南京邮电大学网络攻防平台逆向writeup之[WxyVM]

原创

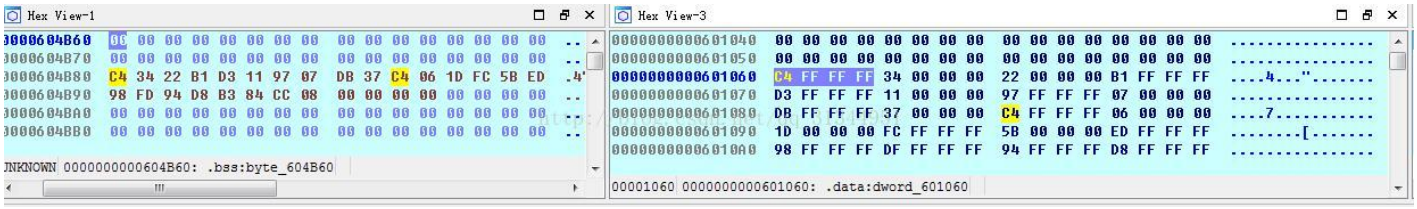
巫师54 于 2017-08-01 14:10:38 发布 1720 收藏 1

#1

找到主函数，看上去很简单，把输入的字符串拿去一个函数作运算，然后拿字节和一个固定的dword对比

```
puts("[WxyVM 0.0.1]");
puts("input your flag:");
scanf("%s", &byte_604B80);
v4 = 1;
sub_4005B6();
if ( strlen(&byte_604B80) != 24 )
  v4 = 0;
for ( i = 0; i <= 23; ++i )
{
  if ( *(&byte_604B80 + i) != dword_601060[i] )
    v4 = 0;
}
if ( v4 )
  puts("correct");
else
  puts("wrong");
return 0LL;
}
```

坑点1：动态调试的时候才找到，是拿byte和dword对比，这里卡了好久，终于找到拿出来对比的字符串了。



#2

进入sub_4005b6();这个就是加密函数，byte_6010C0 是一个类似hash表的东东，有1500字节长，为了方便读取截出来保存为一个二进制文件，winhex打开看一下

```
for ( i = 0; i <= 14999; i += 3 )
{
  v0 = byte_6010C0[(signed __int64)i];
  v3 = byte_6010C0[(signed __int64)(i + 2)];
  result = v0;
  switch ( v0 )
  {
    case 1u:
      result = byte_6010C0[(signed __int64)(i + 1)];
      *(&byte_604B80 + result) += v3;
      break;
    case 2u:
      result = byte_6010C0[(signed __int64)(i + 1)];
      *(&byte_604B80 + result) -= v3;
      break;
    case 3u:
      result = byte_6010C0[(signed __int64)(i + 1)];
      *(&byte_604B80 + result) ^= v3;
      break;
    case 4u:
      result = byte_6010C0[(signed __int64)(i + 1)];
      *(&byte_604B80 + result) *= v3;
      break;
    case 5u:
      result = byte_6010C0[(signed __int64)(i + 1)];
      *(&byte_604B80 + result) ^= *(&byte_604B80 + byte_6010C0[(signed __int64)(i + 2)]);
      break;
```


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	01	10	25	03	0D	0A	02	0B	28	02	14	3F	01	17	3C	01
00000010	00	69	01	12	3F	02	0E	77	03	15	53	02	0E	7D	03	05
00000020	0A	02	04	55	02	15	33	02	15	05	01	05	2F	03	07	43
00000030	01	11	39	03	0D	27	01	05	1E	03	04	3C	01	13	1E	03
00000040	01	78	01	00	20	02	0F	53	03	14	2B	03	14	28	03	0A
00000050	19	03	12	60	01	05	7E	03	0F	20	01	0F	58	02	11	51
00000060	01	0B	24	01	17	79	01	0E	4A	03	10	67	02	16	5C	03
00000070	09	6D	01	17	30	02	0A	2C	03	07	3F	03	07	43	01	04

3#

读这段代码的大致意思，每次循环取0，1，2三位，第一位拿来select case，第2位拿来定位，第3位拿来运算，比如03，04，05的值分别是[03,0d,0a]，在select case 中的运算公式就应该为

byte_604b80[0d]^=0a

坑点2：运算过程中可能会溢出，这里卡了好几天，NND

#4

答案就不给了吧，看一下提交成功的截图