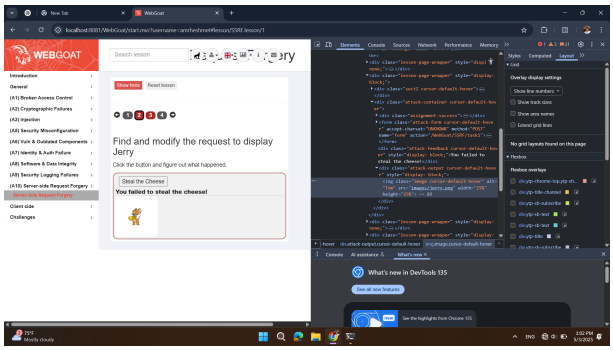
Server-Side Request Forgery (SSRF) Report - WebGoat

Chapter 1: Introduction to WebGoat and SSRF

WebGoat is an open-source educational application provided by OWASP, aiming to teach web application security concepts by presenting intentional vulnerabilities to experiment with and solve. The SSRF (Server-Side Request Forgery) lesson demonstrates how an attacker can exploit the server to send HTTP requests to unauthorized sources.



Chapter 2: What is SSRF Vulnerability?

SSRF vulnerability allows an attacker to trick the server into sending HTTP or other types of requests to attacker-specified destinations, which may lead to accessing internal services, sensitive data, or bypassing security protections.

Chapter 3: Steps to Exploit SSRF in WebGoat

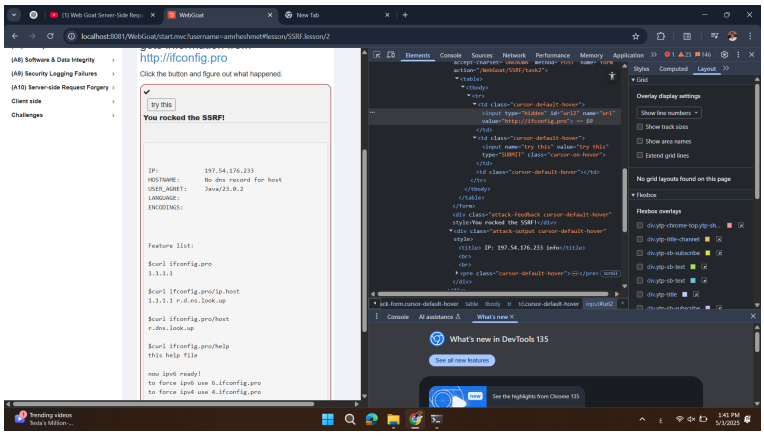
Lesson 1: Attempt to exploit to display the 'Jerry' image instead of 'Tom'.

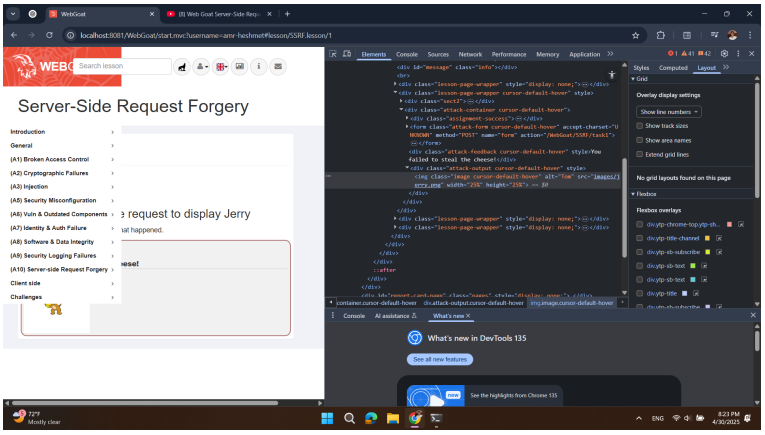
- Using browser developer tools, the request value is modified to change the requested resource.
- Upon sending the modified request, the Jerry image is displayed, indicating successful exploitation.

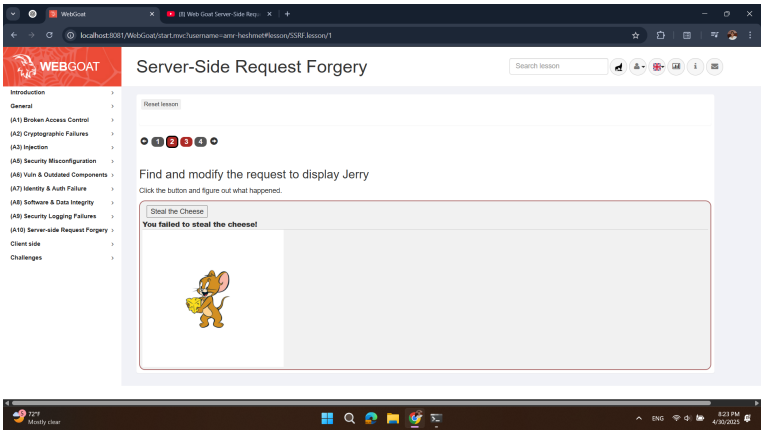
Lesson 2: Exploiting SSRF to access server information such as IP address.

- Modify the hidden form field to point to an external address like http://ifconfig.pro.
- Send the request to retrieve server data.
- Information such as IP, HOSTNAME, and User-Agent appears, confirming successful exploitation.

Server-Side Request Forgery (SSRF) Report - WebGoat







Chapter 4: Practical Example of Request Modification

Using curl, you can simulate a POST request and modify its data to exploit SSRF vulnerability:

```
curl -X POST \
  -d "url=http://ifconfig.pro" \
```

http://localhost:8081/WebGoat/SSRF/task2

You can also modify the request directly via browser developer tools.

Chapter 5: Methods to Test SSRF Vulnerability

- Use manual tools like DevTools to modify requests.
- Use automated tools like Burp Suite and OWASP ZAP.
- Test application logic to verify how input is handled.
- Ensure random or unsafe URLs cannot be requested.

Chapter 6: Protection and Prevention Against SSRF

- Validate input thoroughly.
- Use whitelist of allowed sources.
- Disable unnecessary protocols like file:// or gopher://.
- Restrict access to internal networks with firewalls.
- Log and monitor suspicious requests.

Chapter 7: Conclusion

The SSRF lesson in WebGoat provides a practical environment to learn how to exploit this vulnerability and how to protect against it. It is essential for developers and security engineers to understand SSRF mechanisms to avoid it and secure their applications properly.