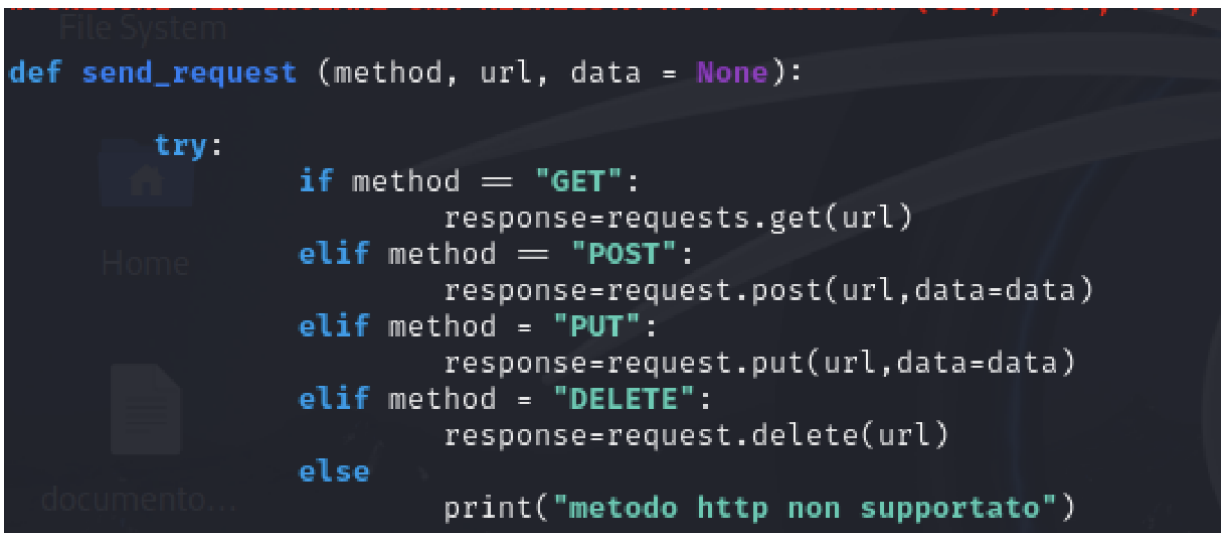


REPORT PROGRAMMA IN PYTHON PER INVIARE RICHIESTE HTTP (GET, POST, PUT, DELETE) AL WEB SERVER E VERIFICARE LE RISPOSTE:

La libreria requests è una libreria usata in Python per inviare facilmente richieste HTTP. Con questa libreria, possiamo fare tutte le operazioni necessarie (GET, POST, PUT, DELETE).

Successivamente inserisci l'indirizzo del server DVWA al quale inviare le richieste. Si presume che tu abbia una copia di DVWA in esecuzione sul tuo computer locale (localhost). Se il server DVWA è su una macchina diversa o su una porta diversa, dovrai modificare questo URL.



```
def send_request (method, url, data = None):  
    try:  
        if method == "GET":  
            response=requests.get(url)  
        elif method == "POST":  
            response=request.post(url,data=data)  
        elif method == "PUT":  
            response=request.put(url,data=data)  
        elif method == "DELETE":  
            response=request.delete(url)  
        else  
            print("metodo http non supportato")
```

Questa funzione è la parte più importante del programma. Gestisce tutte le richieste HTTP (GET, POST, PUT, DELETE) in modo unificato, evitando di scrivere codice duplicato per ogni tipo di richiesta.

Method: Una stringa che indica il tipo di richiesta HTTP da inviare. Può essere "GET", "POST", "PUT" o "DELETE".

Url: L'indirizzo del server al quale inviare la richiesta.

data: (Opzionale) I dati da inviare nel corpo della richiesta, utilizzato solo per i metodi POST e PUT. Per esempio, nel caso di POST, potresti voler inviare un nome utente e una password.

In base al tipo di richiesta (method):

Se il method è "GET", la funzione invia una richiesta GET al server con `requests.get(url)`.

Se il method è "POST", invia una richiesta POST con `requests.post(url, data=data)`, includendo i dati passati come data.

Se il method è "PUT", invia una richiesta PUT con `requests.put(url, data=data)`, includendo i dati passati come data.

Se il method è "DELETE", invia una richiesta DELETE con `requests.delete(url)`.

Se il method non è uno di quelli supportati, la funzione genera un errore tramite `raise ValueError("Metodo HTTP non supportato")`.

```
#STAMPA LA RISPOSTA
def print_response(response):
    print("\n[method] Request:")
    print(f"status code: {response.status_code}")
    print(f"response text: {response.text[:200]} ... ") #MOSTRA SOLO I PRIMI 200 CARATTERI

except requests.exceptions.RequestException as e:
    print(f"errore nella richiesta {method}:{e}")

#FUNZIONE PRINCIPALE
def main():
    #INVIO DELLE RICHIESTE CON I METODI HTTP
    send_request("GET", url)
    send_request("POST", url, data={"username": "admin", "password": "password"})
    send_request("PUT", url, data={"key": "value"})
    send_request("DELETE", url)

main()

```

Una volta inviata la richiesta, il programma stampa:

Il codice di stato HTTP (`response.status_code`), che indica se la richiesta ha avuto successo (ad esempio, 200 per OK).

I primi 200 caratteri della risposta del server (`response.text[:200]`), per dare un'idea di cosa è stato restituito senza stamparlo tutto.

La funzione `main()` è dove vengono effettivamente inviate le richieste HTTP. All'interno di questa funzione, chiamiamo la funzione `send_request` per ogni tipo di richiesta (GET, POST, PUT, DELETE) con i dati appropriati.

Dettaglio:

- `send_request("GET", url)`: Invia una richiesta GET al server (recupera la pagina o le informazioni).
- `send_request("POST", url, data={"username": "admin", "password": "password"})`: Invia una richiesta POST al server con dei dati (in questo caso, un nome utente e una password).
- `send_request("PUT", url, data={"key": "value"})`: Invia una richiesta PUT per aggiornare o modificare qualcosa sul server.
- `send_request("DELETE", url)`: Invia una richiesta DELETE per eliminare qualcosa dal server.