c#应用破解实战之PDFelement

前言

在知乎上发现一款不错的 pdf 处理软件 PDFelement，功能比较强大，格式转换，pdf编辑，ocr，在网上搜了一遍没找到能用的破解版，于是自己动手。

正文

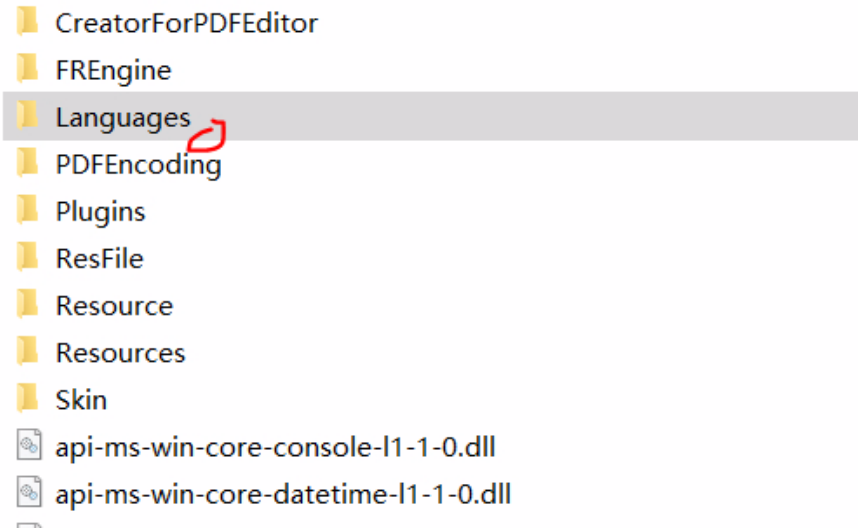
最开始我是拿的 6.0 版本破解的，破解完后发现，这个版本的 bug 特别严重，不能 编辑 pdf（也许是我改坏了， 6.4就没问题），于是后来又下了最新的 6.4. 老版的还有混淆，新版本连混淆都没有。为了总结更多的经验，先写 6.0 版本的破解，然后写 6.4 的破解。

工具

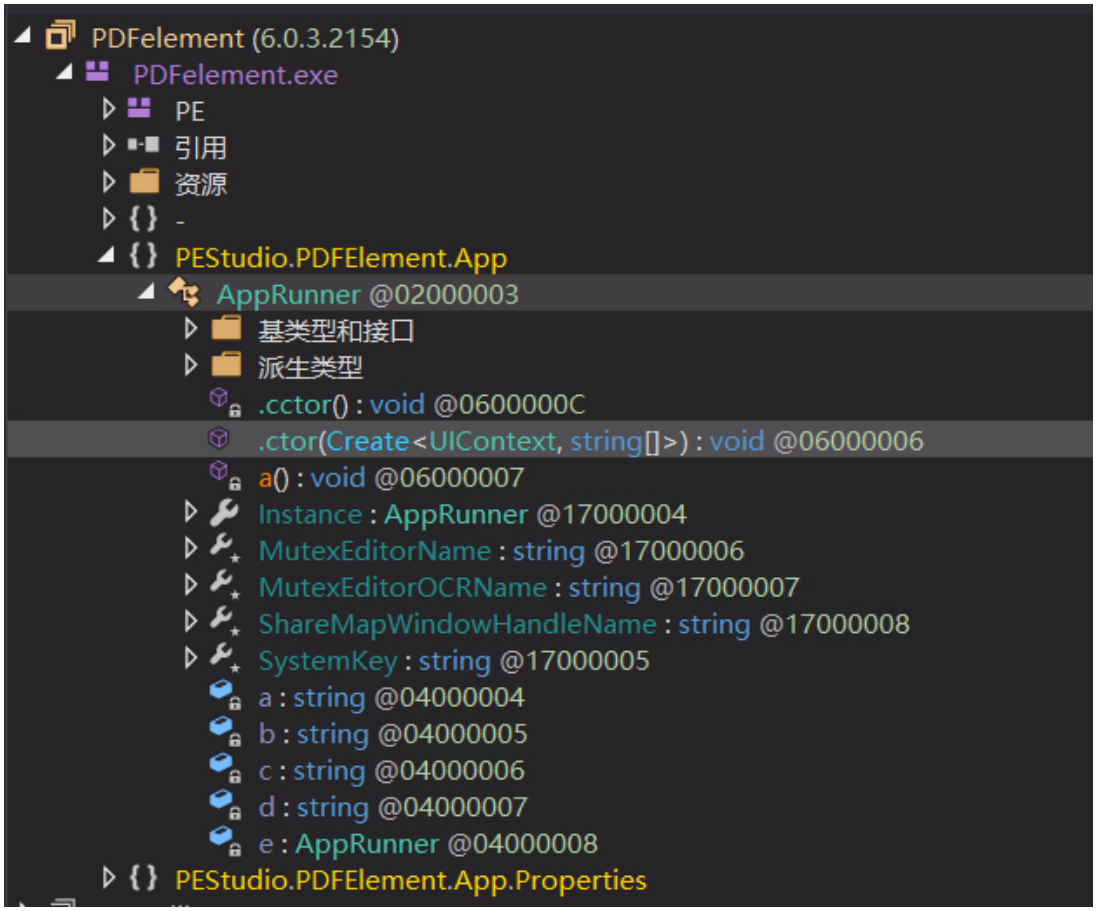
- dnSpy .net 应用反编译，调试工具
- de4dot .net 去混淆

