xctf中php_rec的writeup,一个GET请求拿到flag——XCTF 2018 Final PUBG(WEB 2) Writeup

转载

weixin_39544333 于 2021-03-13 01:34:55 发布 97 收藏
文章标签： xctf中php_rec的writeup
环境

XCTF Final 和 HITB 赶在了周四周五，周四晚上拿到题目，此时队友已经对PHP代码完成了解密工作。

解密后的代码： http://static.cdxy.me/DECODED.zip

github：https://github.com/Xyntax/XCTF-2018-Final-WEB2-PUBG

解密后的代码读起来有点麻烦，并无大碍。

```
require( "../kss_inc/inc.php" );
$_obfuscate_jIaUiIeSjZWKlIqLkIqOioc = new mysql_cls( );
$_obfuscate_lI6OiJSPjZWVi5GQhoiPjpU = _obfuscate_iZSVk4mLkY_LlIeHh5WKlZA(  _obfusca
$_obfuscate_kouLj4_JkJWKkJCQkIaMjZE = "服务端更新";
include( "c_head.php" );
$_obfuscate_koaSiYqGjIqMiZSLk4uGiZU = "serverupdate";
if ( isset( $_GET['pakname'] ) )
{
    $_obfuscate_koaSiYqGjIqMiZSLk4uGiZU = $_GET['pakname'];
}
if ( isset( $_GET['cver'] ) )
{
    $_obfuscate_koaSiYqGjIqMiZSLk4uGiZU = $_GET['cryptver'];
}
if ( !_obfuscate_ipWHiIuOiYuPjIaPkZSThok(  _obfuscate_kpKNjomRlYuUk4qLlI2MkJU: "curl_
{
    $_obfuscate_koaOh5WRio_ThoaLjI2Sk48 = "<pre>";
    $_obfuscate_koaOh5WRio_ThoaLjI2Sk48 .= "当前服务器不支持curl，无法在线升级，请登陆登
    _obfuscate_kYyOhouLjo2Gh4eNj4iQlIg( $_obfuscate_koaOh5WRio_ThoaLjI2Sk48 );
}
if ( is_file(  filename: KSSROOTDIR."kss_tool".DIRECTORY_SEPARATOR."_webup.php" ) )
{
    unlink(  filename: KSSROOTDIR."kss_tool".DIRECTORY_SEPARATOR."_webup.php" );
```

有点魔幻的get flag过程

当晚并没有找到突破口，但发现kss_admin/admin_update函数疑似是exp链路中的一环。

其中120行发现CMS更新功能，从远端主站拉取代码写入本地:

$_obfuscate_koiKkliPjl6UkYeRIlqNhoc = _obfuscate_lY6Gk5KMkYmPjlyPhpCOlYc(
"http://api.hphu.com/import/".$_obfuscate_koaSiYqGjIqMiZSLk4uGiZU.".php?
phpver=".PHP_VERSION."&webid=".WEBID."&rid=".time( ), 300);

跟进Yc函数，发现其注册了两个curl的回调 read_header,read_body：

curl_setopt($_obfuscate_joiNh4alhouViZGQho_Jil4, CURLOPT_HEADERFUNCTION, "read_header");

curl_setopt($_obfuscate_joiNh4alhouViZGQho_Jil4, CURLOPT_WRITEFUNCTION, "read_body");

其中read_body函数会将curl到的content写到本地文件kss_tool/_webup.php

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

想要使用admin_upload.php这个点写shell，需要满足两个条件：

绕过admin权限验证

能控制curl部分的回显

周四当晚并没有突破。周五早9点比赛环境恢复，我打开ipython对这个疑似webshell的地址kss_tool/_webup.php做了监控。如果这个文件被其他队动了，就证明这个思路是可行的。

In [1]: while True:

...: try:

...: r = requests.get('http://guaika.txmeili.com:8888/kss_tool/_webup.php')

...: except Exception,e:

