

Hgame-Writeup Week2

WEB

ID40 从0开始LFI之0

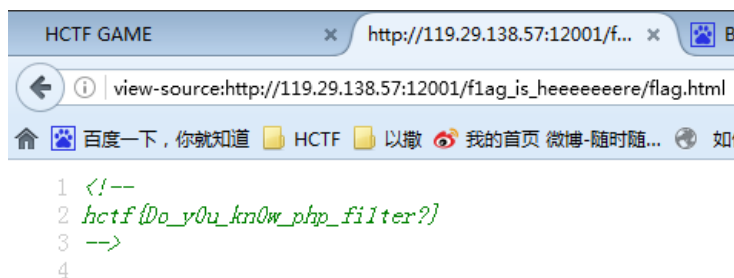
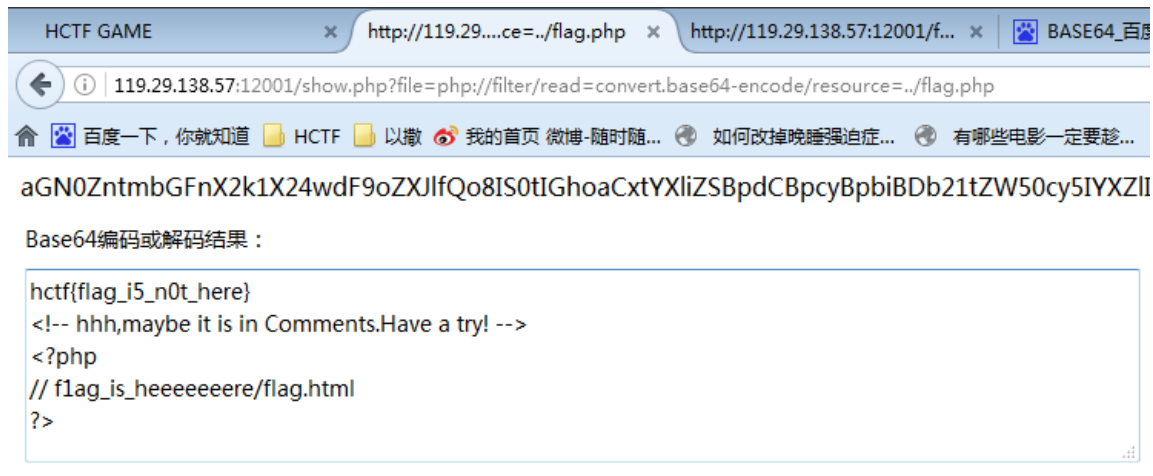
送分题 flag在../flag.php http://119.29.138.57:12000/

查看html源代码发现有文件包含漏洞 ?file=../flag.php 得到flag

ID41 从0开始LFI之1

flag在哪呢? http://119.29.138.57:12001/

同样文件包含漏洞,但是flag.php里没有flag,利用 ?file=php://filter/read=convert.base64-encode/resource=../flag.php查看base64加密后的php源代码,解密后得到flag.html地址,得到flag

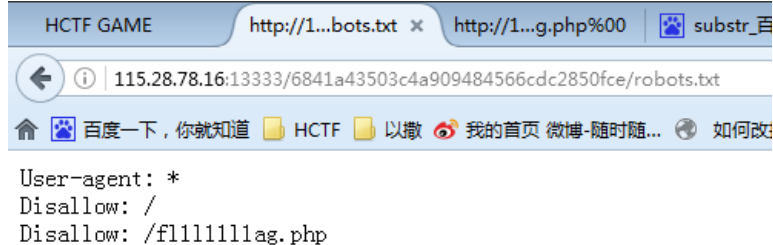


ID43 Explorer的图包

LoRexar一直念念不忘Explorer的图包,一气之下Explorer把图放在了服务器上,当LoRexar又需要的时候可以自己去下。

115.28.78.16:13333/6841a43503c4a909484566cdc2850fce/

查看robots.txt,发现好东西,利用文件下载漏洞下载,发现虽然文件名是对的,内容是index.php的,估计有过滤,绕过去就行了





PENTEST

ID45 我是最简单的渗透题

网上查了很多万能密码，啥也没管一个一个试过去就登陆成功了

后来回头看了一下万能密码原理，用的是SQL注入

CRYPTO

ID34 密码学教室进阶（五）

不多说了，贴上代码，先去<http://www.factordb.com/>分解n，然后根据p,q,e算出d，解出明文

```
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m

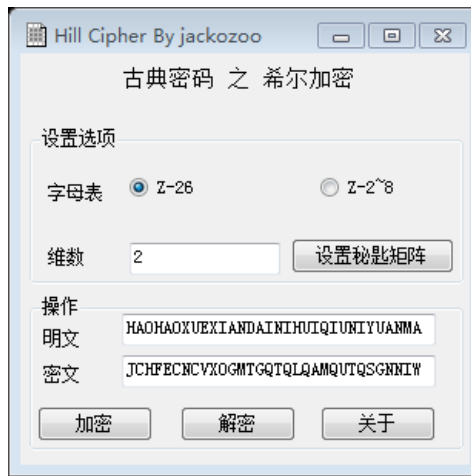
p=input("p=")
q=input("q=")
e=input("e=")
c=input("c=")
d=modinv(e,(p-1)*(q-1))
m=pow(c,d,p*q)
print m
```

