

MAN IN THE MIDDLE ATTACK

Step 1 – Setup Environment (Use Internet Initially)

connect your Kali Linux laptop to the internet (only for setup).

Step 2 – Install Dependencies

```
sudo apt update
```

```
sudo apt install python3 python3-venv
```

Step 3 – Create Virtual Environment

```
python3 -m venv mitm-env
```

```
source mitm-env/bin/activate
```

Step 4 – Install mitmproxy and Fix bcrypt Issue

```
pip install mitmproxy bcrypt==4.0.1
```

The screenshot shows a terminal window on a Kali Linux desktop. The terminal history includes:

- \$ sudo apt install python3 python3-venv
- [sudo] password for kali:
- python3 is already the newest version (3.13.5-1).
- python3-venv is already the newest version (3.13.5-1).
- Summary:
- Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1545
- \$ python3 -m venv mitm-env
- source mitm-env/bin/activate
- (mitm-env)–(kali㉿kali)-[~]
- \$ pip install mitmproxy bcrypt==4.0.1
- Requirement already satisfied: mitmproxy in ./mitm-env/lib/python3.13/site-packages (12.1.2)
- Requirement already satisfied: bcrypt==4.0.1 in ./mitm-env/lib/python3.13/site-packages (4.0.1)
- Requirement already satisfied: aioquic<=1.2.0, >=1.2.0 in ./mitm-env/lib/python3.13/site-packages (from mitmproxy) (1.2.0)
- Requirement already satisfied: argon2-cffi<=25.1.0, >=23.1.0 in ./mitm-env/lib/python3.13/site-packages (from mitmproxy) (25.1.0)
- Requirement already satisfied: asigref<=3.9.1, >=3.2.10 in ./mitm-env/lib/python3.13/site-packages (from mitmproxy) (3.9.1)

Step 5 – Turn OFF Internet

Ensure everything works fully offline.

Step 6 – Create server.py (Simple HTTP Server)

Open a text editor (like nano) and create server.py:

```
nano server.py
```

Paste this code inside server.py:

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import urllib.parse

class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
```

```
def do_GET(self):
    self.send_response(200)
    self.end_headers()
    self.wfile.write(b"""
        <html>
            <body>
                <h2>Login Page</h2>
                <form method="POST">
                    Username: <input type="text" name="username"><br>
                    Password: <input type="password" name="password"><br>
                    <input type="submit" value="Login">
                </form>
            </body>
        </html>
    """)

def do_POST(self):
    content_length = int(self.headers['Content-Length'])
    post_data = self.rfile.read(content_length)
    form_data = urllib.parse.parse_qs(post_data.decode('utf-8'))

    username = form_data.get('username')[0]
    password = form_data.get('password')[0]

    print(f"[+] Intercepted Credentials -> Username: {username}, Password: {password}")

    self.send_response(200)
    self.end_headers()
    self.wfile.write(b"<html><body><h2>Login Successful</h2></body></html>")

server_address = ('0.0.0.0', 8080)
```

```
httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
print("[*] HTTP Server running on http://<local-IP>:8080")
httpd.serve_forever()

Save & exit (Ctrl + O, Enter, Ctrl + X).
```

Step 7 – Create intercept.py (mitmproxy Script)

Create the script file:

```
nano intercept.py
```

Paste this code:

```
from mitmproxy import http
```

```
def request(flow: http.HTTPFlow) -> None:
    if flow.request.method == "POST":
        print("\n[+] Intercepted POST Data:")
        print(flow.request.content.decode())
```

Save & exit (Ctrl + O, Enter, Ctrl + X).

Step 8 – Terminal Setup and Commands

Terminal Action

- ◆ Terminal 1 Run HTTP Server:
python3 server.py
- ◆ Terminal 2 Run mitmproxy with intercept script:
mitmproxy --mode regular -p 8081 --listen-host 0.0.0.0 -s intercept.py
- ◆ Browser Configure proxy:
HTTP Proxy → 127.0.0.1, Port → 8081
Then visit: http://192.168.X.X:8080 and submit the form

Step 9 – Finding Local IP Address

In any terminal:

```
ip a
```

Example result:

```
inet 192.168.43.10/24
```

Use this IP when opening the browser:

<http://192.168.43.10:8080>