6.0版本破解

首先安装好，对于应用破解可以搜关键字符串，像这种比较大的程序，一般会有单独一个目录放置语言文件便于多语言的支持。

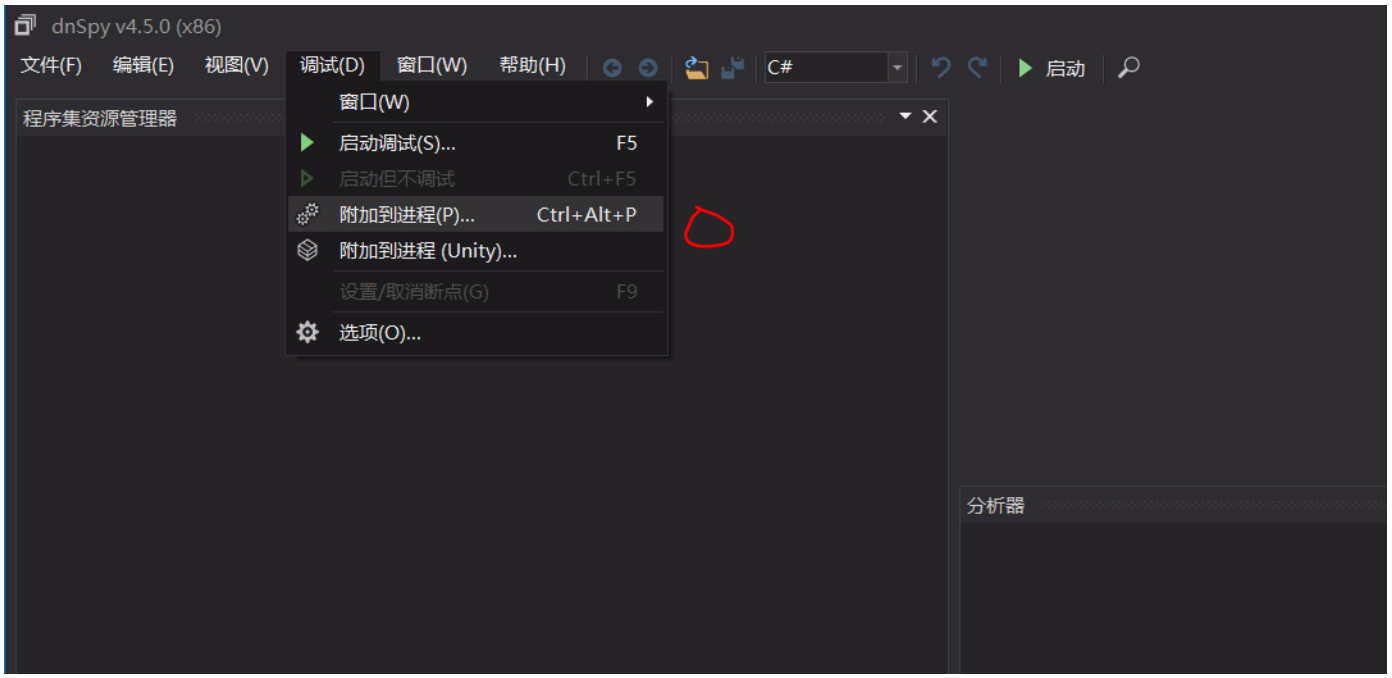


不过该程序的资源文件貌似不是文本格式的，故我放弃了这条路。
接着我去分析了 PDFelement.exe ，想看看能不能根据函数名找到关键的函数。

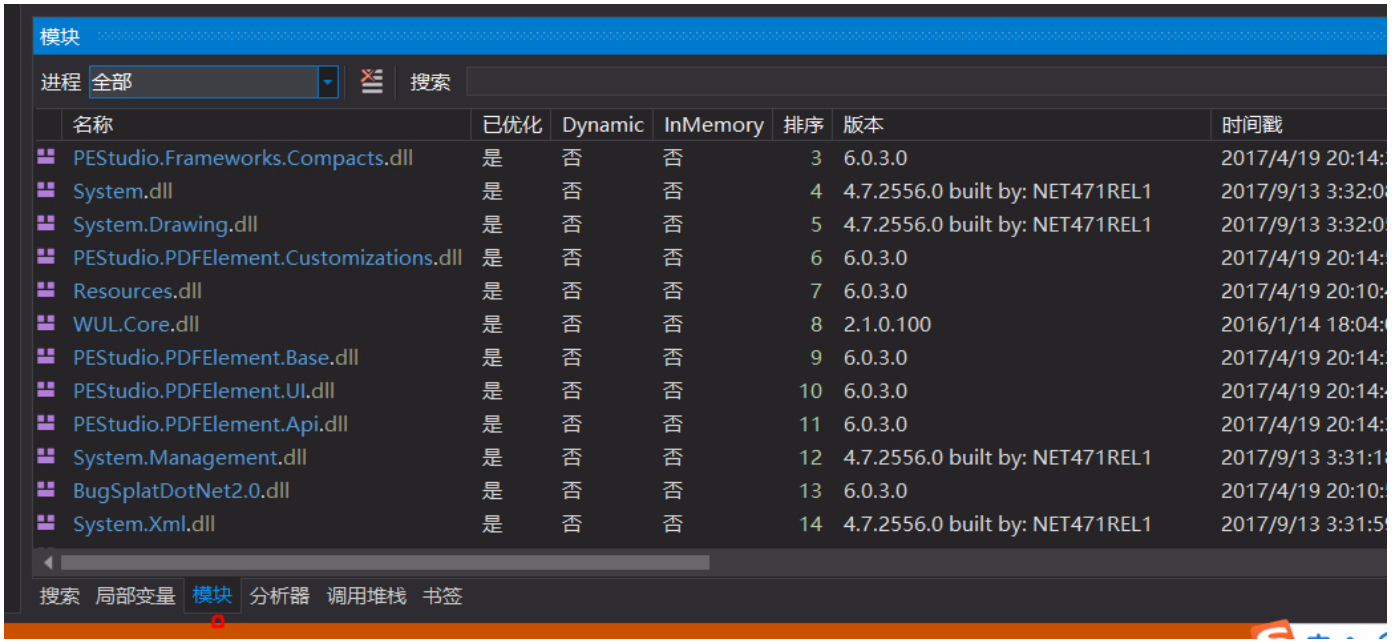