...: print e

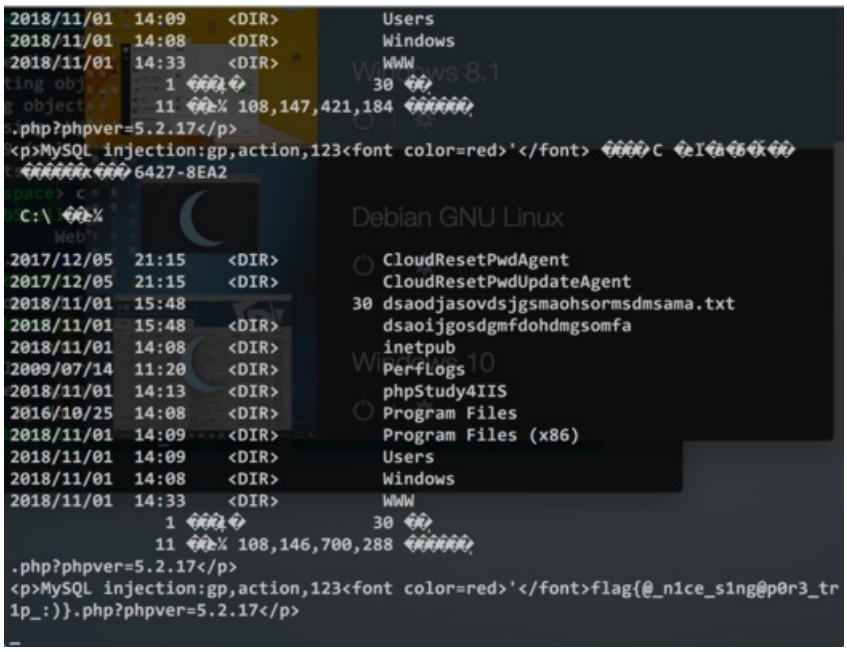
...: continue

...: if r.content not in ans:

...: print r.content

...: ans.append(r.content)

结果发现这个文件的http response在一直变化，看着看着flag就出来了....



然后提交拿了2血，几秒之后看到De1ta也交了flag，感谢De1ta的WEB大佬们送的火箭！

解题过程

当天下午完整打通了exp过程，分为三个部分：

找到注入点，偷数据

构造cookie，拿到admin权限

通过更新功能写shell到本地，读flag

SQL注入

CMS对SQL注入的防御策略

kss_inc/function github link

实现了多种过滤方案，然后在SQL语句拼接取参时，通过传入参数指定取参的位置(GET/POST/COOKIE)和过滤方案(sql/sqljs/num等)

外部取参的代码示例：

$_obfuscate_iJWMjliVi5OGjJOViY2Li48 = _obfuscate_i4mlkpOGkomKiouRhoaMh5I("out_trade_no", "pg", "sql", "");

意思是从POST/GET(pg)中取出参数out_trade_no的值，然后通过sql过滤器的检查后，赋值到i48变量。

先简单看下过滤器的正则：

case "sql" :

if (preg_match("/select|insert|update|delete |union|into|load_file|outfile|char|0x[0-9a-f]{6}|\\.\\/|\\/*|'/i", $_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs))

{

ob_clean( );

$_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs = preg_replace("/(select|insert|update|delete |union|into|load_file|outfile|char|0x[0-9a-f]{6}|\\.\\/|*|')/i", "$1", $_obfuscate_ipCJlJOSlJSQkYqNlYqKlIs);

exit("

MySQL injection:".$_obfuscate_IlyOioeNkY6Vj4qPkJGMiJQ.",".$_obfuscate_iYyTho_HlJCOh4yRj4ePj4k.",".$_obfusca

");
过滤了'，然后匹配到这些危险字符时，会将参数带到html回显，使response可控(XSS敏感)。

正则写的没啥问题，接下来两个方向：

找到忘记使用过滤器直接传参的场景

找到外部没用'包裹的拼接，构造注入

注入构造

kss_inc/payapi_return2.php是一个外部支付功能。其中的chinabank,e138两种支付方式均存在"未使用过滤器"直接传参的漏洞。

else if ($_obfuscate_kYyPkY_PkJKVh4qGjJGlio4 == "chinabank")

{

$_obfuscate_kpGPh4mNh46SkZONh4eLlJU = "";

