Il codice innanzitutto utilizza i file contenenti i **nomi utente** e le **password** più comuni (già presenti su Kali Linux) per creare due liste, una con i nomi utente e una con le password (si noti come invertiamo i primi due elementi della lista dei nomi utente per far stare *admin* in cima, che sappiamo già essere usato come nome utente nel nostro ambiente, rendendo più veloce l'esecuzione del codice per testarlo).

È necessario per prima cosa effettuare il login nella **home** della DVWA.

Per ogni nome utente si tenta il login con tutte le password possibili (2 cicli **for** annidati), creando delle sessioni e lanciando delle richieste **POST** per l'accesso, indicando l'url della pagina e i dati per il login nei parametri; in base al messaggio stampato in caso di login fallito siamo in grado di capire se il login è avvenuto con successo, interrompendo i cicli e visualizzando il nome utente e la password corretti.

Ora si può usare la **sessione** per effettuare altre richieste autenticate (grazie alle sessioni di python vengono automaticamente memorizzati i cookie di sessione).

Per prima cosa scegliamo il **livello di sicurezza** della DVWA mandando una richiesta post che lo aggiorna per la sessione corrente (è necessario farlo poiché ad ogni sessione viene resettata su high di default).

Per entrare nella sezione **brute force** della DVWA la logica per i tentativi di login è la stessa, con una importante differenza. Analizzando i pacchetti con **Burp Suite** e interagendo con la pagina di login di tale sezione notiamo che, al contrario di come comunemente si fa, per il login non viene utilizzato il metodo POST bensì il **GET**. Adattiamo il codice per questa situazione, ma poniamo l'attenzione sul fatto che i dati per il login vengono concatenati all'url per la richiesta e questa scelta non è assolutamente sicura.

Infine visualizziamo la combinazione corretta di nome utente e password anche per questa sezione.