CA03

Robert Skelton

11.2.2014

This problem is taken from picoCTF 2014, and is my own solution that I found while doing some problems with the Auburn CTF team this week.

The problem is called Potentially Hidden Password, worth 100 points, and the prompt is: “This [Daedalus Corp. website](http://web2014.picoctf.com/potentially-hidden-password-3878213/) loads images in a rather odd way... [[Source Code]](https://picoctf.com/problem-static/web/potentially-hidden-password/index.phps)”

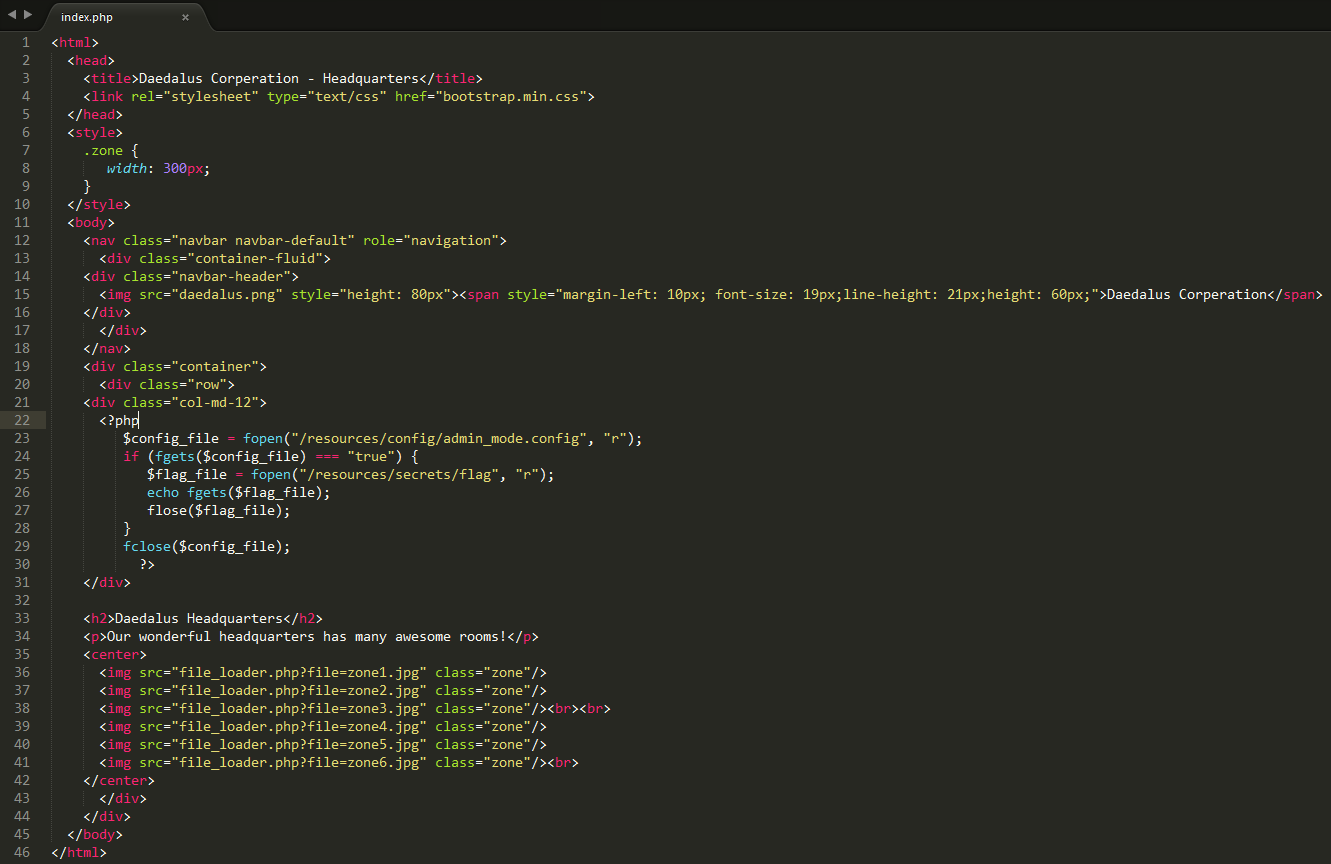
Webpage link: <http://web2014.picoctf.com/potentially-hidden-password-3878213/>

We are also given the hint: “The 'file\_loader.php' page might be able to serve more than just images.”

