Information disclosure - Source code disclosure via backup files

Lab Report

Submitted By:

Name: Arsalan Khan
Position/Role: Internee
Date: July 25, 2025

Platform:

PortSwigger

Objective:

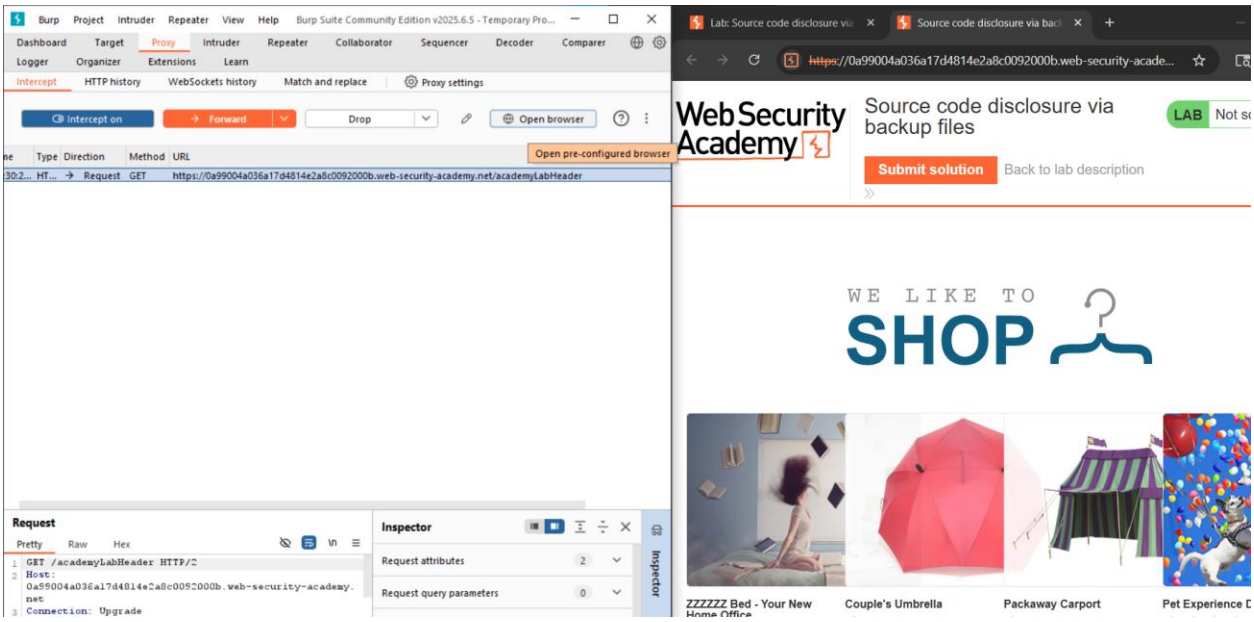
Discover a backup source code file exposed in a hidden directory, extract the hard-coded PostgreSQL database password, and submit it to solve the lab.

Tools Used:

- Burp Suite Community

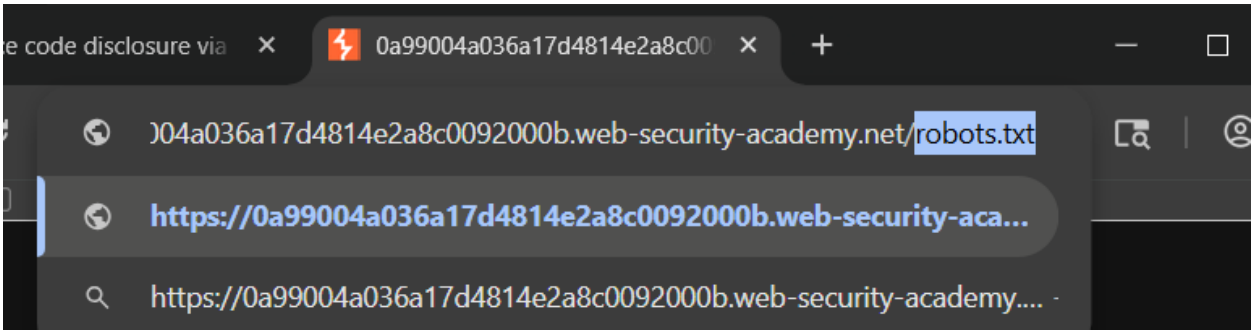
1. Access the Lab

- Opened the homepage to begin analysis.