发现它里面没有啥东西，估计关键逻辑都在 dll 里面，于是决定调试他，然后在调试的时候去 dll 里面找找。

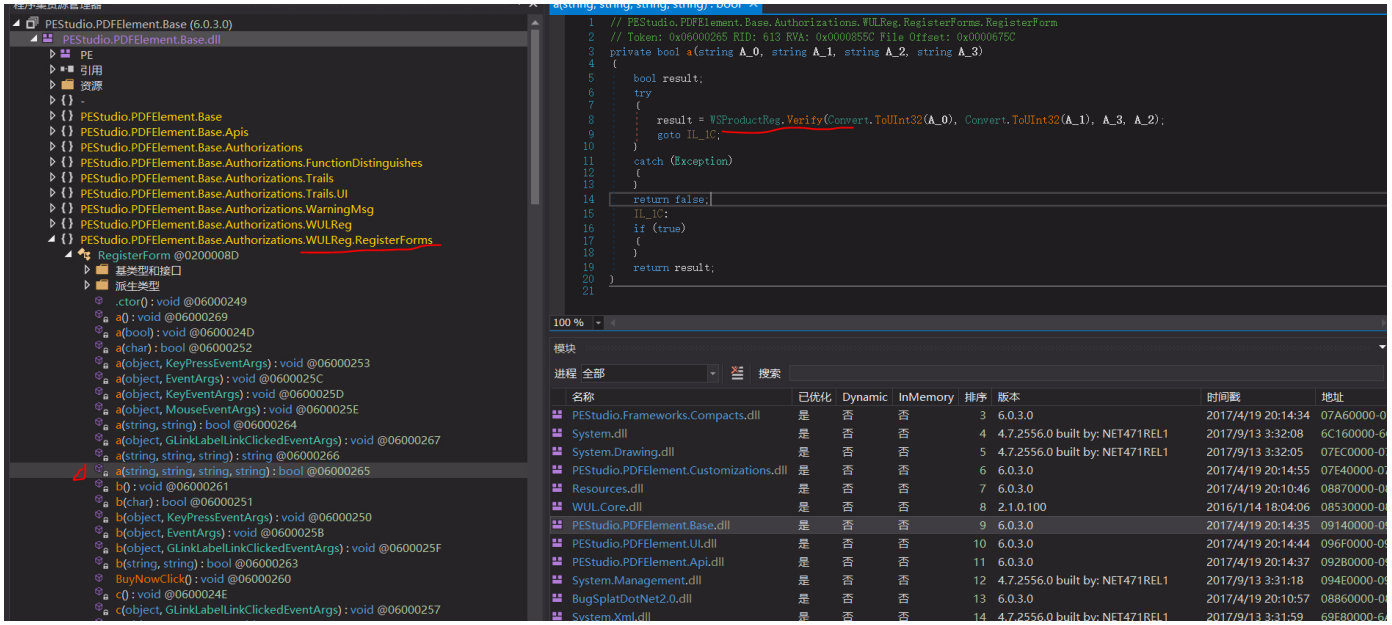
运行程序，然后用 dnspy 附件上去



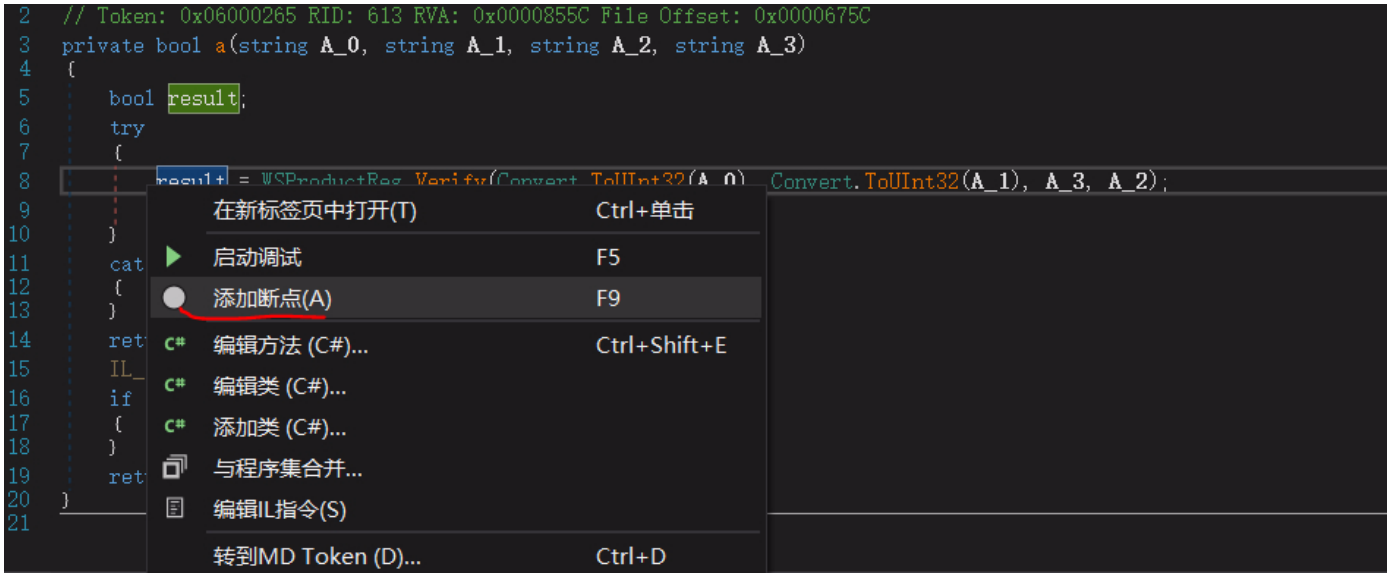
然后在 模块 窗口查看加载的 dll,可以在 调试 - 窗口 -模块 调出该窗口