ID55 进击的 Crypto [0]

<http://www.factordb.com/>发现无法分解，给了多个n，但是e相同，在网上看到过例题，是寻找不同n之间的公因数，随后解出p,q，代码如下

```
def gcd(a, b):
    if a < b:
        a, b = b, a
    while b != 0:
        temp = a % b
        a = b
        b = temp
    return a
n1=input("n1=")
n2=input("n2=")
print gcd(n1,n2)
```

ID42 密码学教室进阶（六）



一开始手工算的，没做出来，后来网上找了个解密工具，成功了

RE

ID48 re从零开始的逆向之旅：Gold Miner

蹭了一道RE题，纯属玩出来的



MISC

ID39 explore的奇怪番外2

百度了工作量证明，根据nc后的反应，是要求我给出一个长度为100的字符串，且该字符串使用sha256加密后的前六位与它给我的相同，贴上我的

```

1  import socket
2  import time
3  import hashlib
4
5  sha = hashlib.sha256()
6  sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7  sock.connect(('121.42.25.113', 20001))
8  k = sock.recv(1024)
9  print k
10 k = sock.recv(1024)
11 print k
12 from random import Random
13 def random_str(randomlength):
14     str = ''
15     while len(str) < randomlength:
16         str += random_str(1)
17     return str
18 def sha256(check):
19     text = ''
20     for a in range(0, 127):
21         for b in range(0, 127):
22             for c in range(0, 127):
23                 for d in range(0, 127):
24                     list = ['0']*96+[chr(a), chr(b), chr(c), chr(d)]
25                     text = ''.join(list)
26                     sha = hashlib.sha256()
27                     sha.update(text)
28                     #print sha.hexdigest()[0:6], a, b, c, d
29                     if(sha.hexdigest()[0:6]==check):
30                         return text
31
32     return text
33 ppp=raw_input()
34 text0=sha256(ppp)
35 print text0
36 sock.send(text0 + "\n")
37 k = sock.recv(1024)
38 print k
39 k = sock.recv(1024)
40 print k
41
42

```

Remove redundant parentheses [more...](#) (Ctrl+F1)

socket

ID61 我是一个有格调的misc题目

摸索了一下Wireshark的用法，使用过滤器查找hctf，得到flag

Wireshark packet capture interface showing a list of network packets. The filter 'hctf' is applied. The following table summarizes the visible packets:

No.	Time	Source	Destination	Protocol	Length	Info
62	0.000000	80	38502	TCP	60	Keep-Alive ACK Seq=456 Ack=414 Win=64240 Len=0
56	0.000000	38478	80	TCP	60	Keep-Alive Seq=413 Ack=558 Win=30078 Len=0
62	0.000000	80	38478	TCP	60	Keep-Alive ACK Seq=558 Ack=414 Win=64240 Len=0
56	0.000000	38496	80	TCP	60	Keep-Alive Seq=413 Ack=457 Win=30016 Len=0
62	0.000000	80	38496	TCP	60	Keep-Alive ACK Seq=457 Ack=414 Win=64240 Len=0
62	0.000000	192.168.186.2	192.168.186.1	TCP	60	Who has 192.168.186.2? Tell 192.168.186.1
563	0.000000	80	39930	HTTP	100	GET /wp/?s=hctf%7Bwh4t_d0_y0u_w4nt%3F%3F%7D&submit=%E6%90%9C%E7%B4%A2 HTTP/1.1
62	0.000000	80	39930	TCP	60	ACK Seq=498 Ack=1576 Win=64240 Len=0

Additional information visible in the packet details pane for packet 563:

- Length: 100
- Info: GET /wp/?s=hctf%7Bwh4t_d0_y0u_w4nt%3F%3F%7D&submit=%E6%90%9C%E7%B4%A2 HTTP/1.1
- Request Method: GET
- Request URI: /wp/?s=hctf%7Bwh4t_d0_y0u_w4nt%3F%3F%7D&submit=%E6%90%9C%E7%B4%A2
- HTTP Version: 1.1
- User-Agent: Mozilla/5.0 (Windows NT 6.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36
- Host: 192.168.186.1
- Accept: */*
- Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
- Accept-Encoding: gzip, deflate
- Connection: keep-alive