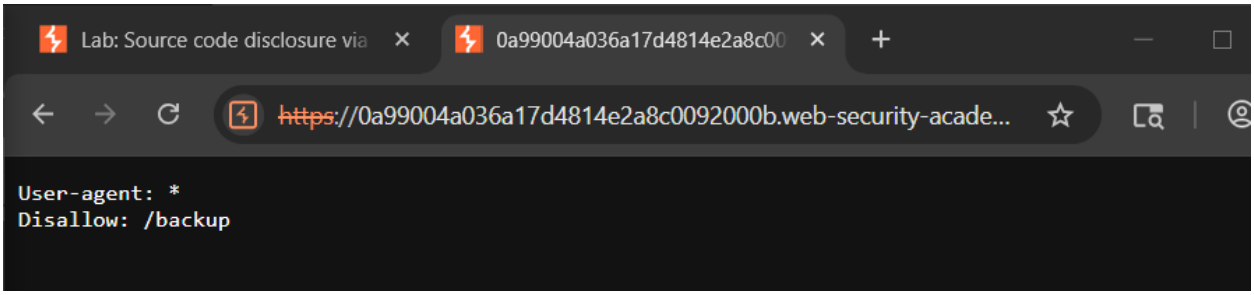


2. Check robots.txt for Hidden Paths

- Appended /robots.txt to the base URL:

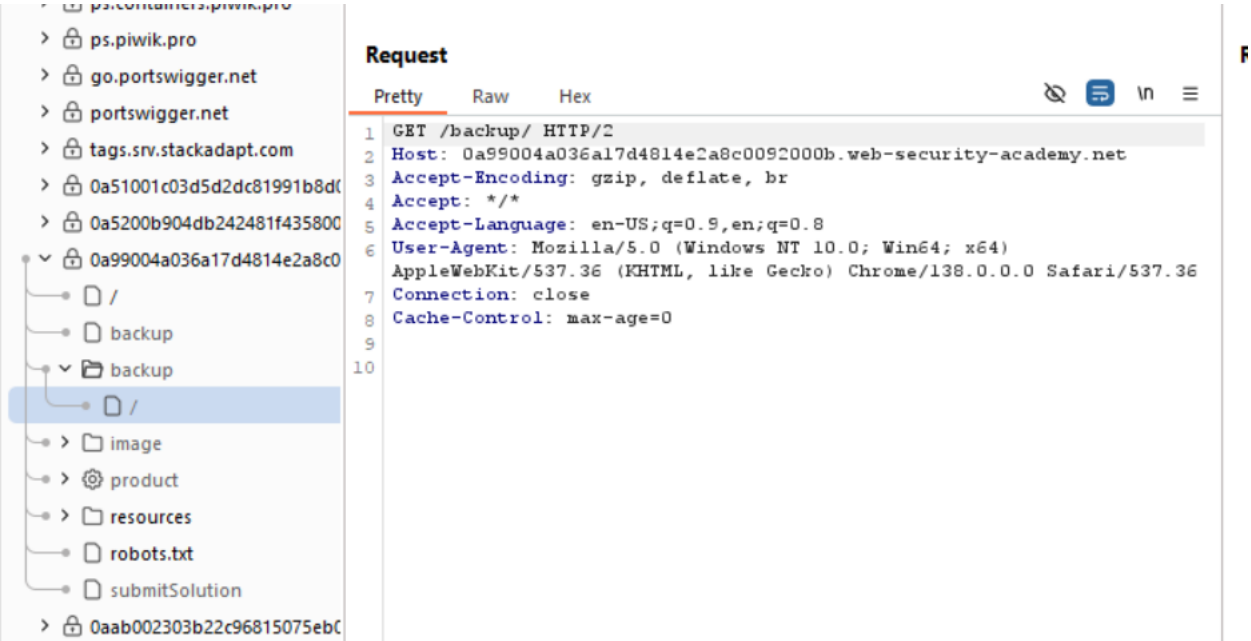


- Found the following entry in the response:



