Brute Force Attack Defense System Using Java

1. Introduction

This project implements a defensive mechanism to protect a login endpoint against brute-force attacks.
Brute force attacks involve repeatedly trying different passwords until the correct one is found. To prevent this, I implemented a rate-limiter and automatic IP lockout mechanism.

2. Objective

- Detect multiple login attempts from the same user
- Block suspicious behavior
- Log all attempts for analysis
- Demonstrate cybersecurity controls in a safe environment

3. Tools Used

- Java
- CMD / PowerShell
- localhost server
- logs.txt for output proof

4. Defensive Approach

✓ IP-based attempt counting

✓ Threshold = 3 attempts

✓ Lockout = 30 seconds

✓ Detailed logging

✓ Safe, local-only server

5. Working

1. Server runs locally on port 8080
2. Every refresh of /login counts as a login
3. After 3 attempts → IP locked
4. Logs created automatically

PROCEDURE (Step-by-Step)

1. **Setting Up the Environment**
   - Install **Java JDK 8 or above** on your system.
   - Create a new folder on your Desktop named: brute defense_lab
   - Download or copy the file **DefenseServer.java** into this folder.
2. **Opening the Command Prompt**
   - Press **Win + R**
   - Type **cmd**
   - Press **Enter**
   - Navigate to the project folder using the command:

     cd "C:\Users\Hp\OneDrive\Desktop\brute defense_lab"
3. **Compiling the Defense Program**
   - In the command prompt, type: javac DefenseServer.java
   - This command compiles the Java file and checks for errors.
   - If no error appears, the program is successfully compiled.
4. Running the Defense Server

Run the server with: java

DefenseServer

You will see the message:

Defense Server running on http://127.0.0.1:8080/login

This means the defensive system is active and listening for login attempts.

5. Testing the Login Protection System

1. Open any browser (Chrome, Edge, Firefox).
2. In the address bar, enter:

http://127.0.0.1:8080/login

3. Each refresh of the page simulates a login attempt.
4. Observe the following output:

Attempt Number	**Browser Output**
1	Login attempt recorded. Attempt: 1
2	Login attempt recorded. Attempt: 2
3	Login attempt recorded. Attempt: 3
4+	Too many attempts! You are locked for 30 seconds.

6. Observing the Lockout Mechanism

- After **3 failed attempts**, the system automatically blocks further attempts.
- During lockout, the browser will show:

  LOCKED OUT. Try again in XX seconds.

  This demonstrates **successful brute-force detection + mitigation**.

7. Generating and Viewing Logs

- Inside the project folder, a file named **logs.txt** will be automatically created.

- Every login attempt and lockout event is recorded inside this file.
- Sample entries:

Fri Nov 24 - Login attempt from 127.0.0.1 | Count = 1

Fri Nov 24 - IP locked: 127.0.0.1

Fri Nov 24 - Blocked login from 127.0.0.1 (still locked)

This proves the system **detected**, **blocked**, and **logged** the attack attempts.