We are given the source code, in a phps file. Apparently this is syntax highlighted php, but none of my programs open it by default, so I just renamed it index.php.

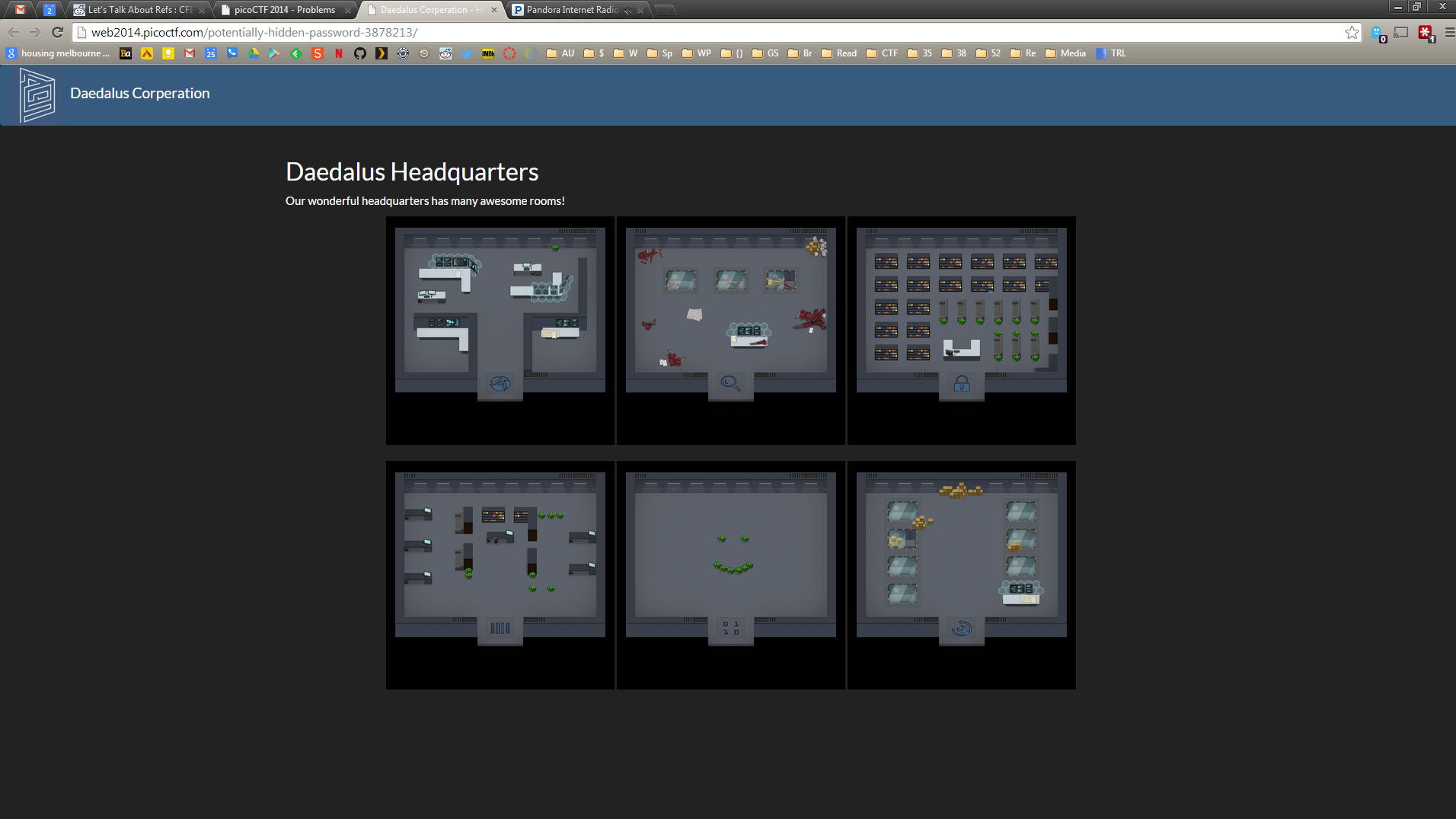


The full source code is shown below

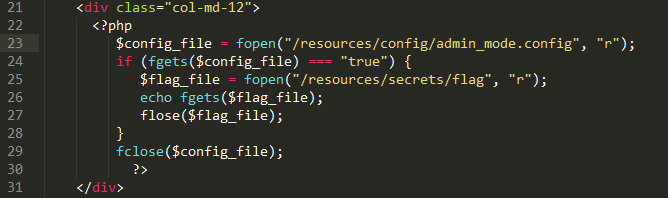


The important parts that we’ll use are Line 23, which is where the flag is stored, and line 36-41’s file\_loader.php function.