```

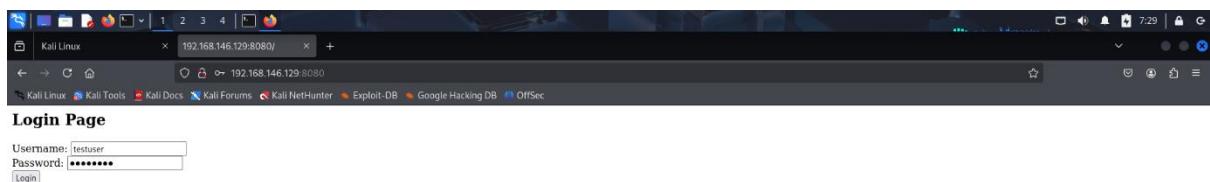
mitm-env:kali:~] $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:21:8f:2e brd ff:ff:ff:ff:ff:ff
    inet 192.168.146.129/24 brd 192.168.146.255 scope global dynamic noprefixroute eth0
        valid_lft 1752sec preferred_lft 1752sec
    inet6 fe80::c7c4:95da:4ee2:707c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:4c:23:9b:3c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

mitm-env:kali:~] $ python3 server.py
[*] HTTP Server running on http://<local-IP>:8080

```

Step 10 – Run the Demo

1. Open browser → Visit <http://192.168.43.10:8080>
2. Submit dummy username & password (e.g., testuser / mypassword).
3. Observe:
 - o mitmproxy terminal prints intercepted POST data.
 - o HTTP server terminal prints credentials.



Step 11 – Cleanup After Demo

- Stop HTTP Server:
- Ctrl + C
- Stop mitmproxy:

- Ctrl + C
 - Deactivate virtual environment:
 - deactivate
-

Full Command Summary

```
# Install Dependencies (once)
```

```
sudo apt update
```

```
sudo apt install python3 python3-venv
```

```
# Create Virtual Env
```

```
python3 -m venv mitm-env
```

```
source mitm-env/bin/activate
```

```
# Install mitmproxy
```

```
pip install mitmproxy bcrypt==4.0.1
```

```
# (Turn OFF Internet)
```

```
# Find Local IP
```

```
ip a
```

```
# Run HTTP Server (Terminal 1)
```

```
python3 server.py
```

```
# Run mitmproxy with script (Terminal 2)
```

```
mitmproxy --mode regular -p 8081 --listen-host 0.0.0.0 -s intercept.py
```

```
# Configure Browser Proxy: 127.0.0.1:8081
```

```
# Visit in Browser: http://<local-IP>:8080
```

```
# Submit dummy credentials
```

```
# Stop everything when done:
```

```
Ctrl + C
```

```
Deactivate
```

OUTPUT

The terminal window shows the following session:

```
(mitm-env)kali@kali:~$ python3 server.py
[*] HTTP Server running on http://<local-IP>:8080
^CTraceback (most recent call last):
  File "/home/kali/server.py", line 38, in <module>
    httpd.serve_forever()
  ~~~~~^C
  File "/usr/lib/python3.13/socketserver.py", line 235, in serve_forever
    ready = selector.select(poll_interval)
  File "/usr/lib/python3.13/selectors.py", line 398, in select
    fd_event_list = self._selector.poll(timeout)
KeyboardInterrupt

(mitm-env)kali@kali:~$ python3 server.py
[*] HTTP Server running on http://<local-IP>:8080
192.168.146.129 - - [16/Sep/2025 07:29:28] "GET / HTTP/1.1" 200 -
[+] Intercepted Credentials → Username: testuser, Password: password
192.168.146.129 - - [16/Sep/2025 07:29:49] "POST / HTTP/1.1" 200 -
```

The terminal then switches to a NetworkMiner-like interface showing a captured POST request:

Flow Details

Request	Response	Detail
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Content-Type: application/x-www-form-urlencoded Content-Length: 35 Origin: http://192.168.146.129:8080 Connection: keep-alive Referer: http://192.168.146.129:8080/ Upgrade-Insecure-Requests: 1 Priority: u=0, i URLEncoded form username: testuser password: password	Firefox/128.0 200 OK 51b 8ms	[*:auto]

Flow: e Edit D Duplicate r Replay x Export d Delete b Save body
Proxy: ? Help q Back E Events O Options i Intercept f Filter

[*:8081]