**Information Security COMP 421**

**Spring 2025 Section B**

**Assignment 1**

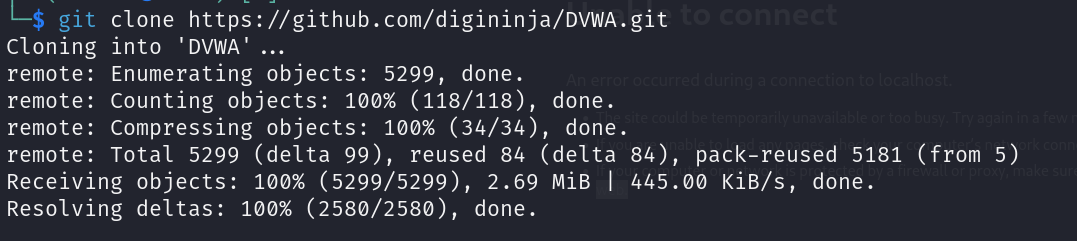
**Hafsah Shahbaz 251684784**

**✅ Task 1: Identify File Inclusion Point (LFI)**

**Observation:**  
By modifying the page parameter in the DVWA File Inclusion module to point to a system file (../../../../etc/passwd), we were able to successfully exploit a Local File Inclusion vulnerability. The contents of the system password file were displayed in the browser, confirming that the application does not properly sanitize user input.

**What I Learned:**

* LFI allows attackers to read sensitive files on the server.
* File inclusion points are often exploited using directory traversal (../).
* Input validation and restricting file paths are critical to prevent LFI attacks.

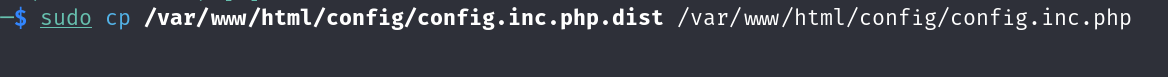


A black background with white text

AI-generated content may be incorrect.

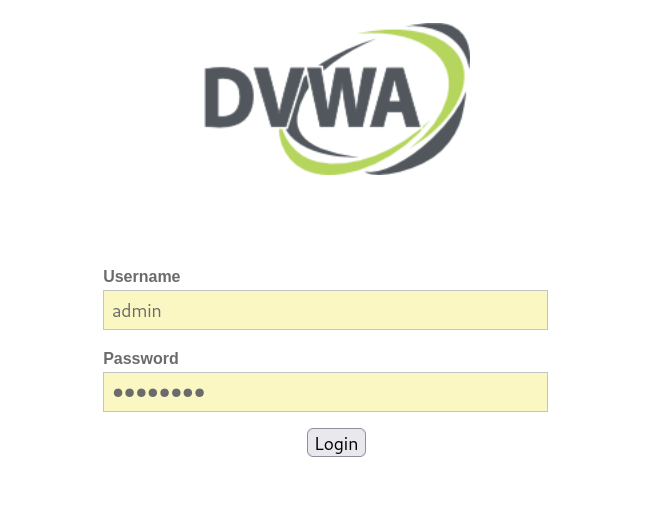
A screen shot of a computer

AI-generated content may be incorrect.



A screenshot of a computer code

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.



**✅ Task 2: Remote File Inclusion (RFI)**

**Observation:**  
A malicious shell.php file containing <?php system($\_GET['cmd']); ?> was hosted using a local PHP server. By passing a remote URL as the page parameter and including a command, we successfully triggered code execution on the server. The output of the id command was returned, confirming the success of the RFI exploit.

**What I Learned:**

* RFI vulnerabilities allow remote attackers to execute arbitrary code.
* RFI requires the PHP directive allow\_url\_include to be enabled.
* Hosting remote payloads and chaining parameters (like %3Fcmd=id) can bypass broken input handling.
* Disabling allow\_url\_include and properly sanitizing user inputs are essential protections.

