Protection methods include using trusted sources, verifying updates, and securing the deployment pipeline.

3.2.9 Security Logging and Monitoring Failures

Insufficient logging and monitoring can allow attackers to go undetected for long periods. Protection methods include comprehensive logging, real-time monitoring, and effective incident response integration.

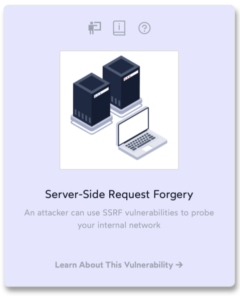


Logging and Monitoring

Comprehensive logging and monitoring will tell you what your site is doing at runtime, which is key to spotting security events

3.2.10 Server-Side Request Forgery

Server-Side Request Forgery (SSRF) occurs when a web application fetches a remote resource without validating the user-supplied URL. This can lead to unauthorized requests to internal systems. Protection methods include validating and sanitizing URLs and implementing network access controls.



Server-Side Request Forgery

An attacker can use SSRF vulnerabilities to probe your internal network

4 Lab Completion

In the end of Laboratory work, I have learned about the most common web application vulnerabilities and how to protect against them. I have explored the Hacksplaining Security Training for Developers and the OWASP Top 10 web application security risks. I have gained valuable insights into the importance of secure coding practices and the need for continuous security awareness.