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

$_obfuscate_iJWMjliVi5OGjJOViY2Li48 = $_obfuscate_k42NkY2RkoiNjJCKlZSKilg;

```
$_obfuscate_iluQkYaUioqGIl6ljluMiI8$ = trim( $_POST['v_pstatus'] );
$_obfuscate_jpGJk5SSkJOIk4iQiI_OhpU$ = trim( $_POST['v_amount'] );
$_obfuscate_lluQk5OGjpKVjY6UiI_QjJM$ = $_obfuscate_jpGJk5SSkJOIk4iQiI_OhpU$;
$_obfuscate_hpCRlJCSjl6Ki5WSipCLkpQ$ = trim( $_POST['v_moneytype'] );
$_obfuscate_lJSPjJCOi5CliJSSkZWNh4Y$ = trim( $_POST['remark1'] );
$_obfuscate_iImJjYmQjYyOjluVkluMjls$ = trim( $_POST['v_md5str'] );
if ( $_obfuscate_iluQkYaUioqGIl6ljluMiI8$ == "20" )
{
$_obfuscate_i5CMioaGil6ShomNiluKjJE$ = "TRADE_FINISHED";
}
else
{
$_obfuscate_i5CMioaGil6ShomNiluKjJE$ = "WAIT_BUYER_PAY";
}
}
else if ( $_obfuscate_kYyPkY_PkJKVh4qGjJGlio4$ == "e138" )
{
$_obfuscate_kpGPh4mNh46SkZONh4eLlJU$ = "";
$_obfuscate_k42NkY2RkoiNjJCKlZSKilg$ = trim( $_POST['SerialNo'] );
$_obfuscate_iJWMjliVi5OGjJOViY2Li48$ = $_obfuscate_k42NkY2RkoiNjJCKlZSKilg$;
$_obfuscate_iluQkYaUioqGIl6ljluMiI8$ = trim( $_POST['Status'] );
$_obfuscate_jpGJk5SSkJOIk4iQiI_OhpU$ = trim( $_POST['Money'] );
$_obfuscate_lluQk5OGjpKVjY6UiI_QjJM$ = $_obfuscate_jpGJk5SSkJOIk4iQiI_OhpU$;
$_obfuscate_iImJjYmQjYyOjluVkluMjls$ = trim( $_POST['VerifyString'] );
if ( $_obfuscate_iluQkYaUioqGIl6ljluMiI8$ == "2" )
{
$_obfuscate_i5CMioaGil6ShomNiluKjJE$ = "TRADE_FINISHED";
}
else
{
$_obfuscate_i5CMioaGil6ShomNiluKjJE$ = "WAIT_BUYER_PAY";
```

