实验吧逆向defcamp--WP

转载

weixin_30408675 于 2017-12-01 16:59:00 发布 35 收藏
文章标签： 数据结构与算法
原文链接：http://www.cnblogs.com/ha2ha2/p/7943339.html
版权
题目地址:http://www.shiyanbar.com/ctf/2020

拖到IDA里进行反汇编查看伪代码

```
for ( i = 1; i <= 10; ++i )
{
  v3 = malloc(0x10uLL);
  *(_DWORD *)v3 = i;
  v3[4] = *(_DWORD *)v3 + 109;
  a3 = (char **)qword_601080;
  *((_QWORD *)v3 + 1) = qword_601080;
  qword_601080 = (__int64)v3;
}
```

循环结束后，建立了一个链表。一共十个元素。

qword_601080里的值是个地址，该地址指向链表第一个元素。

链表的每一个数据域的前一个双字里是序号（从0Ah一直到01h）后一个双字存放着数值（对应数值与序号的关系是数值=109+序号值）。

```
printf("Enter the password: ", a2, a3);
if ( fgets(&s, 7, stdin) )
{
  if ( (unsigned int)sub_40074D((__int64)&s) )
  {
    puts("Incorrect password!");
    result = 1LL;
  }
  else
  {
    puts("Nice!");
    result = 0LL;
  }
}
```

查看40074D函数

```
signed __int64 __fastcall sub_40074D(__int64 a1)
{
  signed int i; // [sp+8h] [bp-50h]@1
  signed int j; // [sp+8h] [bp-50h]@9
  int v4; // [sp+Ch] [bp-4Ch]@2
  __int64 v5; // [sp+10h] [bp-48h]@2
  __int64 v6; // [sp+18h] [bp-40h]@1
  __int64 v7; // [sp+20h] [bp-38h]@1
  __int64 v8; // [sp+28h] [bp-30h]@1
  int v9; // [sp+38h] [bp-20h]@1
  int v10; // [sp+3Ch] [bp-1Ch]@1
  int v11; // [sp+40h] [bp-18h]@1
  int v12; // [sp+44h] [bp-14h]@1
  int v13; // [sp+48h] [bp-10h]@1
  int v14; // [sp+4Ch] [bp-Ch]@1

  v6 = 0LL;
  v7 = 0LL;
  v8 = 0LL;
  v9 = 5;
  v10 = 2;
  v11 = 7;
  v12 = 2;
  v13 = 5;
  v14 = 6;
  for ( i = 0; i <= 5; ++i )
  {
    v5 = qword_601080;
    v4 = 0;
    while ( v5 )
    {
      if ( *(_BYTE *)(v5 + 4) == *(_BYTE *)(i + a1) )
      {
        v4 = *(_DWORD *)v5;
        break;
      }
      v5 = *(_QWORD *)(v5 + 8);
    }
    *((_DWORD *)&v6 + i) = v4;
  }
  for ( j = 0; j <= 5; ++j )


  {
    if ( *((_DWORD *)&v6 + j) != *(&v9 + j) )
      return 1LL;
  }
  return 0LL;
}
```

第一个for进来后5次循环

每次循环都将我们输入的password的对应字符与链表数据域中的数值进行比对，如果相等，就会把该数值对应的序号存放在v6数组中。

第二个for就是v6数组与v9数组进行比对。

v9数组 5 2 7 2 5 6

所以对应数值 114 111 116 111 114 115

即password:rotors

转载于:https://www.cnblogs.com/ha2ha2/p/7943339.html