**FILE UPOLAD**

Uploaded files represent a significant risk to web applications. The first step in many attacks is to get some code to the system to be attacked. Then the attacker only needs to find a way to get the code executed. Using a file upload helps the attacker accomplish the first step.

The consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system, forwarding attacks to backend systems, and simple defacement. It depends on what the application does with the uploaded file, including where it is stored.

**Objective**

Execute any PHP function of your choosing on the target system (such as phpinfo() or system()) thanks to this file upload vulnerability.

**Description**

The File Upload component is used to upload files from a remote machine to a specified location on the Web server. The component can be used to allow users to upload files to the server.

For example: The File Upload component can be added to a Record form or an Editable Grid

**Impact:**

The impact of this vulnerability is high, supposed code can be executed in the server context or on the client side. The likelihood of detection for the attacker is high.

Client-side attacks: Uploading malicious files can make the website vulnerable to client-side attacks such as XSS or Cross-site Content Hijacking.

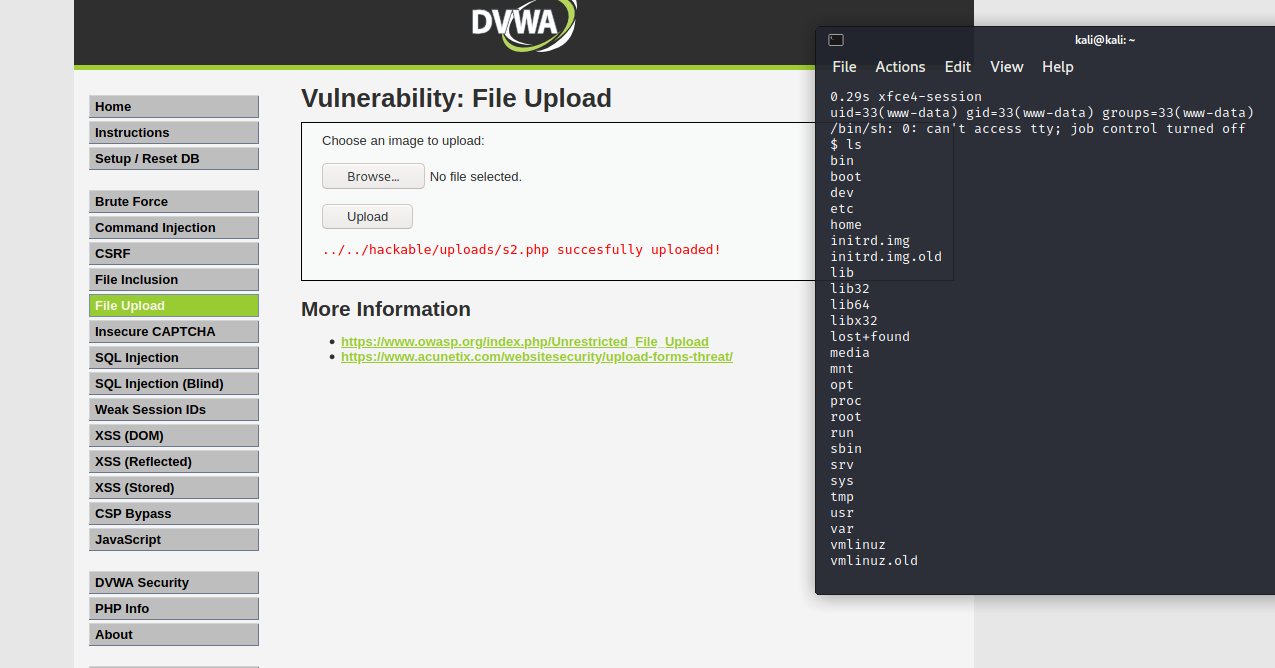
**Prevention:**

1. Only allow specific file extensions.
2. Only allow authorized and authenticated users to use the feature.
3. Check any file fetched from the Web for content. Make sure it is actually an image or whatever file type you expect.
4. Serve fetched files from your application rather than directly via the web server.
5. Store files in a non-public accessibly directory if you can.
6. Write to the file when you store it to include a header that makes it non-executable.

