Challenge 2: Web Server Vulnerabilities

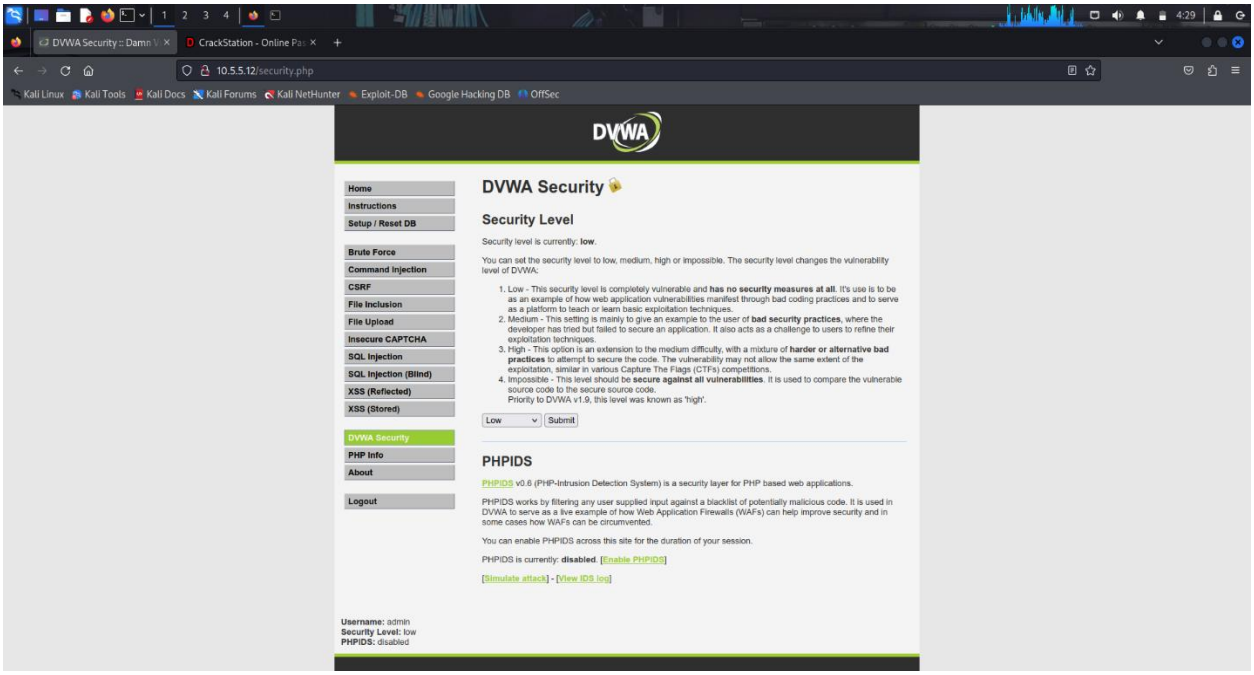
In this part, you must find vulnerabilities on an HTTP server. Misconfiguration of a web server can allow for the listing of files contained in directories on the server. You can use any of the tools you learned in earlier labs to perform reconnaissance to find the vulnerable directories.

In this challenge, you will locate the flag file in a vulnerable directory on a web server.

Step 1: Preliminary setup

a. If not already, log into the server at 10.5.5.12 with the **admin / password** credentials.

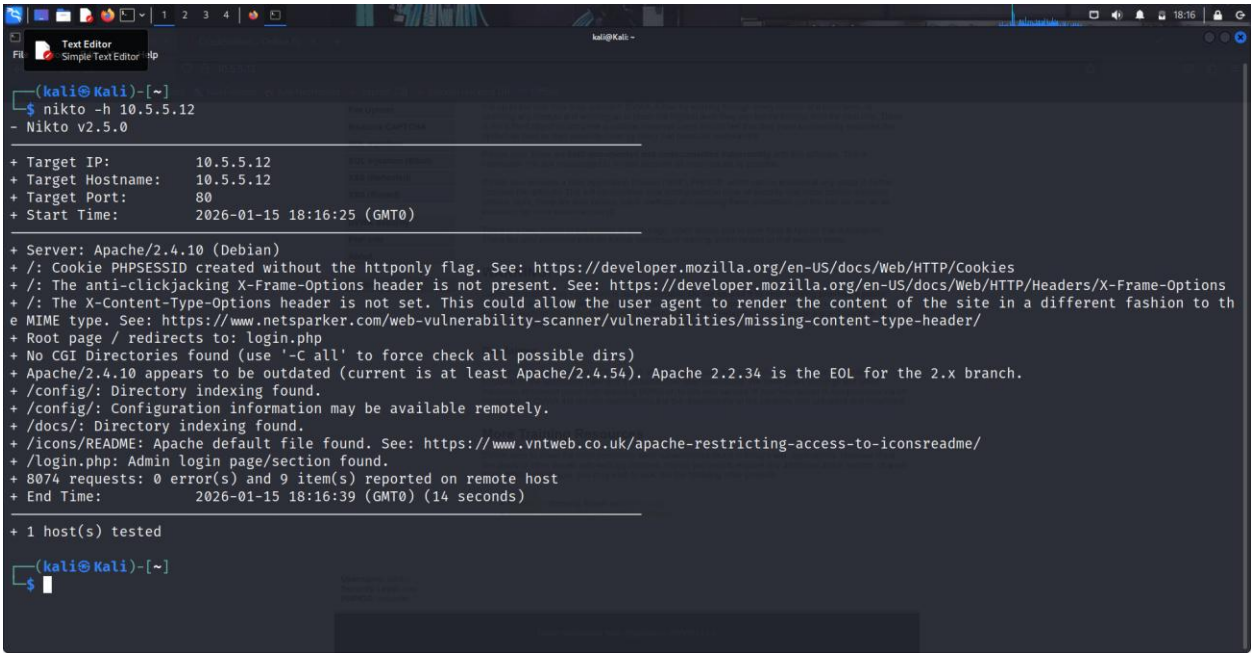
b. Set the application security level to low.