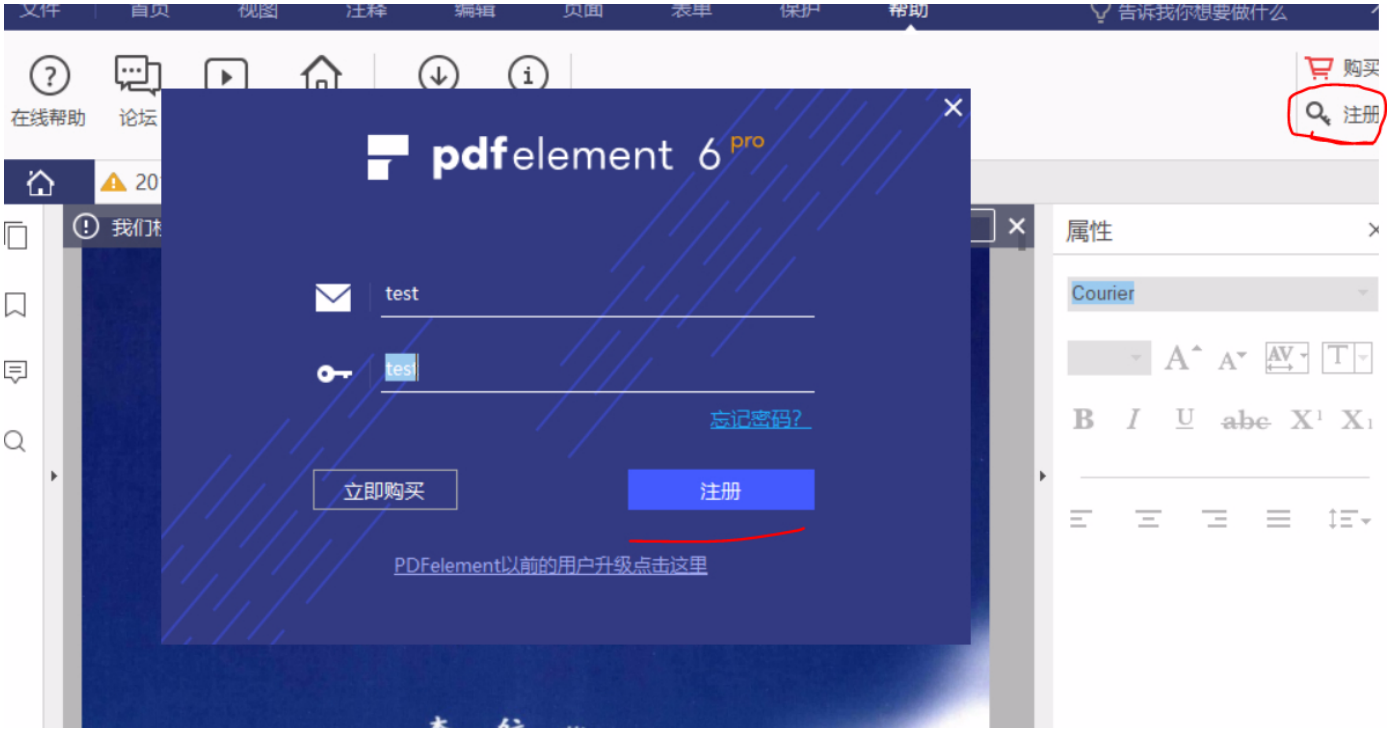
最后在 PEStudio.PDFElement.Base.dll 里面找到了有意思的函数。