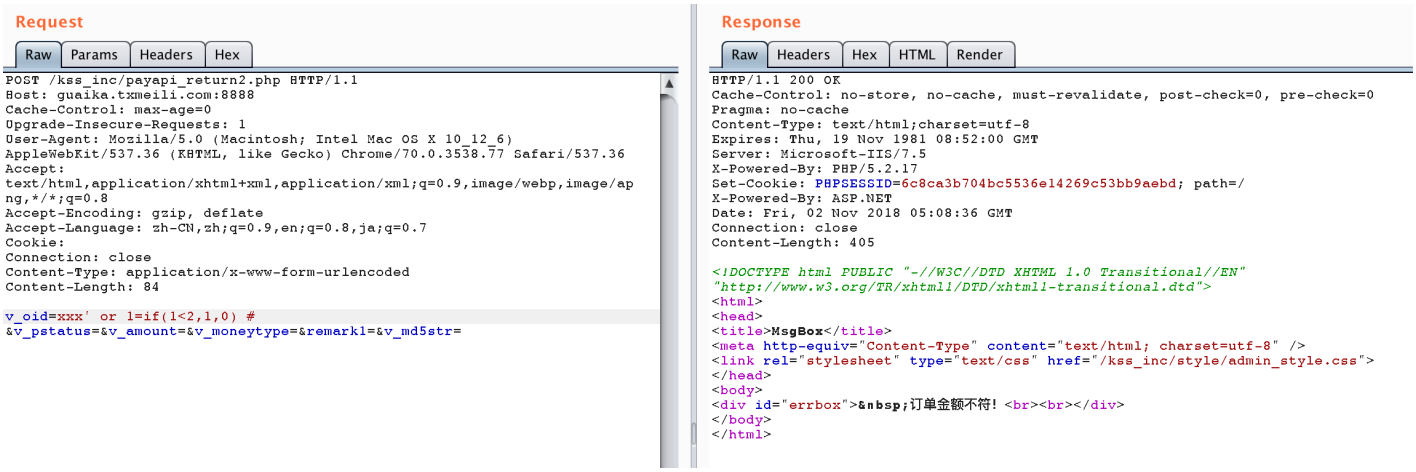
}

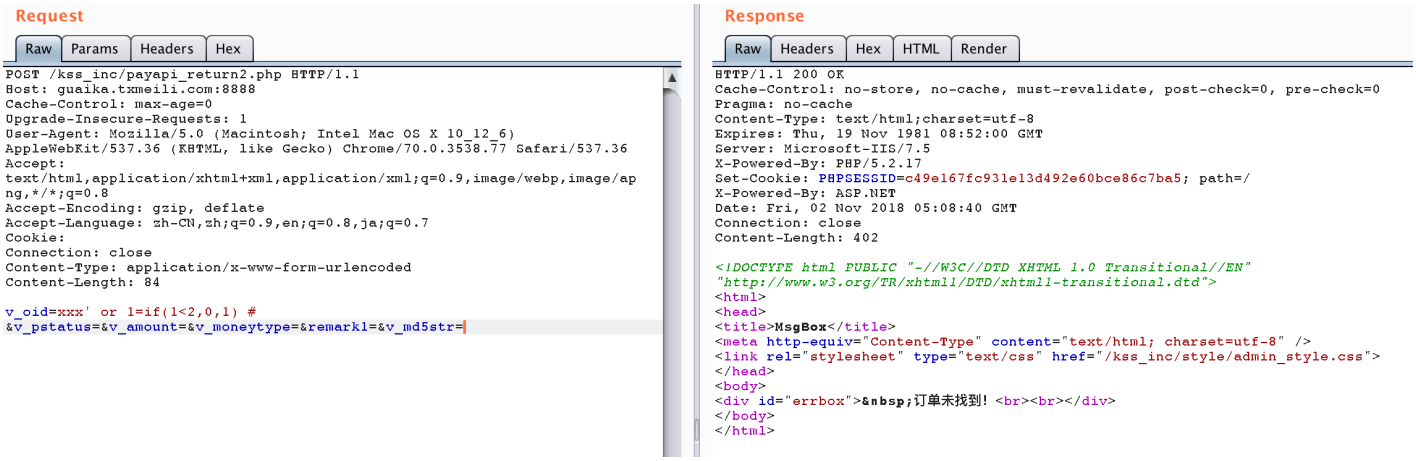
}

SQL执行时GP被带入i48变量。

$_obfuscate_IZGQj4iOj4mTIZGNjZGUj5E\u{fffd} = $_obfuscate_jIaUileSjZWKIlqLklqOioc\u{fffd}->_obfuscate_iY6OkJCRkY2PjpCPk5CRkJA\u{fffd}("select * from kss_tb_order where ordernum='".$_obfuscate_iJWMjliVi5OGjJOViY2Li48\u{fffd}."');

这里因为没有过滤，可以'闭合然后构造一个布尔盲注。





Cookie构造

在使用admin_upload.php写shell之前，有权限校验，可使用从数据库中注出的数据，按源码的验证逻辑构造出cookie，拿到admin权限。

cookie构造逻辑在kss_inc/db_function.php line 300。

有两个k-v需要构造，下图红框部分为注入跑出来的数据，蓝框部分从源码配置文件里拿到:

webshell写入

回到最开始提到的从远程服务器更新代码的逻辑：

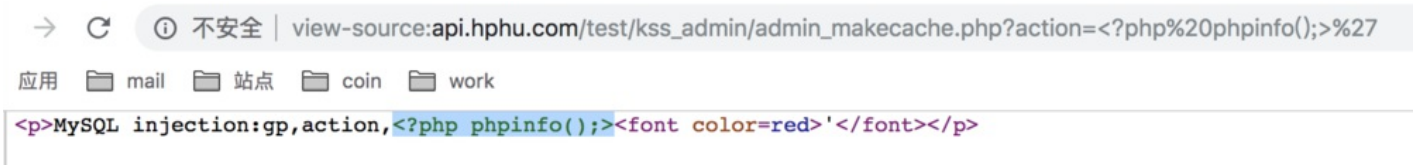
$_obfuscate_koiKkliPjl6UkYeRIlqNhoc\u{fffd} = _obfuscate_lY6Gk5KMkYmPjlyPhpCOlYc\u{fffd}(
"http://api.hphu.com/import/".$_obfuscate_koaSiYqGjlqMiZSLk4uGiZU\u{fffd}.".php?
phpver=".PHP_VERSION."&webid=".WEBID."&rid=".time( ), 300);

