# PHP Lab 6 Local File Inclusion (LFI) and Directory Traversal

This example is derived from CyberAces Module 3, Basic Web Security slide 5.

Also, look at <https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_File_Inclusion>

Some large sites label their pages by number and link them by calling the page number with GET or POST methods. We will simulate that with a master page that allows the user to request a page number. There will be two small pages and a PHP script to call them.

This is HTML for the master page that we will use for demonstrating LFI and RFI. Paste it into gedit and save it as /var/www/html/LFItest.html. Note: this uses an older method for creating HTML forms.

<html>

<title> LFI Example </title>

<body>

<h4 LFI Example </h4>

This site has two pages, 1 and 2

<br>

<br>

<form action="getpage.php" method="get">

Enter 1 or 2: <input name="Page" type="text" />

<input type="submit" />

</form>

</body></html>

Now, let's create simple pages 1 and 2.

Paste this into 1 and save it as /var/www/html/1 :

<?php

echo "this is page 1";

?>

Paste this into 2:

<?php

echo "this is page 2";

?>

This is the php script called by the html form, LFItest.html. Paste it into /var/www/html/getpage.php

<?php

include($\_GET['Page']);

?>

All of the above files (LFItest.html, 1, 2, and getpage.php) should be in your /var/www/html directory.

When you point your browser to http://localhost/LFItest.html, you should see a page asking you to enter a page number (see LFIwithPage1.PNG). If you don't see anything, make sure your httpd service is running (su -; service httpd start). Now enter 1 for the page number and click Submit Query.

Note the URL is <http://localhost/getpage.php?Page=1>.

Can you think of a way to abuse this by entering something other than 1 or 2?

Hmmm, what happens if change the URL in the browser, so we get this?

http://localhost/getpage.php?Page=/etc/passwd

Yikes! Do you recognize the output? This is using LFI to compromise information on the server. If the site allows us to save file content (blog posts?) we could use LFI to execute our scripts.

Try to view the /etc/shadow file. Did it work? Why not? Hint: use the trusty ps aux and pipe it into a grep for apache2. This will allow you to see the apache2 processes and determine the user they are running under. In addition to validating user input so LFI does not occur, it is wise to limit the access of the web site user.

## Directory Traversal

Often the attacker does not know where they are in the web server’s file system, and paths that start at root may be blocked. Their alternative is to use variations of ../ which means go up one directory. They will have to experiment to find the correct number of times to use ‘go up one directory’ to use to get to the directory they want to reach. Paths can get long, so it is not uncommon to see something like this in your server logs: ../../../../../../../../../etc/passwd.

<https://owasp.org/www-community/attacks/Path_Traversal>

Use directory traversal to print etc/passwd on your browser.

## Server logs

Examine your server logs and find an example of each attack you made against your server.

# Hand In

1. Hand in a sample from your server logs showing each type of attack you made?
2. Why is it important to monitor logs?
3. Were you able to view /etc/shadow from your browser? Why or why not?
4. What could you do to prevent the LFI attack you just demonstrated?