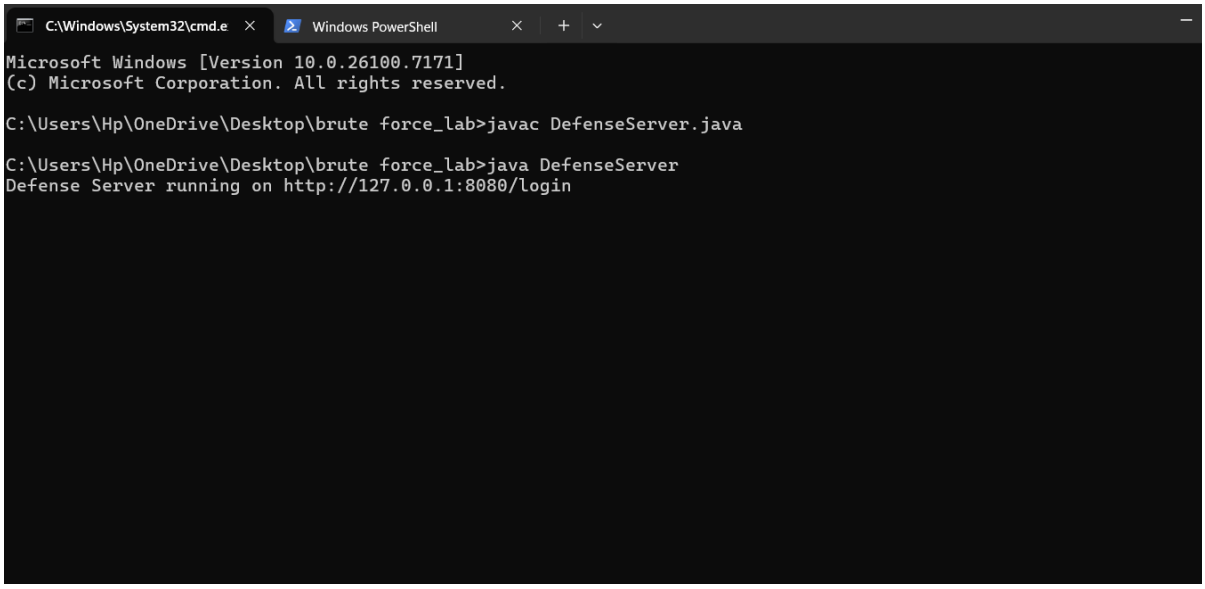
8. Stopping the Server

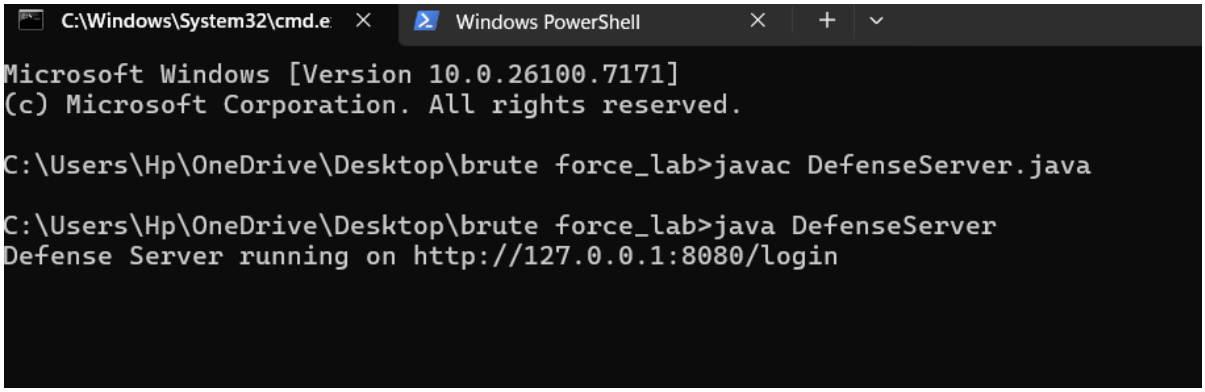
To stop the defense server:

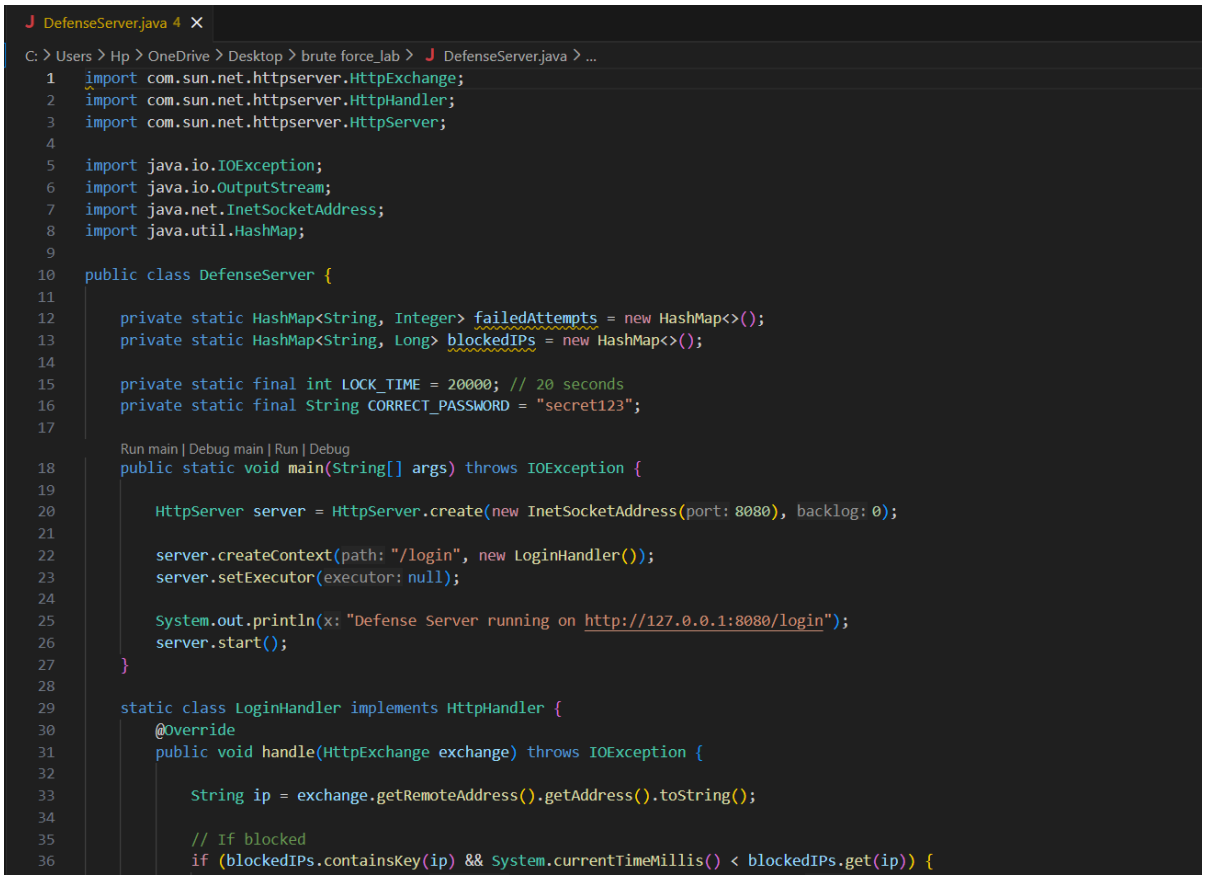
- Go to the command window where it is running
- Press **CTRL + C**

This terminates the program.

9. Output (Screenshots):







Conclusion:

The brute-force defense system successfully detects and blocks repeated login attempts from the same user. By counting login attempts, applying a lockout after the threshold, and logging all suspicious activity, the system demonstrates an effective real-world mitigation technique. The entire setup runs safely on localhost, making it suitable for laboratory use without causing any security risks. This implementation

fulfills the requirement of identifying the brute-force pattern, preventing further attempts, and providing clear evidence through logs, proving that the defense mechanism works correctly.