Step 2: From the results of your reconnaissance, determine which directories are viewable using a web browser and URL manipulation.

Perform reconnaissance on the server to find directories where indexing was found.

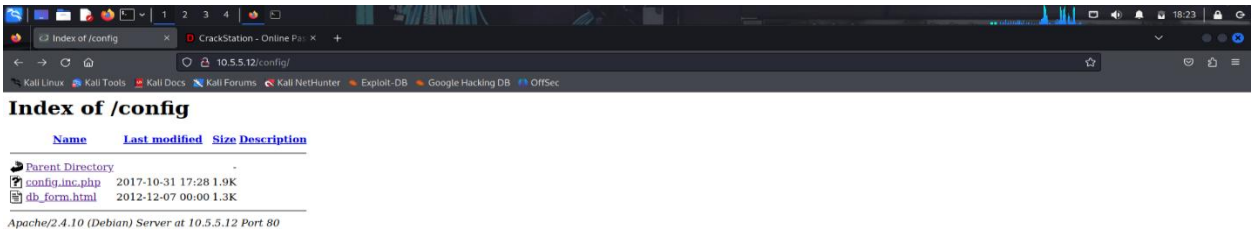
In the screenshot below I perform reconnaissance on the server to find directories where indexing was found using Nikto Command: nikto -h 10.5.5.12



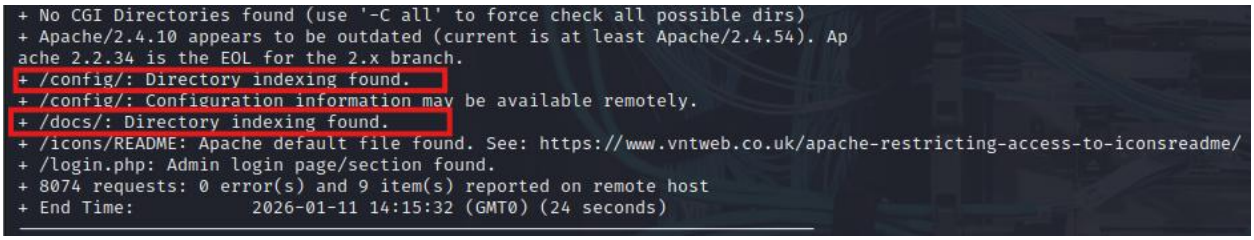
Which directories can be accessed through a web browser to list the files and subdirectories that they contain?

Step 3: View the files contained in each directory to find the file containing the flag.

Create a URL in the web browser to access the viewable subdirectories. Find the file with the code for Challenge 2 located in one of the subdirectories.

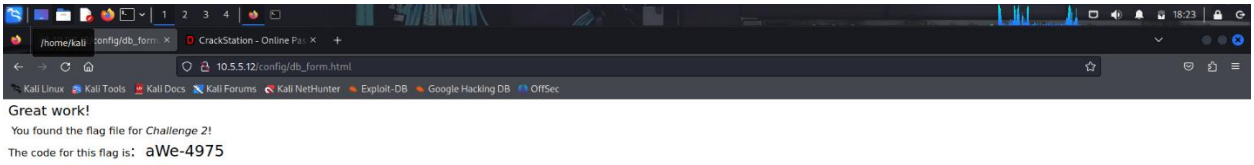


In which two subdirectories can you look for the file?



What is the filename with the Challenge 2 code? db_form.html

Which subdirectory held the file?



Step 4: Research and propose directory listing exploit remediation.

What are two remediation methods for preventing directory listing exploits?

Two key methods to prevent directory listing exploits are **disabling directory indexing** in your web server's configuration (e.g., using Options -Indexes in Apache or autoindex off in Nginx) and **placing a default index file** (like index.html) in directories, which the server serves instead of a list of files. Additionally, restricting access with proper file permissions and moving sensitive files outside the web root are crucial defenses.

Remediation Methods

1. **Disable Directory Indexing via Server Config**:
    - **Apache**: Add Options -Indexes within <Directory> blocks in your Apache configuration (httpd.conf or .htaccess) to stop directory listings.
    - **Nginx**: Set autoindex off; in your location block within the nginx.conf file.
    - **IIS**: Use the IIS Manager console to disable Directory Browsing for specific sites or directories.
2. **Add Default Index Files**:
    - Create a simple index.html (or similar default file) in each directory that should not list its contents. When a user requests the directory, the server serves this file instead of a directory listing.