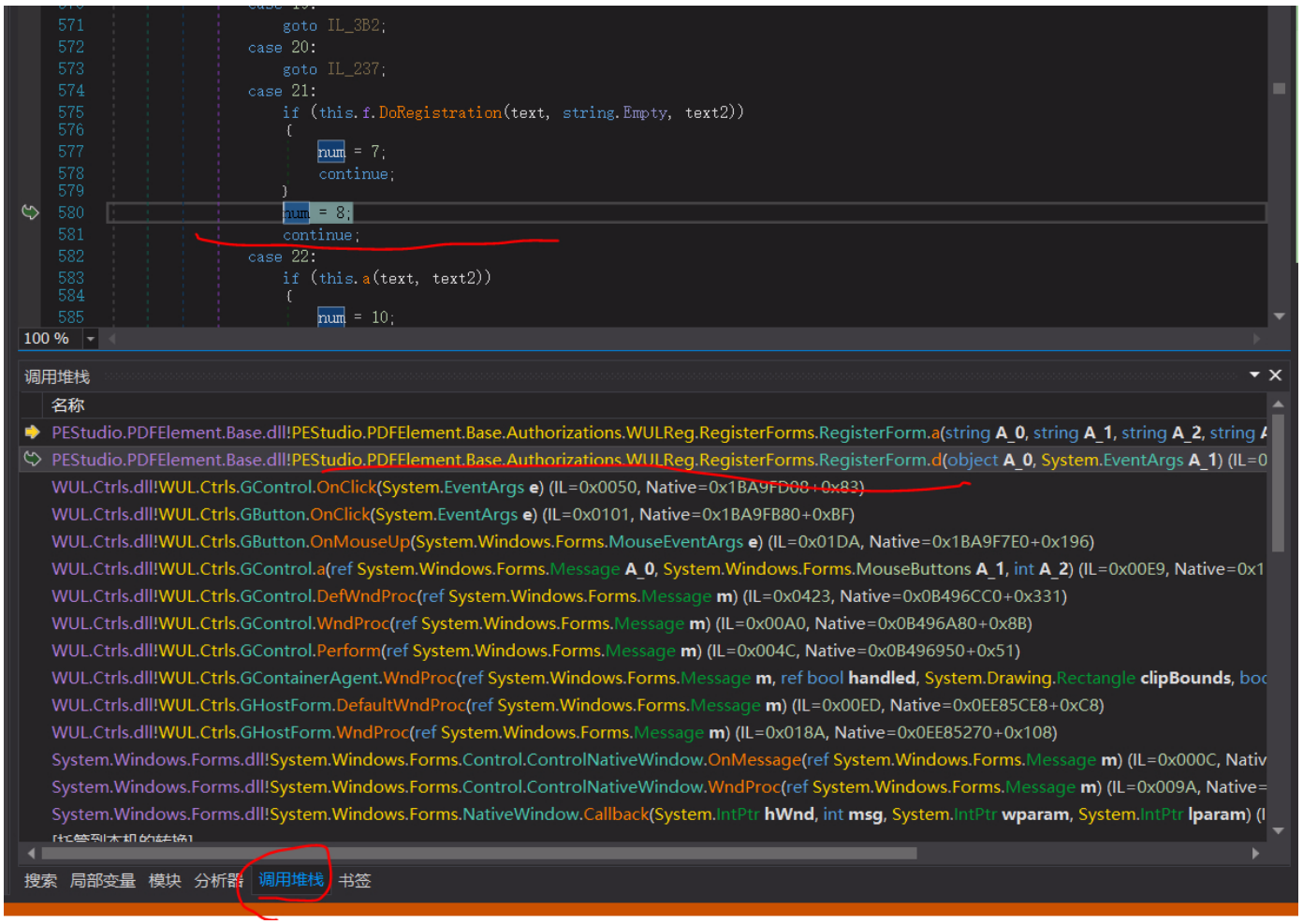
看这些函数名，估计这里是在注册时，用来校验的函数了。在这里下个断点



然后我们去注册，触发这个逻辑然后看看她是在哪里被调用的

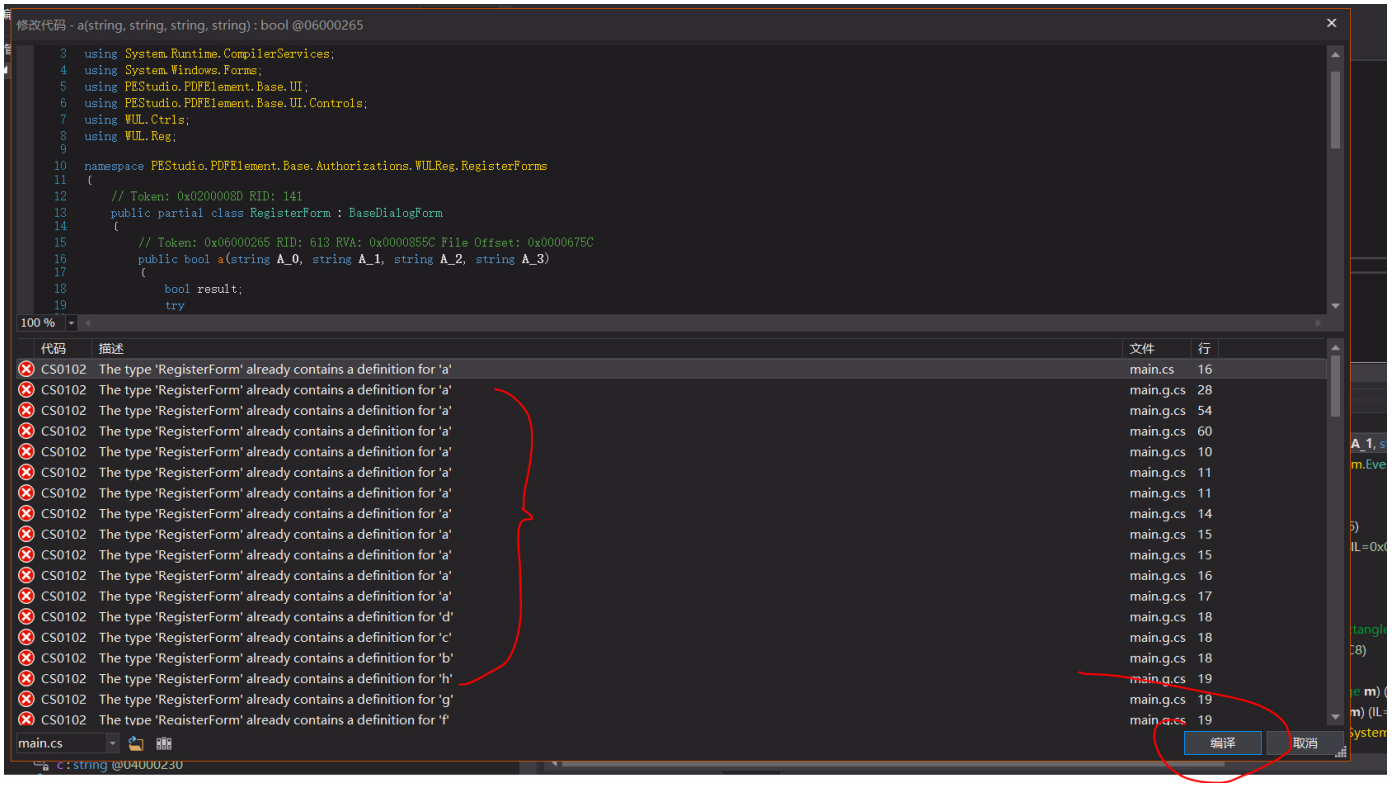


断下来后查看调用堆栈信息



