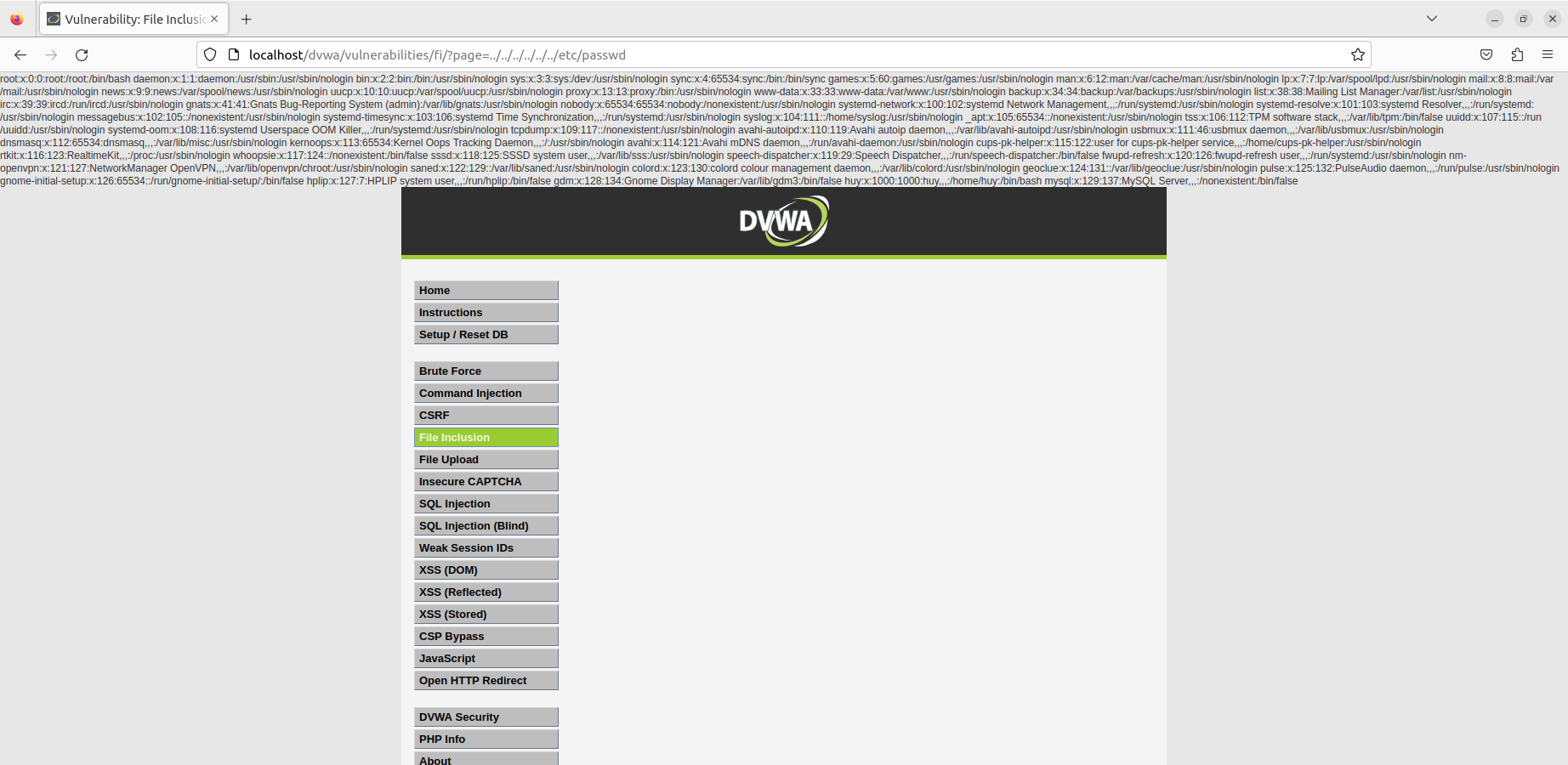
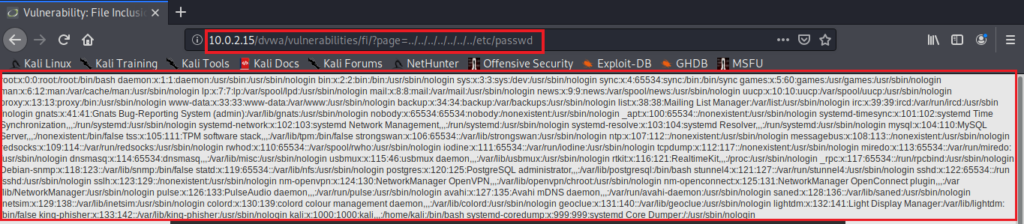
**Difficulty: LOW**

Go to file inclusion tab and change the URL from incude.php to

**?page=../../../../../../../etc/passwd**





As you can see, we got the data of **/etc/passwd** file. You can also read other important files to gather more sensitive data about the web-server so that you can plan your next exploit.

Now, let’s try to execute some commands, for that first start burp suite and make sure the interceptor is on.

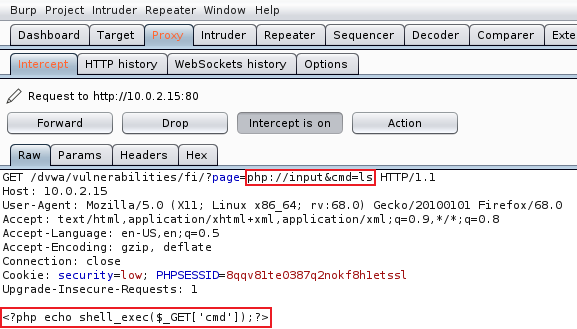
Reload the page and head over to burp suite.

Change **include.php** to

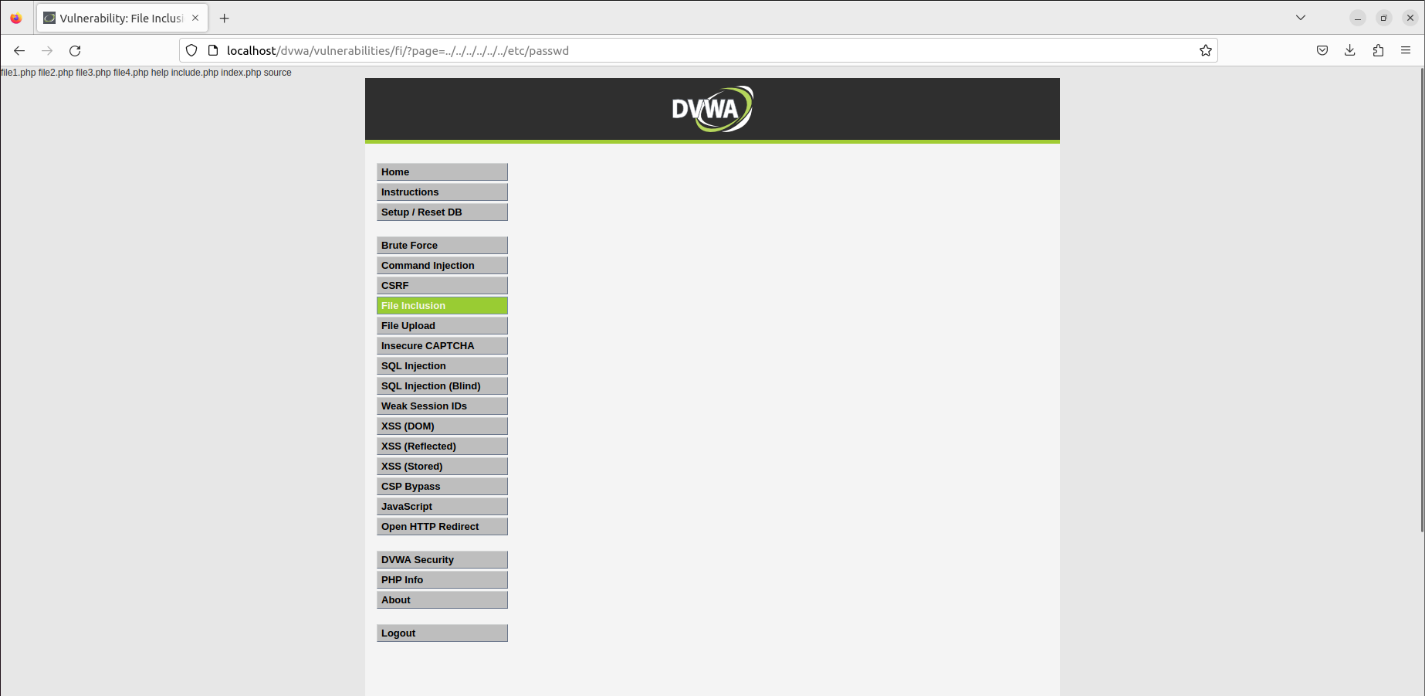


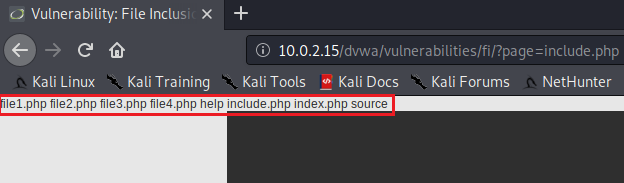
Next, add the following PHP code to execute the above command.





Forward the request.