**LOW**

**Steps to reproduce:**

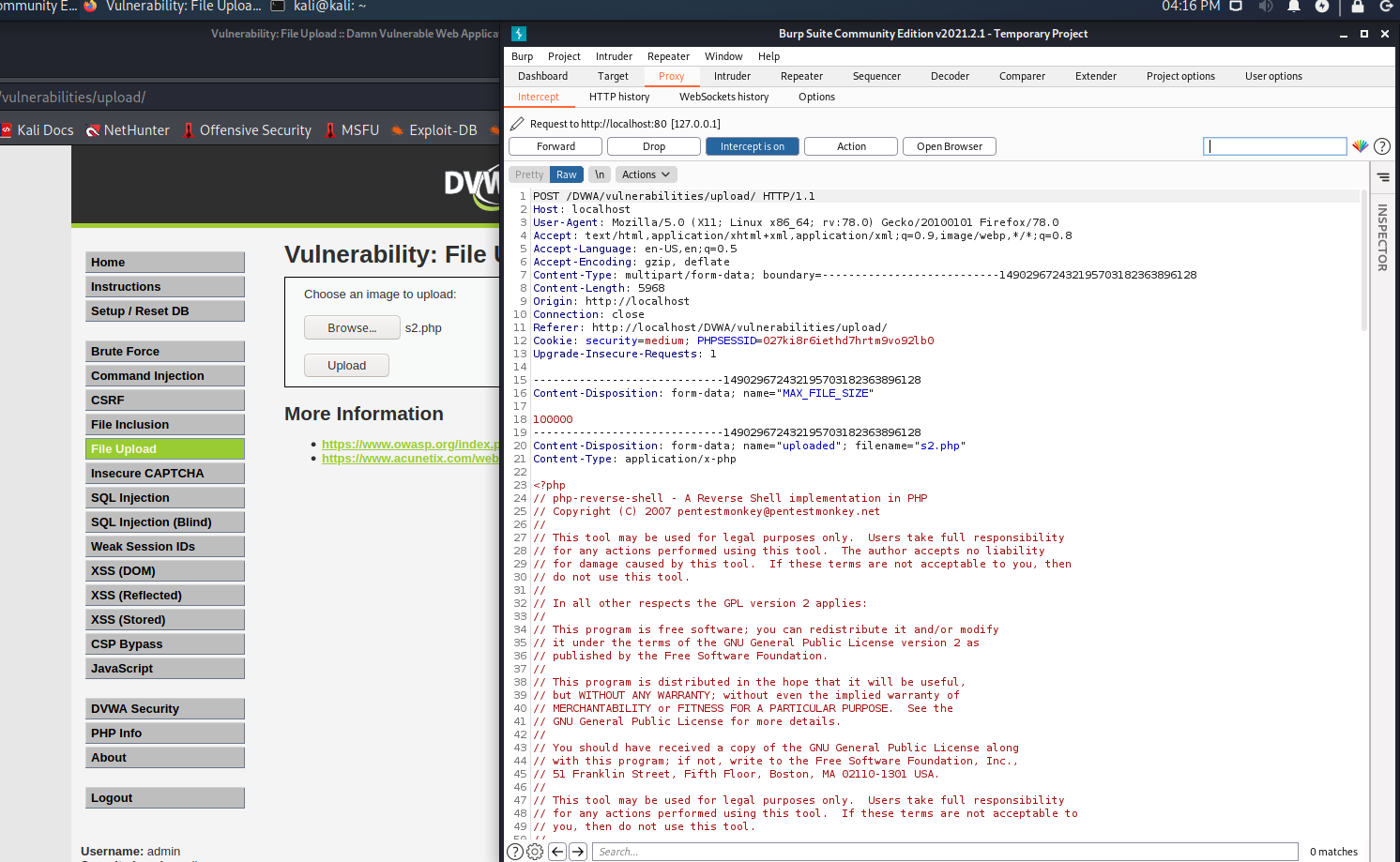
1. Configure your browser.
2. Go to the dvwa page and set level of file upload to the low level.
3. Create reverse shell
4. Open terminal and type command ‘ locateshell.php ‘ we will use this shell ‘/usr/share/webshells/php/php-reverse-shell.php ‘
5. Copy this shell to desktop and rename this file to s2.php.
6. Open this file and change ip and port in this file and save this.
7. In ip enter your won ip and in port enter that port number which was used in reverse connection
8. Now transfer this file to the folder /var/www/html and we can access via ‘http://127.0.0.1/s2.php’
9. In dvwa a web page click on the browser and select s2.php file and click on the upload.
10. The file path for s2.php is “../../hackable/uploads/s2.php”
11. Start ncat listener and replace # with “../../hackable/uploads/s2.php” in the url.
12. Press enter and we got access to the web server In our terminal.



