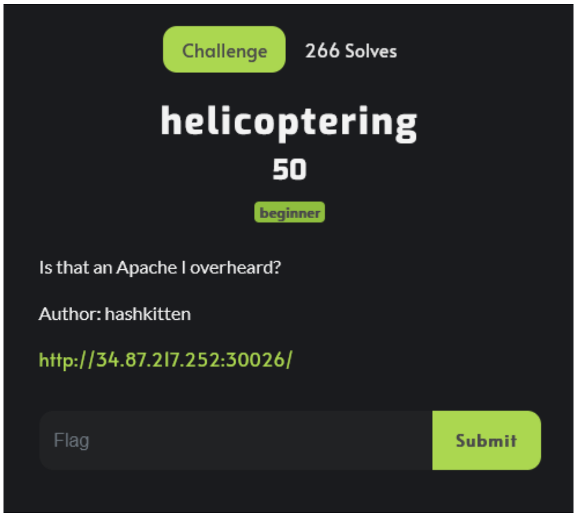
First hint we get is that this is some sort of Apache issue. Let's check out the website.



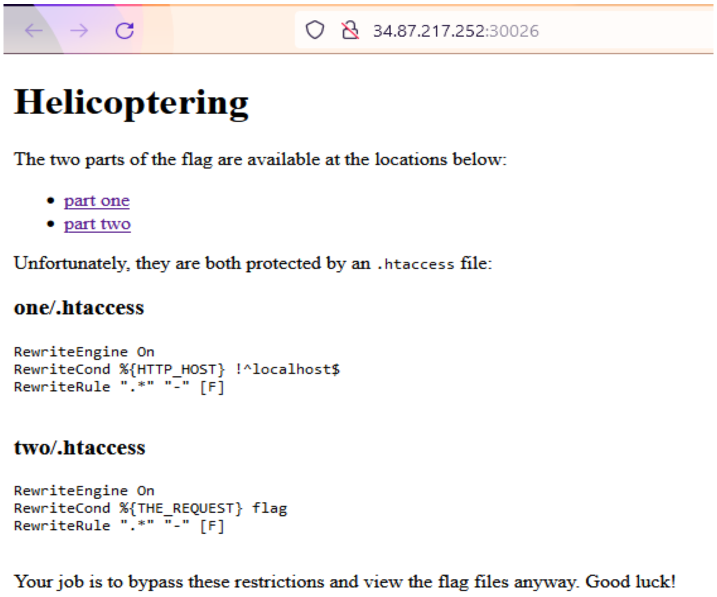
helicoptering

50

Is that an Apache I overheard?

Author: hashkitten

http://34.87.217.252:30026/



Helicoptering

The two parts of the flag are available at the locations below:

- part one
- part two

Unfortunately, they are both protected by an `.htaccess` file:

one/.htaccess

```
RewriteEngine On
RewriteCond %{HTTP_HOST} !^localhost$
RewriteRule ".*" "-" [F]
```

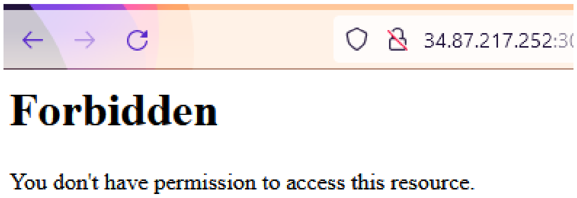
two/.htaccess

```
RewriteEngine On
RewriteCond %{THE_REQUEST} flag
RewriteRule ".*" "-" [F]
```

Your job is to bypass these restrictions and view the flag files anyway. Good luck!

So we have a two part flag, each protected by a simple Apache .htaccess file. I've only worked with Apache a few times, but I know that the .htaccess file is essentially a part of the server that will manage access to certain pages via a set of defined rules.

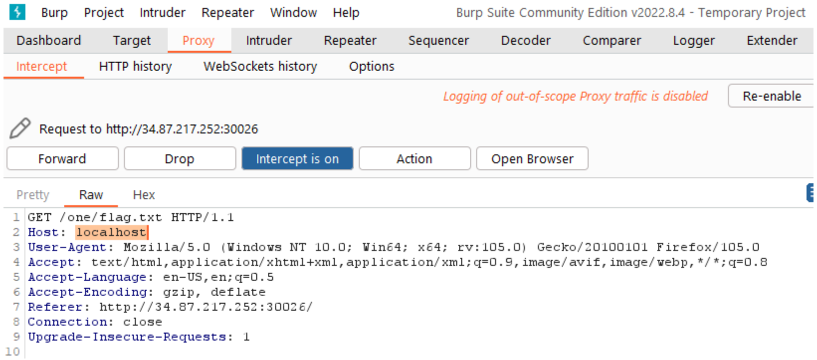
Here, we can see that if we go to part one of the flag, we are given a 403 Forbidden. Let's examine the .htaccess rules for the first part that is preventing us from seeing the flag,



Forbidden

You don't have permission to access this resource.

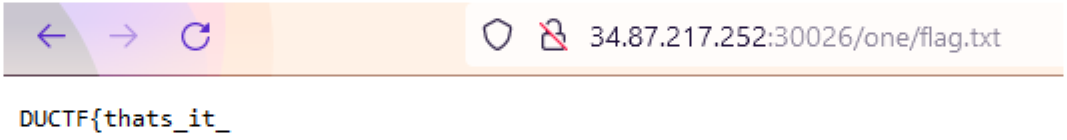
"RewriteCond %{HTTP_HOST} !^localhost$". What is this doing? (I have no idea, let's google.) Basically, it is telling the server that when it receives a request, it should check the HTTP_HOST field to see if it is coming from "localhost". Notice the !, this rule is saying if the request is NOT from localhost, do not let it through, and also the regex ^ $ format… So let's force it to think we're localhost.



Request to http://34.87.217.252:30026

```
GET /one/flag.txt HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://34.87.217.252:30026/
Connection: close
Upgrade-Insecure-Requests: 1

```

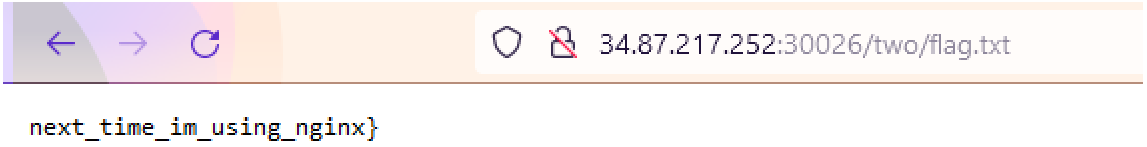


34.87.217.252:30026/one/flag.txt

```
DUCTF{thats_it_
```

We send off the request, and we get part 1 of the flag.

Part 2 is a little easier, the rule says “RewriteCond %{THE_REQUEST} flag” meaning THE_REQUEST can’t have any occurrence of “flag” anywhere in it… So how would we bypass this to request /two/flag.txt??? URL encoding! The server is only checking for “flag” exactly, what if we give it something like fl%61g.txt? We know that %61 = ‘a’ in URL encoding, so when we make the request, we are given part 2.



34.87.217.252:30026/two/flag.txt

```
next_time_im_using_nginx}
```