转到调用者这里，发现各种跳转，看起来是做了混淆，于是使用 `de4dot` 去混淆，如果不去混淆的话修改代码保存会出很多的错。

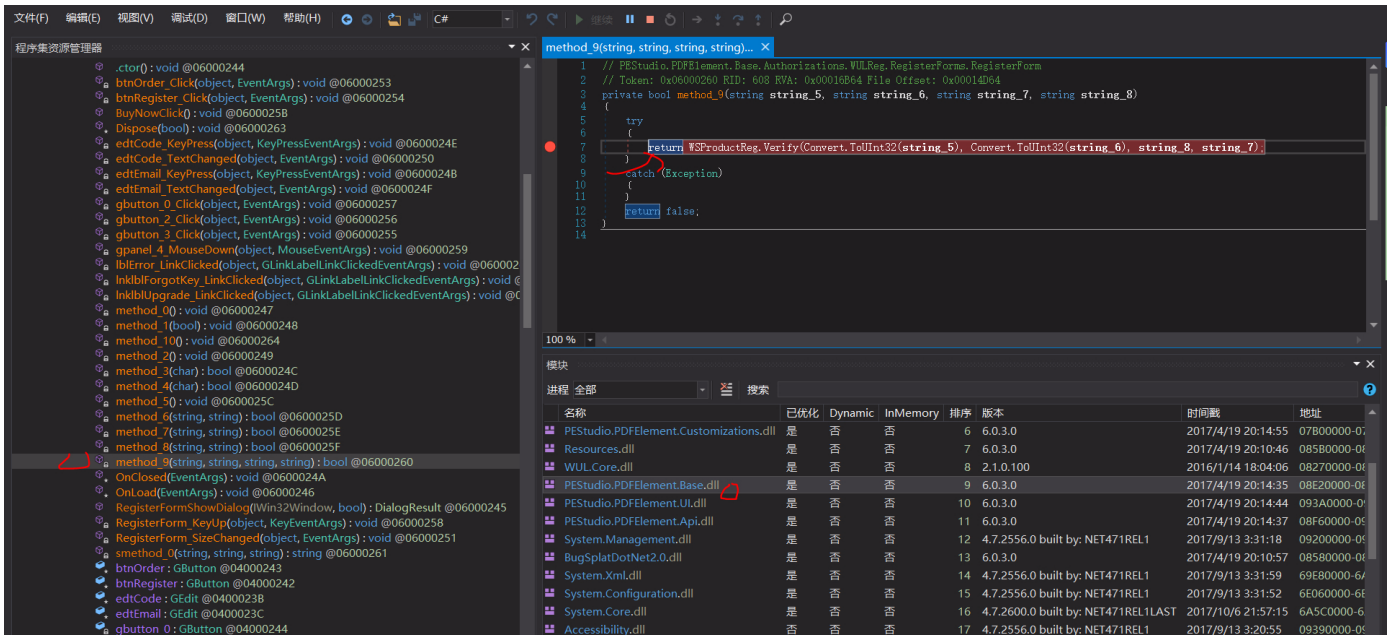
修改代码的方式很简单，进入方法内部，右键-修改方法，然后按照 `c#` 的语法进行修改即可，最后点击 编译 就可以保存了。