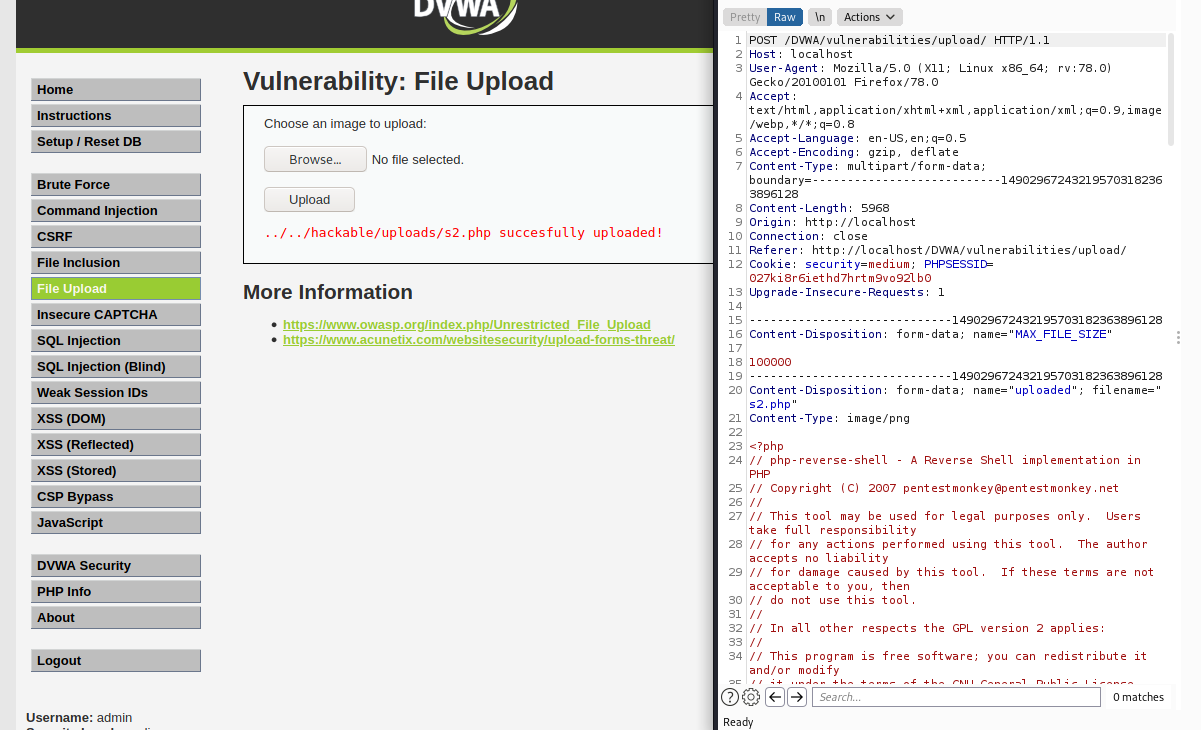
**MEDIUM**

**Steps to reproduce:**

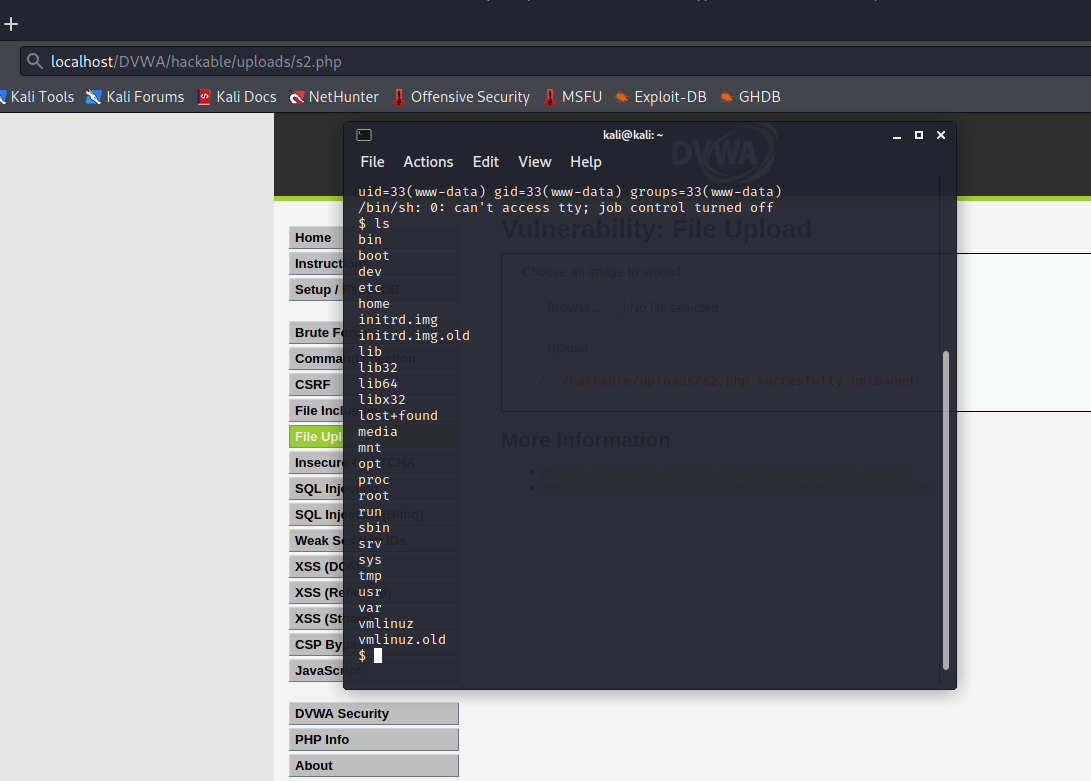
1. Configure your browser and configure your burp suite.
2. Go to the dvwa page and set level of file upload to the medium level.
3. Create reverse shell
   * + 1. Open terminal and type command ‘ locateshell.php ‘ we will use this shell ‘/usr/share/webshells/php/php-reverse-shell.php ‘
       2. Copy this shell to desktop and rename this file to s2.php.
       3. Open this file and change ip and port in this file and save this.
       4. In ip enter your won ip and in port enter that port number which was used in reverse connection
       5. Now transfer this file to the folder /var/www/html and we can access via ‘http://127.0.0.1/s2.php’
4. Now click on the browser and upload s2.php file and capture the request in the tool like burp suite.



