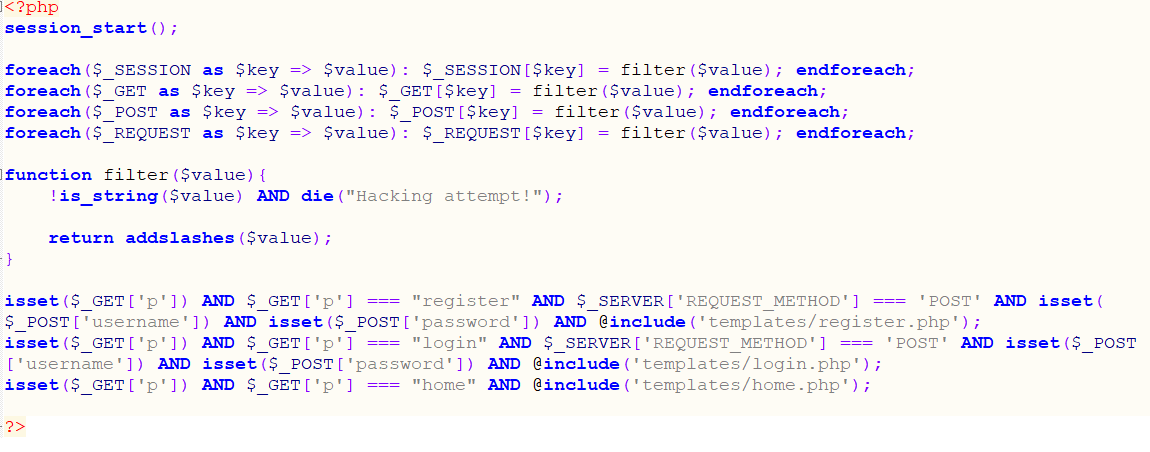
**Baby sql is not baby anymore**

This task was about having a web application that filters each user input from $\_SESSION, $\_GET, $\_POST and $\_REQUEST and all the pages are included via index.php.



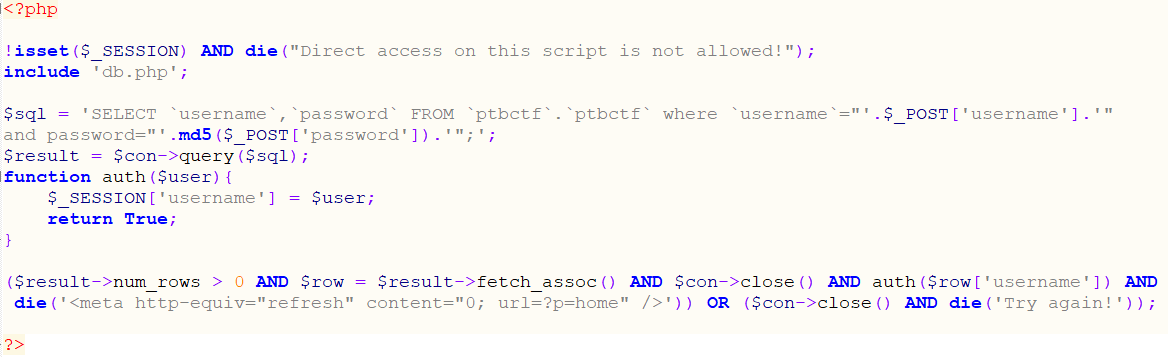
As we could see, we cannot exploit an SQL Injection if we access a page via index.php. If we try to access one of the pages directly we will end up getting the following message:

|  |
| --- |
| Direct access on this script is not allowed! |

That’s because all pages start with the following php line:

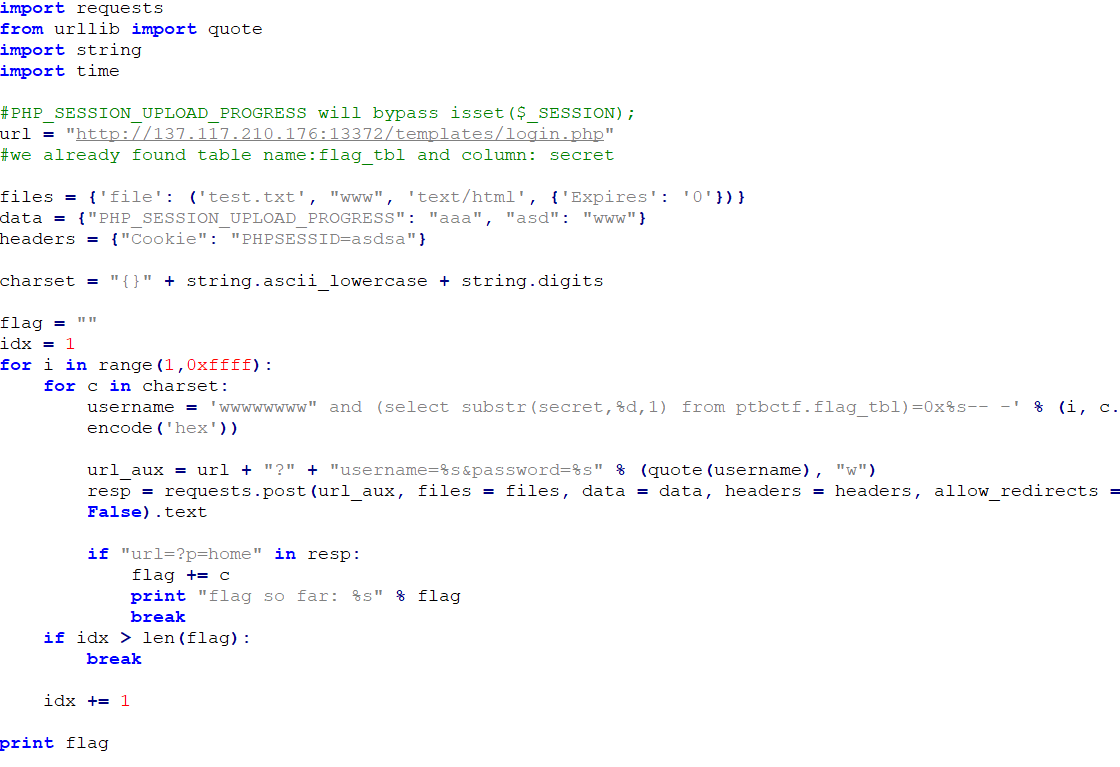
|  |
| --- |
| !isset($\_SESSION) AND die("Direct access on this script is not allowed!"); |

Example: templates/login.php:



There are many applications that use this kind of check isset($\_SESSION). However, it can be bypassed because we can force the session to start using “PHP\_SESSION\_UPLOAD\_PROGRESS”; I saw this in one of Orange Tsai's tasks. With this task I wanted to show a real situation I found in one of my penetration tests.

Having a bypass for the first line of login.php, then we can exploit an SQL Injection attack on the login form.



The script above encoded in base64 (for testing purpose):

|  |
| --- |
|  |

Running the script above we will get the flag:

