TASK 4

1. FIND 3 WEBSITES VULNERABLE TO DIRECTORY/PATH TRAVERSAL VULNERABILITY

2. FIND 2 WEBSITES VULNERABLE TO FILE UPLOAD VULNERABILITY

3. FIND 1 WEBSITE VULNERABLE TO PARAMETER TAMPERING VULNERABILITY

1.Directory / Path Traversal vulnerability:

SUMMARY:

Directory traversal vulnerability/path traversal vulnerability: Arises in websites which takes user inputs and finds files with that user input.

2 types:

1. LFI-LOCAL FILE INCLUSIN

2. RFI-REMOTE FILE INCLUSION

.../-directorial traversal Folders in Linux:/root/desktop/

IMPACT: critical vulnerability Include function in PHP is used to retrieve files from the given directory, if they include does not validate the user input for filename or directory i.e., not restricting the user access to sensitive folders then we can grab the files and information easily for the directory we give.

Payloads Used:

./../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../etc/passwd

.../../../../../etc/shadow - has hashed passwords of Linux user account

.../../../../../etc/hosts - gives respective servers of the given Ip address

STEPS TO REPRODUCE VULNERABILITY:

1. Open the vulnerable website

2. Go to URL

3. Replace the parameter and parameter with = with the payload previously mentioned.

4. Press "Enter”.

5. If the files from that website are relived, local files are shown together with the user shell.

6. As a result, the website is exposed.

Websites:

1. <https://m.dressfair.pk/index.php?route=product/special&sort=p.model&order=DESC&page=23>

Payload used:

./../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../etc/passwd



2.

<https://fotoaz.ch/gallery.php?OPEN=Eltern>

A screenshot of a website

Description automatically generated with low confidence

REMEDIAL ACTIONS:

The website should validate the input parameter or value before processing it if that’s not possible.

the input values must be filtered first before accepting them.

After validating it should verify the path before displaying the output

The permissions must be given so that only the authorized person can gain the access.

Do not use user input to directly call a file.

The data should not be interpreted, it should be encoded, escaped, and cleaned.

File Upload Vulnerability:

Web apps have file upload functionality.

The server should verify the files that are uploaded are not executable files.

This means that the server should prevent the user to upload executable files.

The validation of file extension should be done on both client and server end.

Impact:

File upload vulnerability leads to remote code execution.

File upload vulnerability can also lead to DOS attack.

File upload vulnerability leads to defacement of the website or webservers.

Broken access control which leads to file upload vulnerability.

Google dorks used:

inurl :/assets/filemanager/

index of admin/fckeditor/editor/filemanager/

STEPS TO REPRODUCE VULNERABILITY:

1. First open burp suite browser

2. Upload the php/html file instead of jpg file

3. Turn on the intercept

4. Go to response

5. If the response is 200 OK, then it is implemented to be accepted

6. If the website accepts only these extensions that is non-executable extensions

7. Then it’s not vulnerable

8. If it accepts executable extension files, then its vulnerable

1.<http://home.deds.nl/~piejoo/test/admin/fck/editor/filemanager/browser/default/>

A screenshot of a computer

Description automatically generated with medium confidence

Frmupload.html:

A picture containing text, screenshot, envelope, stationary

Description automatically generated

Burp suite request:

A screenshot of a computer

Description automatically generated with medium confidence

Now we upload 2.php as a txt file in it.

A picture containing text, screenshot, envelope, stationary

Description automatically generated

Burp suite request:

A screenshot of a computer

Description automatically generated

As we can see the webpage has accepted the 2.php and remote file execution can be performed.

REMEDIAL ACTIONS:

File type verification must be done before accepting the file.

Restrict the file extension specifications to avoid potentially malicious content to your site.

Malware prevention with multiple anti-malware tools

User authentication to validate the identity of the person requesting the private information.

Store the files in external directory to prevent attackers from executing malicious files through a website URL.

PARAMETER TAMPERING:

SUMMARY:

If the parameter values are modified and the server does not validate or compare the actual values

and proceed with the fake values, then it leads to this vulnerability.

Proxy is used for this, and the parameter values are changed.

The manipulated http request is then forwarded, but if the server does not compare the amount in

the parameter’s values with the actual amount and proceeds with the payment then the amount payable will be different.

STEPS TO REPRODUCE THE VULNERABILITY:

1. OPEN THE WEBSITE

2. Give the email and click on submit

3. Intercept with proxy

4. Replace the email with payload

5. Forward the http post request with modified email parameter value

6. 200 ok implies the request is accepted

7. Therefore, proving the vulnerability

Impact:

Financial loss, account take over, enumeration other user’s private data, sensitive data.

Leakage/disclosure, privilege escalation, violation of private policy.

Remedial actions:

Use a whitelist format for the application’s inputs, web application firewalls for utmost protection.

Encrypt the session cookies to prevent tampering, avoid including parameters into the query string.

Implementing role-based controls to check what permissions the user has is an effective way to determine.