1. In burp suite we have to change content type because we saw earlier that only image/png type can upload php files and forward the request.



1. File is upload successfully.
2. The file path for s2.php is “../../hackable/uploads/s2.php”
3. Start ncat listener and replace # with “../../hackable/uploads/s2.php” in the url.
4. Press enter and we got access to the web server In our terminal.



**HIGH**

**Steps to reproduce:**

1. Configure your browser.
2. Go to the dvwa page and set level of file upload to the high level.
3. Create reverse shell.

Open terminal and type command ‘ locateshell.php ‘ we will use this shell ‘/usr/share/webshells/php/php-reverse-shell.php ‘

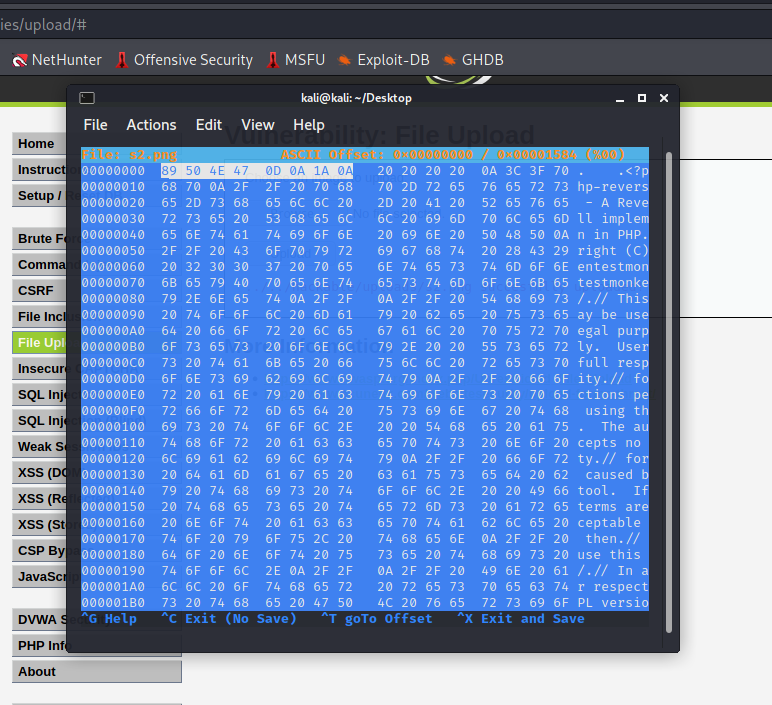
Copy this shell to desktop and rename this file to s2.php.

Open this file and change ip and port in this file and save this.

In ip enter your won ip and in port enter that port number which was used in reverse connection

Now transfer this file to the folder /var/www/html and we can access via ‘http://127.0.0.1/s2.php’

1. Rename the s2.php file to s2.png because page will not support php file.
2. Still not upload now open the s2.png file and give 8 free space in first line.
3. Now change the file signature of png file from haxeditor and save it.



1. We can change the name file to s2.php for execution.
2. For this we use command injection tab for copy the file through this command “127.0.0.1|cp ../../hackable/uploads/s2.png ../../hackable/uploads/s2.php”.
3. Start ncat listener and replace # with “../../hackable/uploads/s2.php” in the url.
4. Press enter and we got access to the web server In our terminal.