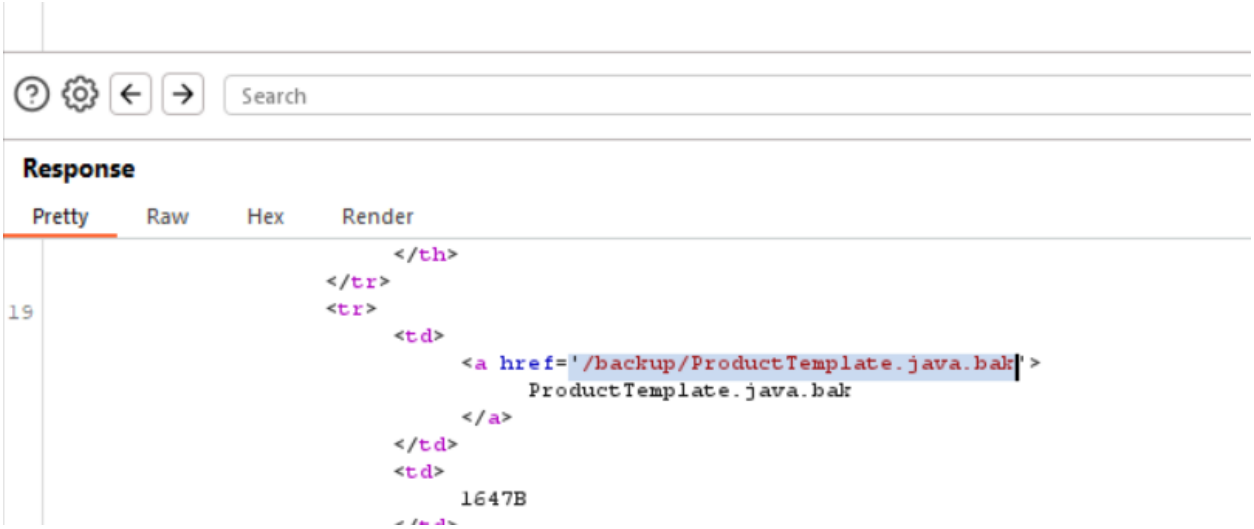
3. Explore the /backup Directory

- Navigated to /backup:

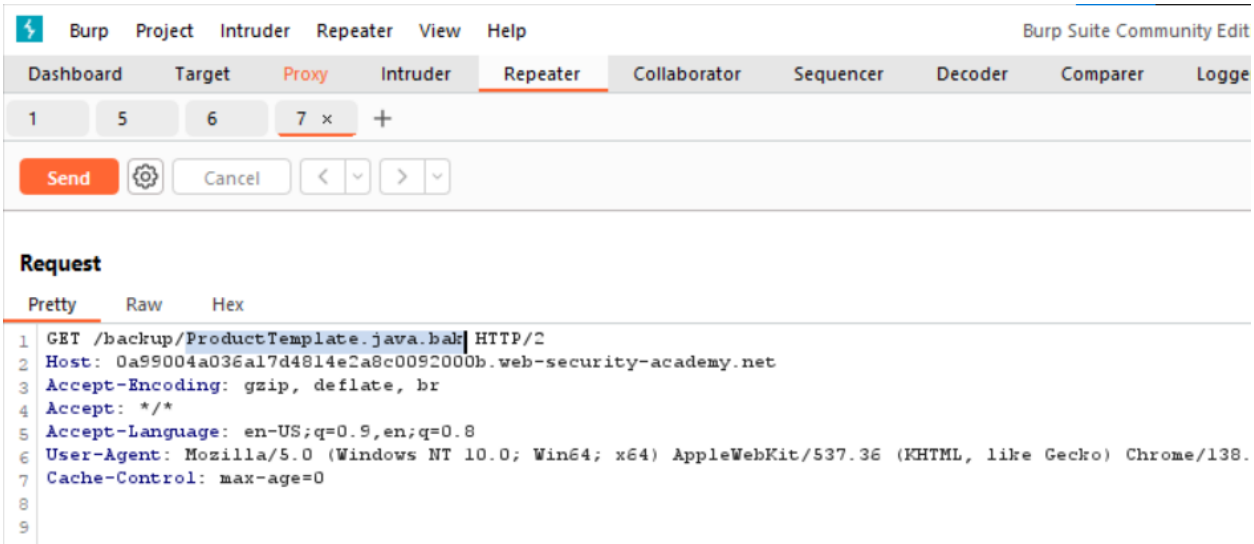


4. Send the Backup File Request to Repeater

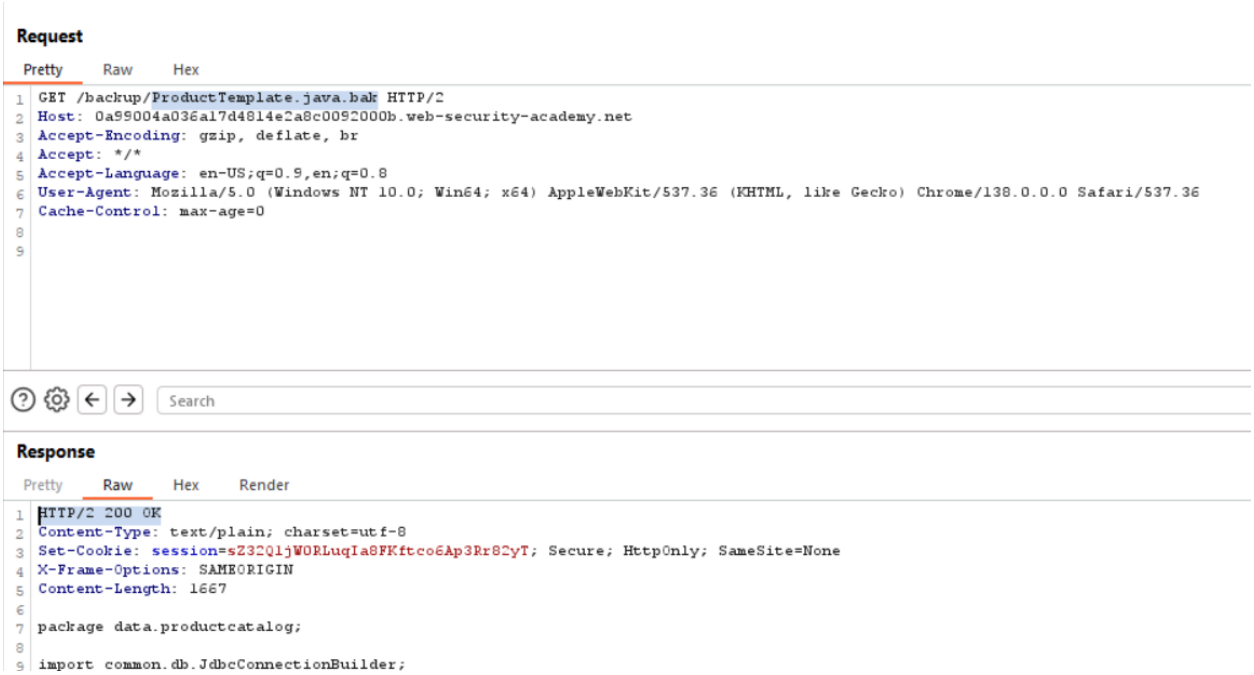
- In Burp Suite, sent the request to `/backup/` to Repeater to manually inspect contents.
- Found the backup Java file: ProductTemplate.java.bak



- Appended the filename directly to the `/backup/` path in the Repeater URL field:



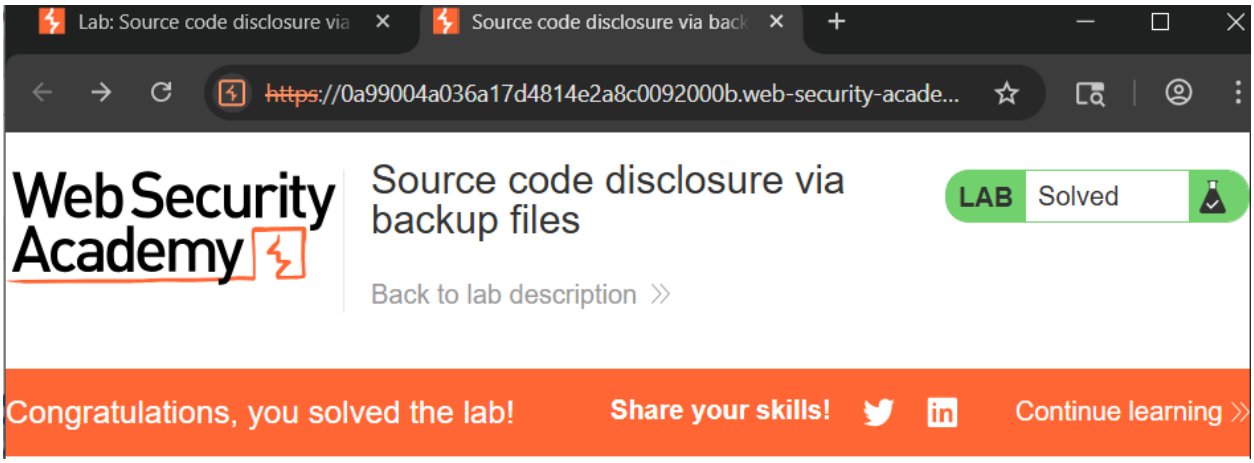
- Sent the request and received the response containing the source code.



5. Identify and Extract the Password

- Located the `getConnection()` or similar method, containing a line like:

6. Submit the Solution



Vulnerability Analysis:

- **Vulnerability Type:** Sensitive file exposure via backup file in web-accessible directory
- **Root Cause:** Backup file left in a public location without proper access control
- **Sensitive Data Leaked:** Hard-coded PostgreSQL database password
- **Risk Level:** High – Credentials leakage can lead to unauthorized database access

Mitigation Recommendations:

- Never store or deploy backup files (.bak, .old, .zip) in web-accessible directories.
- Configure the web server to block access to sensitive paths and file extensions.
- Use .gitignore, deployment automation, or secure pipelines to prevent accidental file exposure.
- Scan your environment regularly for exposed files using automated tools.
- Avoid hard-coding sensitive credentials in source code. Use environment variables or secret managers.

Conclusion:

This lab illustrates how backup files can unintentionally leak source code and credentials. By discovering the /backup/ProductTemplate.java.bak file via robots.txt, we extracted the hard-coded database password and successfully completed the lab.

End…