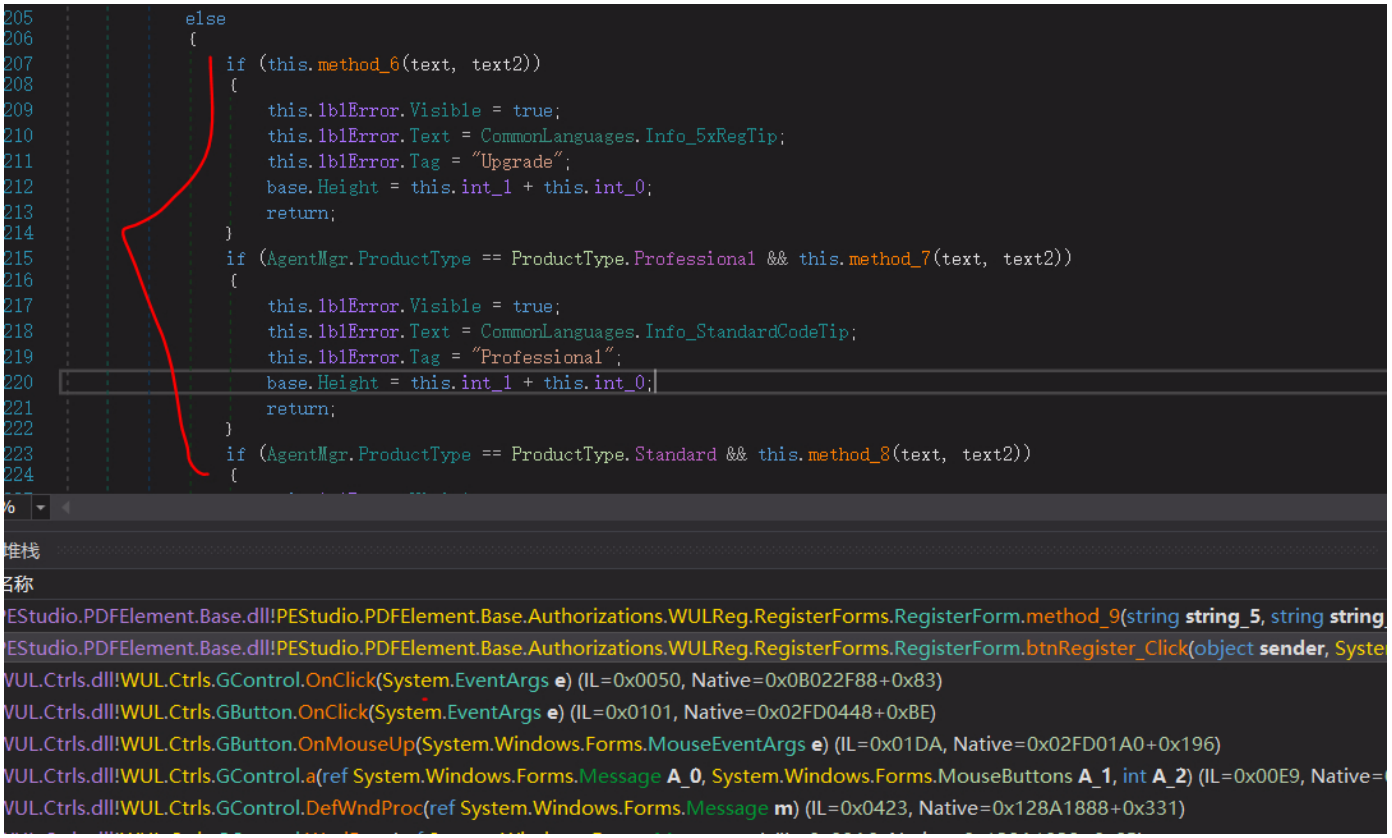
对于上面的错，把出错的位置删掉即可（在这里折腾了好久，才发现直接删掉即可）。

为了方便调试分析，把去掉混淆的 dll 替换掉安装目录里面的。

根据参数和其他的特征定位到我们刚刚找到的关键函数



设下断点，断下来后往上回溯



我们需要的逻辑就是

```
if (AgentMgr.ProductType == ProductType.Professional && this.method_7(text, text2))
                {
                    this.lblError.Visible = true;
                    this.lblError.Text = CommonLanguages.Info_StandardCodeTip;
                    this.lblError.Tag = "Professional";
                    base.Height = this.int_1 + this.int_0;
                    return;
                }
```

那么把其他的去掉，如果是前面那个逻辑正确的话，会提示是 `5.x` 版本的验证码。

那么就把那个删掉

```
else
{
    if (AgentMgr.ProductType == ProductType.Professional && this.method_7(text, text2))
    {
        this.lblError.Visible = true;
        this.lblError.Text = CommonLanguages.Info_StandardCodeTip;
        this.lblError.Tag = "Professional";
        base.Height = this.int_1 + this.int_0;
        return;
    }
    if (AgentMgr.ProductType == ProductType.Standard && this.method_8(text, text2))
    {
        this.lblError.Visible = true;
        this.lblError.Text = CommonLanguages.Info_ProfessionalCodeTip;
        this.lblError.Tag = "Standard";
        base.Height = this.int_1 + this.int_0;
        return;
    }
    this.lblError.Tag = "";
    this.lblError.Visible = true;
    this.lblError.Text = this.string_4;
    base.Height = this.int_1 + this.int_0;
}
```

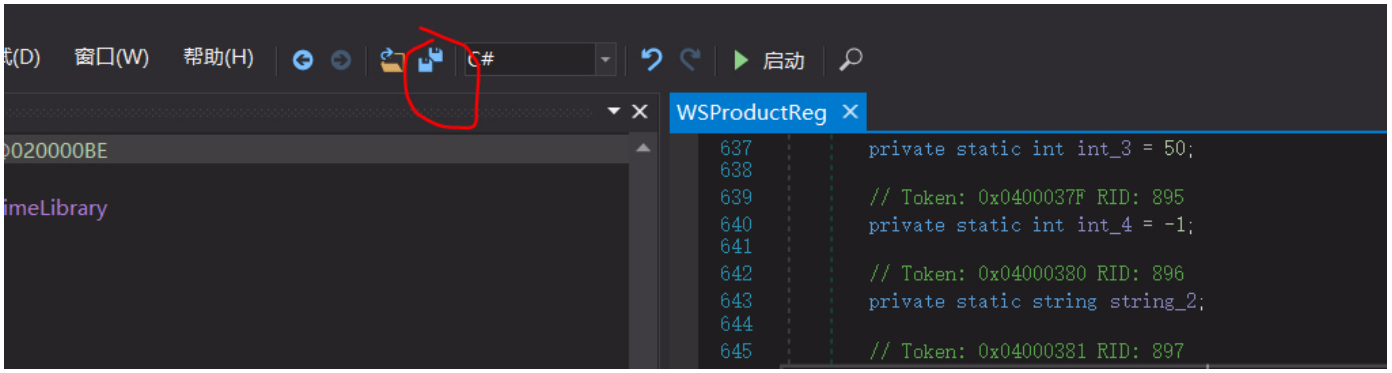
this.method_7 最终调用的也是 WSProductReg.Verify,这个函数最终调用了WSProductReg.Class12.smethod_0

```
// Token: 0x060003FA RID: 1018 RVA: 0x0001EA50 File Offset: 0x0001CC50
public static bool Verify(uint keyn, uint keyd, string regCode, string email)
{
    if (!string.IsNullOrEmpty(email) && email.Length <= WSProductReg.int_3 && email.Length >= WSProductReg.int_4 && !
      string.IsNullOrEmpty(regCode))
    {
        uint tickCount = (uint)Environment.TickCount;
        do
        {
            Thread.Sleep(100);
        }
        while (Environment.TickCount < (int)(tickCount + 1000u));
        return WSProductReg.Class12.smethod_0(keyn, keyd, regCode, email, WSProductReg.string_3, string.Empty);
    }
    return false;
}
```

WSProductReg.Class12.smethod_0 才是真正的 校验函数，于是修改掉他的返回值。

```
// Token: 0x060003FF RID: 1023
internal static bool smethod_0(uint uint_0, uint uint_1, string string_0, string string_1, string string_2, string string_3)
{
    return true;
}
```

