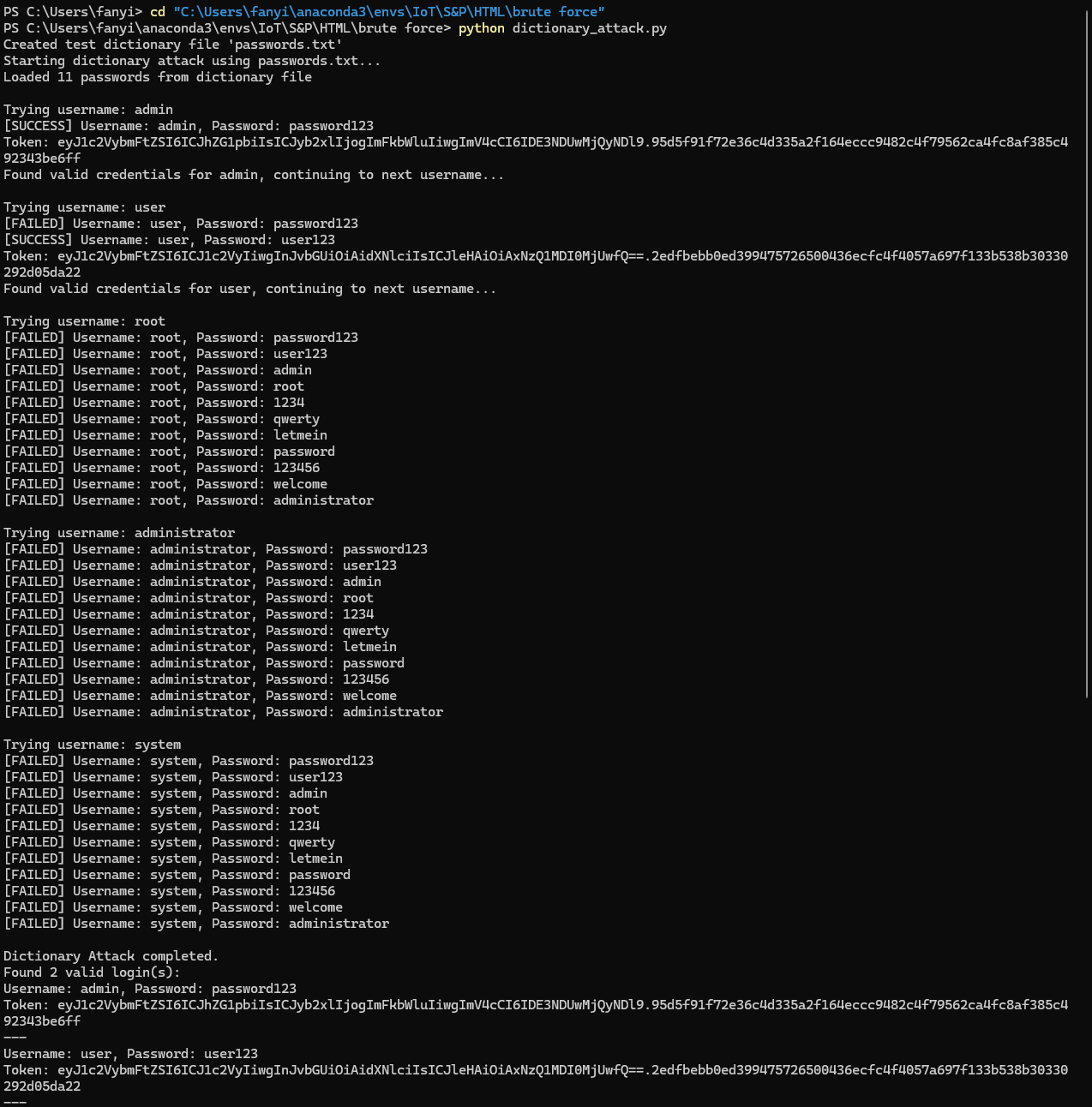
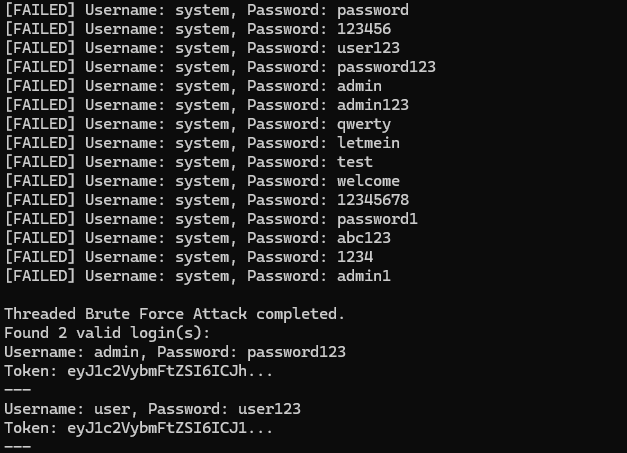
Dictionary Attack:



Threaded Attack:  

### **1. Dictionary Attack (dictionary\_attack.py)**

This script performs a **dictionary-based brute force attack** on a login API (/api/login). The attacker iterates over a list of common usernames and tries passwords from a dictionary file (or a built-in list if the file is missing). The attack is **sequential**, trying one username/password pair at a time.

* **Purpose**: To find valid login credentials by trying likely password guesses.
* **Method**:  
  + Tries each username with all passwords from the dictionary.
  + Sends a POST request for each combination.
  + On a successful login (detected via the response JSON), it logs the credentials and token.

### **2. Threaded Brute Force Attack (threaded\_attack.py)**

This script enhances the dictionary attack by introducing **concurrency** using Python threads, speeding up the process.

It offers two modes:

* **Queue-based threading**: A work queue holds username/password pairs, and multiple threads fetch from it and try logins concurrently.
* **Thread pool**: Uses ThreadPoolExecutor to manage a pool of worker threads and map login tasks.
* **Purpose**: To perform brute force attacks more efficiently by parallelizing login attempts.
* **Features**:  
  + Configurable thread count and delay.
  + Uses synchronization (Lock) to safely update shared state and print outputs.
  + Supports optional dictionary file input.