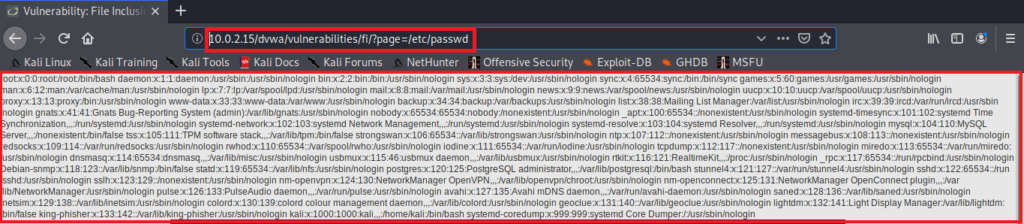
we have successfully exploited LFI vulnerability to read files and run commands.

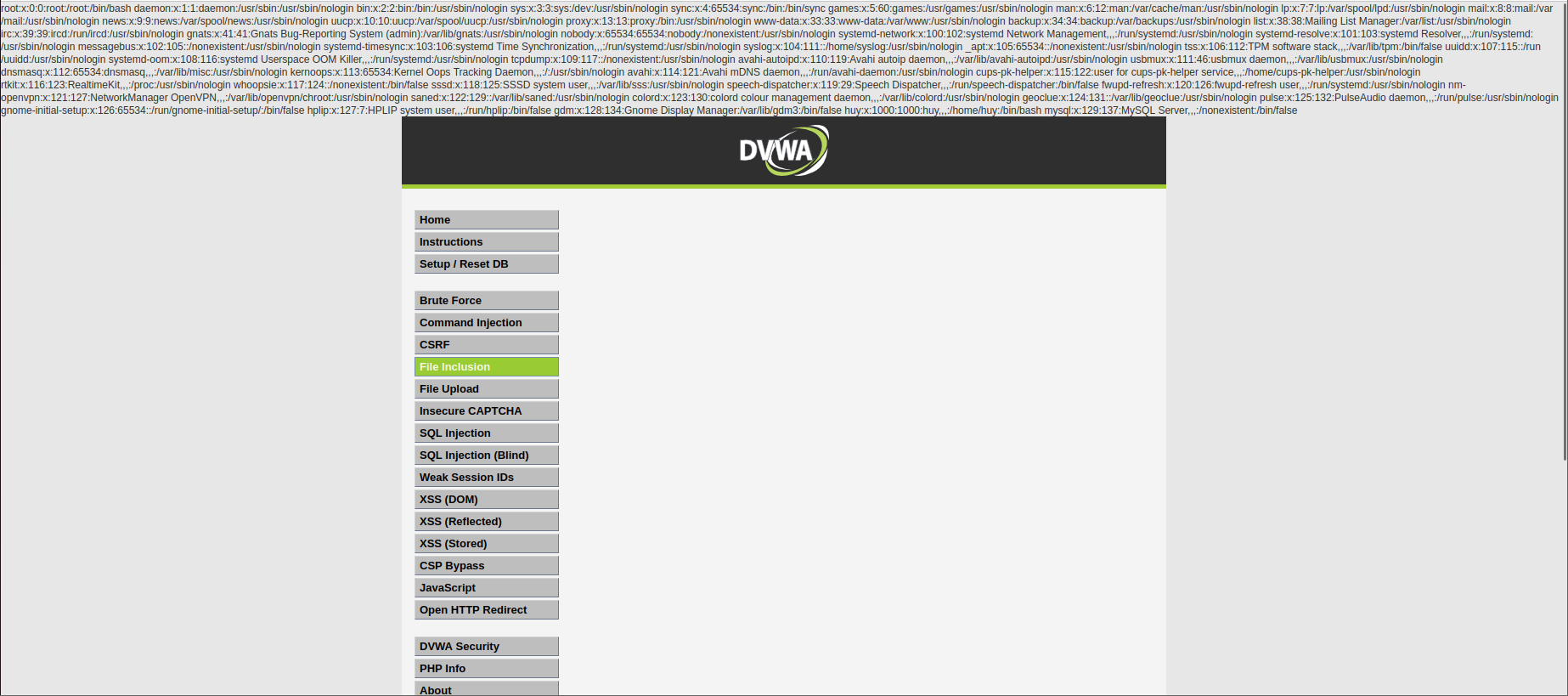
### Difficulty: MEDIUM

go on and try the exploits we used in low difficulty. You will notice that you can’t read files like before using the directory traversal method. However, we can still execute commands using php://input.

So, the server is more secure and is filtering the ‘../’  pattern. Let’s try to access the file without ‘../’

Change **include.php** to **/etc/passwd**





### Difficulty: HIGH

the target is more secure, as it is only accepting “include.php” or inputs starting with the word “file”. If you try anything else, it will show “File not Found”.

In this level of security, it’s hard to gain reverse shell, but we can still gather sensitive info using the “File” URI scheme. (because it starts with the word “file”)

Change the URL from include.php to

**?page=file:///etc/passwd**