This is the webpage before finding the flag, or doing anything really:



Checking out the source code given, there is a little php that runs in the page. Looks like we’re trying to get to the flag which might be located at /resources/secrets/flag

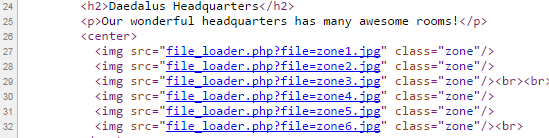


Notice if you don’t use the given source code, and you just examine the web page’s source code, you can’t see that there is any php that runs on the page.

To open the source code in a browser, right click and then click View Source.



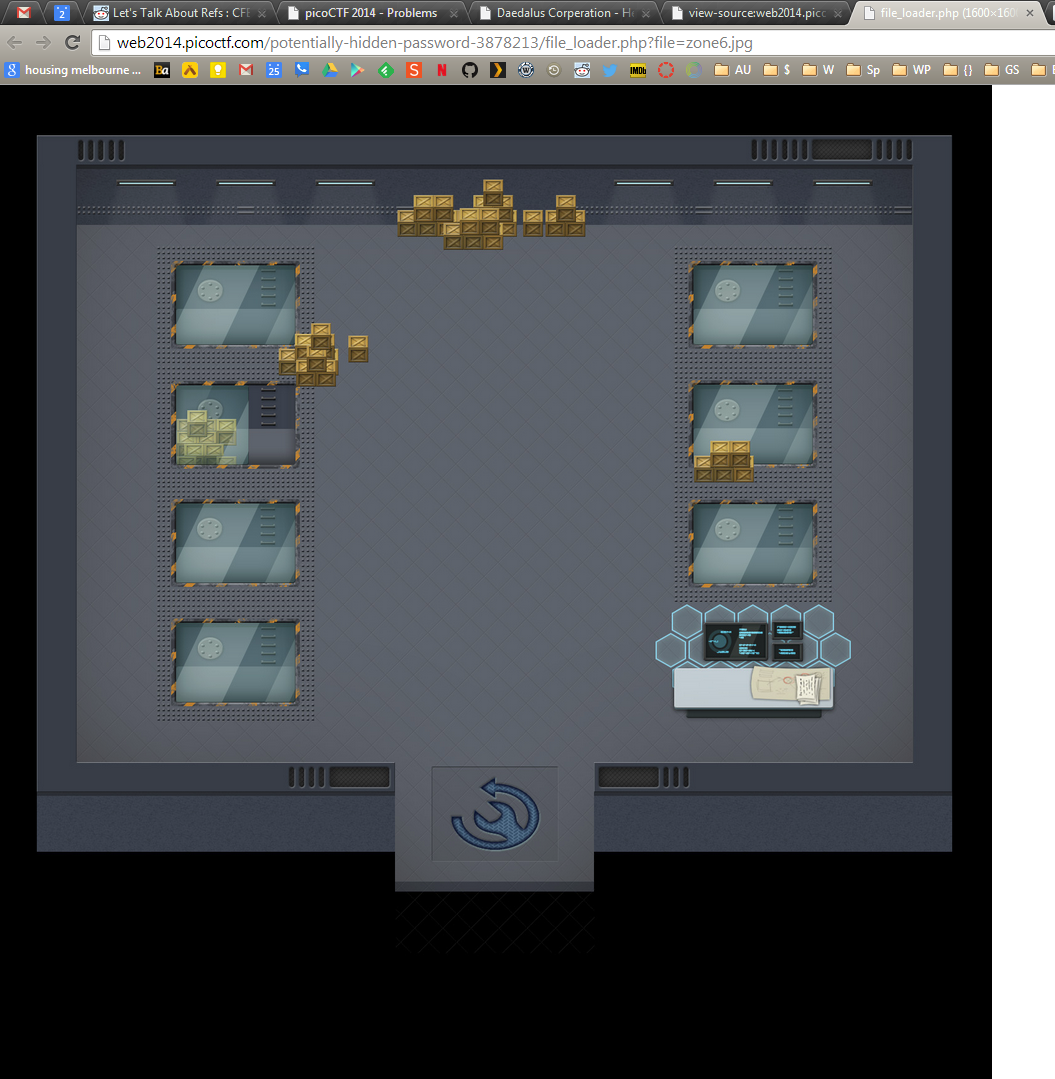
This file\_loader.php will probably be useful, so let’s click on that.



