ctf.360.cn第二届，逆向部分writeup——第四题

转载

weixin_33726313 于 2014-11-26 15:20:47 发布 76 收藏
原文链接：http://blog.51cto.com/cugou/1582824

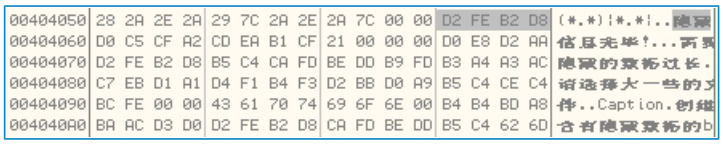
题目：见附件

这题开始有些小复杂了。

> 提示：压缩包中给定的这张图中含有答案KEY，而加密该图片的程序也在该压缩包中,请逆向该EXE，提取其中的KEY

运行程序，可以看见界面如下

注意看“隐藏信息完毕！”字符串的位置



直接搜索push 40405c，找到代码位置401a48。

网上翻找到代码块的入口地址4017a0。很显然这是一个非常长的函数，使用IDA进行静态分析。

```
v2 = CreateFileA(*((LPCSTR *)v1 + 24), 0x80000000u, 1u, 0, 3u, 0x80u, 0);
if ( v2 == (HANDLE)-1 )
{
  result = CWnd::MessageBoxA(v1, &unk_4040D4, 0, 0);
}
else
{
  v4 = CreateFileA(*((LPCSTR *)v1 + 26), 0x80000000u, 1u, 0, 3u, 0x80u, 0);
  hObject = v4;
  if ( v4 == (HANDLE)-1 )
  {
    result = CWnd::MessageBoxA(v1, &unk_4040BC, 0, 0);
  }
  else
  {
    v33 = CreateFileA(*((LPCSTR *)v1 + 25), 0x40000000u, 1u, 0, 2u, 0x80u, 0);
    if ( v33 == (HANDLE)-1 )
    {
```

```
    result = CWnd::MessageBoxA(v1, &unk_40409C, 0, 0);
  }
  else
  {
    SetFilePointer(v2, 2, 0, 0);
    ReadFile(v2, &Buffer, 4u, &NumberOfBytesWritten, 0);
    SetFilePointer(v2, 4, 0, 1u);
    ReadFile(v2, &v35, 4u, &NumberOfBytesWritten, 0);
    SetFilePointer(v2, 0, 0, 0);
    v5 = operator new(Buffer);
    lpBuffer = v5;
    ReadFile(v2, (LPVOID)v5, Buffer, &NumberOfBytesWritten, 0);
    v6 = (int)((char *)v5 + v35);
    v7 = GetFileSize(v4, 0);
    v31 = v7;
    v28 = operator new(v7);
    ReadFile(hObject, v28, v7, &NumberOfBytesWritten, 0);
    v8 = Buffer - v35 - 32;
    if ( 8 * v7 <= v8 )
    {
      v24 = v7;
      v9 = 16;
      do
      {
        LOWORD(v8) = *(_BYTE *)v6 & 1;
        v10 = v24 & 1;
        v8 ^= v10;
        if ( (_WORD)v8 )
        {
          v11 = (rand() & 1) == 0;
          v12 = *(_BYTE *)v6;
          if ( v11 )
            v13 = v12 - 1;
          else
            v13 = v12 + 1;
          *(_BYTE *)v6 = v13;
        }
        v24 >>= 1;
        ++v6;
        --v9;
      }
      while ( v9 );
      v14 = 16;
      v25 = v7 >> 16;
      do
      {
        LOWORD(v10) = *(_BYTE *)v6 & 1;
        v10 ^= v25 & 1;
        if ( (_WORD)v10 )
        {
          v11 = (rand() & 1) == 0;
          v15 = *(_BYTE *)v6;
          if ( v11 )
            v16 = v15 - 1;
          else
            v16 = v15 + 1;
          *(_BYTE *)v6 = v16;
        }
        LOWORD(v25) = (unsigned __int16)v25 >> 1;
        ++v6;
```

```
          --v14;
        }
        while ( v14 );
        v17 = 0;
        v26 = 0;
        if ( v7 )
        {
          do
          {
            v18 = 8;
            v19 = *((_BYTE *)v28 + v17);
            do
            {
              if ( (v19 ^ *(_BYTE *)v6) & 1 )
              {
                v11 = (rand() & 1) == 0;
                v20 = *(_BYTE *)v6;
                if ( v11 )
                  v21 = v20 - 1;
                else
                  v21 = v20 + 1;
                *(_BYTE *)v6 = v21;
              }
              v19 >>= 1;
              ++v6;
              --v18;
            }
            while ( v18 );
            v17 = v26++ + 1;
          }
          while ( v26 < v31 );
        }
        v22 = lpBuffer;
        v23 = v33;
        WriteFile(v33, lpBuffer, Buffer, &NumberOfBytesWritten, 0);
        operator delete((void *)v22);
        operator delete(v28);
        CloseHandle(v2);
        CloseHandle(hObject);
        CloseHandle(v23);
        result = CWnd::MessageBoxA(v30, &unk_40405C, 0, 0);
      }
      else
      {
        result = CWnd::MessageBoxA(v30, &unk_40406C, "Caption", 0);
      }
    }
  }
}
return result;
```

要理解代码，首先要了解bmp文件的格式，可以参考http://www.cnblogs.com/kingmoon/archive/2011/04/18/2020097.html。

实际上，题目的算法是跳过bmp文件头和最前面的32字节的像素，然后每8个字节编码一个所要加密的明文字节。其中用每一个像素字节的最后一位来表示要加密的明文字节的响应位。

代码中+1、-1实际上就是当像素字节的最后一位与明文字节对应位不一致时，修正到相同。

所以知道了算法就知道如何解密：取出藏有密文的字节的最后一位，拼出相应的明文。

提取源代码就不贴了。

转载于:https://blog.51cto.com/cugou/1582824