URL里面拼接的变量是外部可控的，我们在这个主站的test目录下发现了一套Demo的CMS：

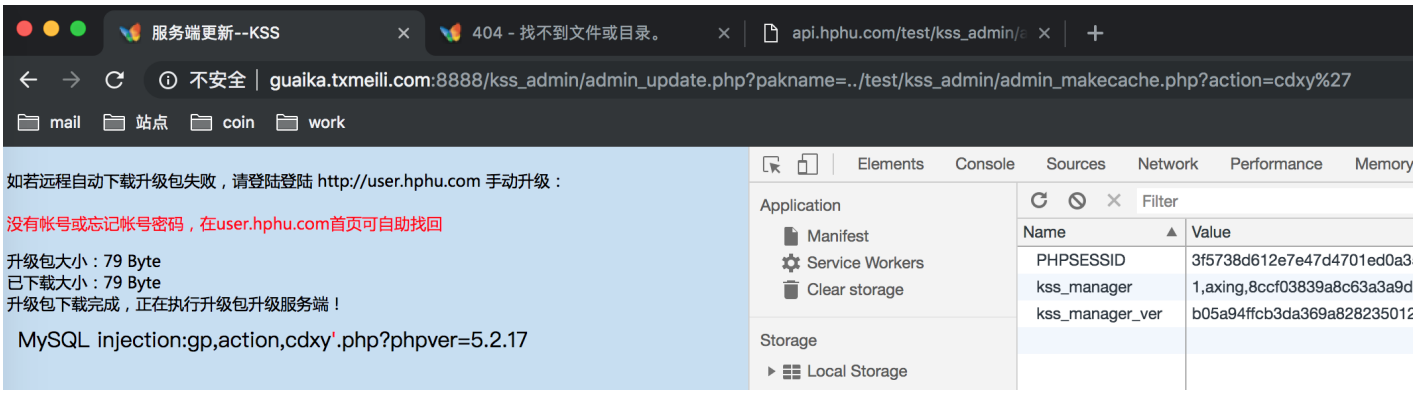
http://api.hphu.com/test/kss_admin/index.php

回想之前的SQL注入过滤机制，我们可以触发这个机制，将php代码写入http回显，然后admin_upload.php通过curl读内容时会将页面中的php代码写入_webup.php，完成webshell植入。

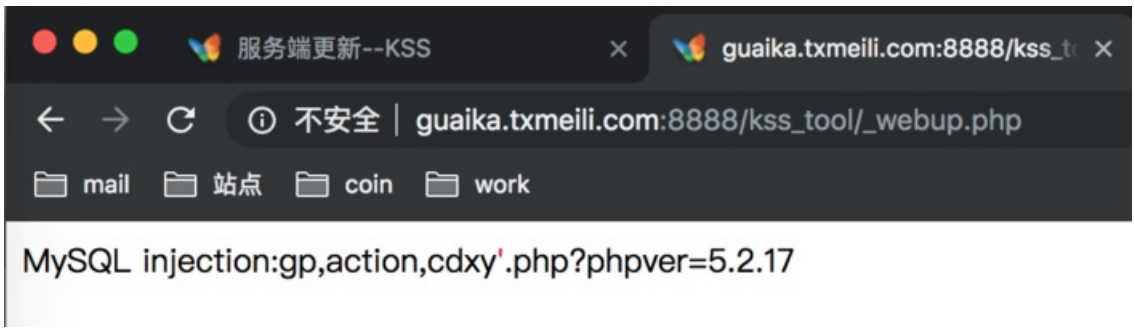
主站回显构造：



Exp构造(带上之前构造好的cookie)：



最终将可控回显写入_webup.php



然后通过webshell读到C盘根目录下的flag文件。

给 @r3kapig 队友递茶

给提供思路的队内WEB大佬递茶 @麦香 @zzm @hear7v @lynahex @n0b0dy

给本题做的比我们快的 @Dubhe 和 @De1ta 两队WEB大佬递茶

给出题人递茶 @RicterZ

总排行榜

排名	战队名	解题数	总分	Web 总得分：14167.85 解题次数：21		Pwn 总得分：25534.11 解题次数：38				Misc 总得分：35207.07 解题次数：56				
				PU.	be.	no.	re.	TSH	Z	Ba.	Pa.	My.	St.	Ba.
1	r3kapig	11	7851.21											
2	0ops	10	7021.2											
3	Dubhe	9	6010.49											
4	Whitzar	8	5345.2											
5	Nu1L	8	5113.42											
6	De1ta	7	4763.74											
7	kn0ck	7	4514.74											
8	Lancet	7	4451.38											
9	VidarTe	6	3733.89											
10	PwnThy	5	3229.9											
11	SixStars	5	3201.64											
12	CNSS	5	3139											
13	XMan	5	3064											
14	AAA	5	3033.67											
15	Redbud	4	2594											
16	SU	4	2425											
17	lilac	4	2352.55											
18	X1ct34r	3	1699											
19	ROIS	2	1365											

截图(Alt + A)

由赛宁网络靶场平台(CP-CR)驱动