Lab Assignment: Investigating SSRF Vulnerabilities in LLM-Integrated Applications

CSCD 525 Advanced Computer and Information Security

1. Objective

This lab explores how LLM-integrated services that fetch external documents for summarization or chat responses can be manipulated to trigger SSRF vulnerabilities. You will build such a system from scratch, conduct controlled attacks, and propose countermeasures. You will also implement the defense and will show the results after hardening/ countermeasures. There is an extra credit for implementing more LLM related attacks.

2. Instructions

A. What to Build

As shown a demo in class, implement the following:

a. Simulated Internal Server:

- A mock internal server with routes like /config and /admin/secret (Port 8000).

b. Vulnerable LLM App:

- You may use any publicly available pre-trained summarization model.
- A vulnerable LLM summarizer app that fetches and summarizes any URL input (Port 5000).
- Accepts `url` param, fetches content, summarizes with LLM
- Clean web UI, no input validation (intentional SSRF)

C. Demonstrate SSRF Attack

Input example: http://localhost:8000/admin/secret

D. Propose and Implement Defenses

Propose and implement defese against this attack. You may Block requests to internal IPs and reserved hostnames or Optionally add domain allowlists and timeout/content limits.

E. Extra Credit (5 pts)

Choose one from the OWASP LLM Top 10:
`https://owasp.org/www-project-top-10-for-large-language-model-applications/`

4. Submission Guidelines

This assignment is due on **May 26, 2025**

A. Whitepaper/Report

1. Title & Author Info
2. Abstract
3. Introduction
4. Experiment Setup
5. Attack Procedure & Results
6. Defense Proposal & Results
7. Conclusion & Reflection
8. References

B. Code and Demo Video

Submit both Flask apps as `.py` files. Comment vulnerabilities and defenses. Document your code well. Share the GitHub repo Link with a Demo video Showing web UI, terminal logs, and defense proof.

5. Evaluation Rubric

Criteria	**Points(100)**
System built from scratch (2 apps)	20
Successful SSRF attack demonstration	20
Quality of whitepaper documentation	25
Defense implementation & testing	20
Code quality and comments	10
Bonus OWASP vulnerability (optional)	5