为了保持修改，需要先关掉程序，然后使用 dnspy 的保存功能即可。



重启程序，随便输入 序列号 和 邮箱

但是会提示

估计是他有在线校验，断网使用就没问题了。最后使用了 `idea` 破解一样的办法，抓到 验证包， 修改 `host` 是他无法验证。

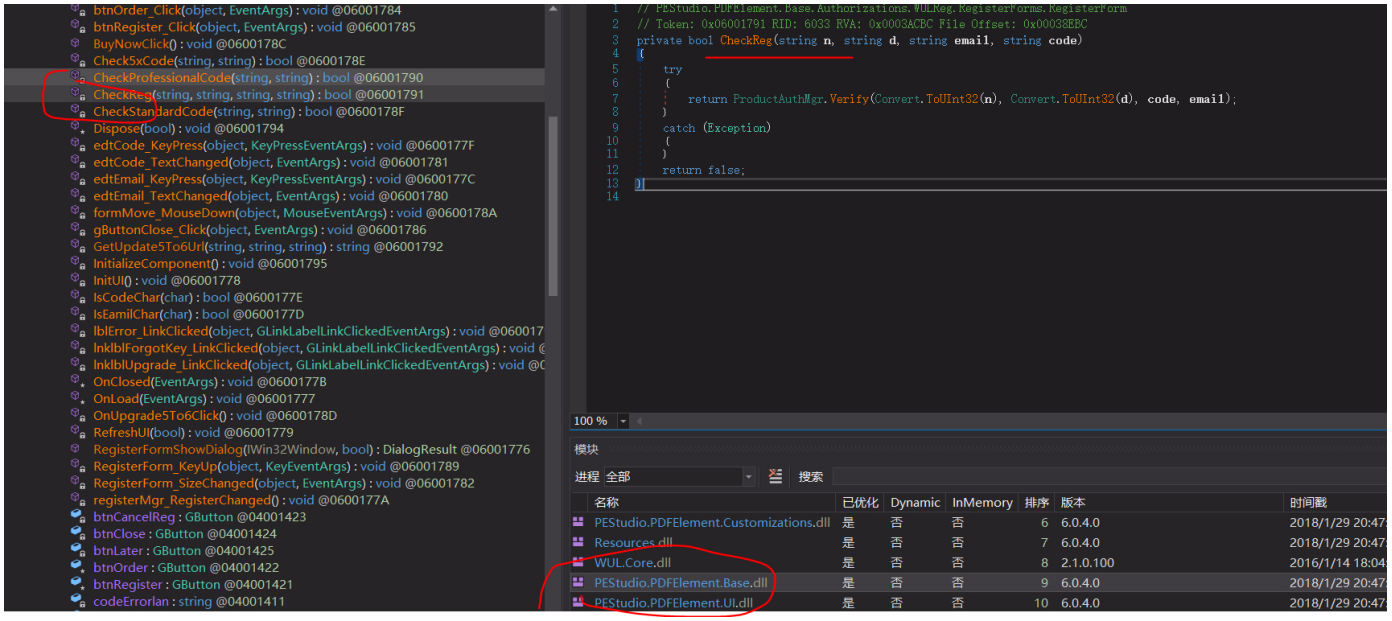
```
GET /interface.php?m=co&client_sign={E7E88F54-EFCA-4C1D-90CE-0103FF51E013}&product_id=3173&version=6.0.3.2154&email=adf
%40qq.com&lang=hk&type=1&cc=497E3141F7904B8721DE98CA32BB81B8&interface_version=1.1&is_coupload=0 HTTP/1.1
Host: platform.wondershare.com
Accept: */*
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/4.0 (compatible; MSIE 6.00)
Cookie: PHPSESSID=lcornkfglobmao7uh1rbdc1tf6
Connection: Keep-Alive
Accept-Encoding: gzip,deflate

HTTP/1.1 200 OK
Server: Tengine/unknown
Date: Fri, 02 Feb 2018 10:22:59 GMT
Content-Type: text/xml; Charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Expires: Fri, 02 Feb 2018 10:22:58 GMT
Cache-Control: no-cache
Pragma: no-cache

<?xml version="1.0" encoding="utf-8"?><CheckResult><Status><![CDATA[quit]]></Status><Message><!
[CDATA[.....................................................................................]]></Message><Cc><!
[CDATA[209A3323D29007379C00AB32A78ECA35]]></Cc><DetailStatus><![CDATA[quit]]></DetailStatus></CheckResult>
```

然后把这个 `0.0.0.0 platform.wondershare.com` 放到 `host` 文件里面。

然后应该就没有问题了。

6.4最新版破解

最新版连混淆都没有，逻辑更加的清晰。



调试找到了 `6.0` 中的相对应的 校验函数 `WSProductReg.RSAKeyCodec.VerifySN`

```
// Token: 0x060000A3 RID: 163 RVA: 0x000030C4 File Offset: 0x000012C4
public static bool Verify(uint keyn, uint keyd, string regCode, string email)
{
    if (string.IsNullOrEmpty(email) || email.Length > WSProductReg.maxChars || email.Length < WSProductReg.minChars |
      string.IsNullOrEmpty(regCode))
    {
        return false;
    }
    uint tickCount = (uint)Environment.TickCount;
    do
    {
        Thread.Sleep(100);
    }
    while (Environment.TickCount < (int)(tickCount + 1000u));
    return WSProductReg.RSAKeyCodec.VerifySN(keyn, keyd, regCode, email, WSProductReg.product, string.Empty);
}
```

修改掉

```
internal static bool VerifySN(uint n, uint d, string sn, string email, string product, string version) { return true; }
```

同时还找到了 在线校验的函数`ProductClient.RegCheck`

```
// Token: 0x060000C0 RID: 192 RVA: 0x00003340 File Offset: 0x00001540
public static bool RegCheck(string Email, string LangID, RegCheckType CheckType)
{
    bool result;
    try
    {
        if (ProductClient.startSuccess)
        {
            ProductClient.SetAntiEmailBreakerFinishCallBack(ProductClient.callBackHandler);
            result = ProductClient.RegCheckInner(Email, LangID, CheckType);
        }
        else
        {
            result = false;
        }
    }
    catch
    {
        result = false;
    }
    return result;
}
```

修改

```
public static bool RegCheck(string Email, string LangID, RegCheckType CheckType)
{
    return true;
}
```

即可。

来源： https://www.cnblogs.com/hac425/p/9416944.html