OK, that loads the images, but it can probably also load files.

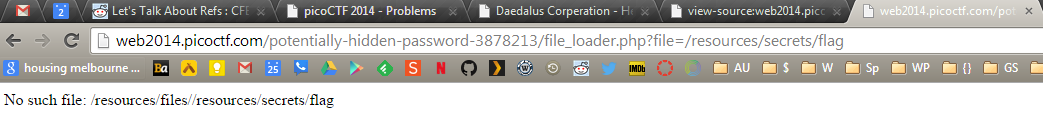
Notice that file\_loader has an argument in the URL.

Let’s see if we can use file\_loader.php to get us to the location of the flag by changing the URL from file=zone6.jpg to file=something different.



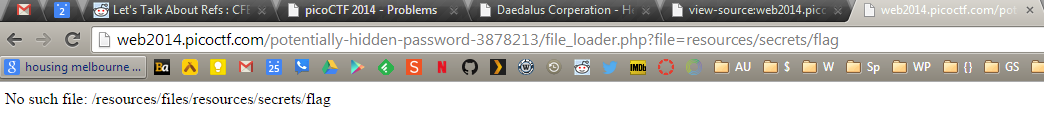
I changed the file= to point to /resources/secrets/flag. This spits out the output “No such file: /resources/files//resources/secrets/flag”.

These are some handy debug messages.

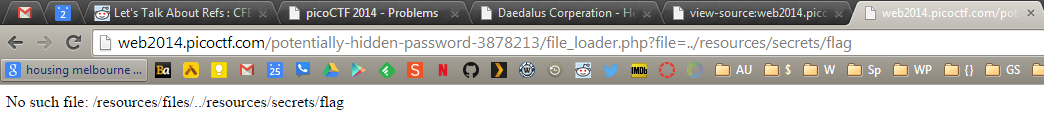


It looks like the webpage is located in /resources/files on the server. We can probably exploit this address with a little knowledge of how Linux file systems are organized.

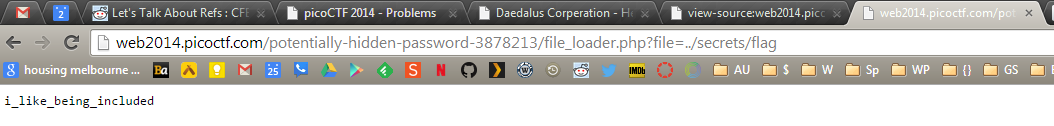
Taking out the first slash before resources gives us this:



We obviously need to back up a directory, in Linux you can use the “..” command to go up a level.



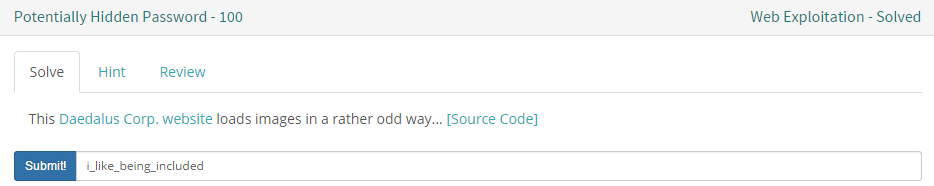
This should be it… except we’re still not in the right directory to see the flag. The resources directory is repeated; let’s try taking that out of the URL being passed through file\_loader.php.



Sweet, this brings us to the directory that the flag is included at. According to the php script in index.php, if we can navigate to /resources/secrets/flag, the flag will be fopen’d.

Therefore, the flag here is i\_like\_being\_included

So what we did is use “..” to go up a directory, then cd to /secrets/flag, all by changing the arguments of the file\_loader.php script.



Submit it to picoCTF, get 100 points,???, PROFIT!!!

This attack is an example of Web Parameter Tampering (via <https://www.owasp.org/index.php/Web_Parameter_Tampering>), and explicitly is the Example 3 that is pointed out.

This attack is carried out by editing the URL directly, and can be used on lots of other websites by simply modify parameters in URL addresses.

The End.