

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

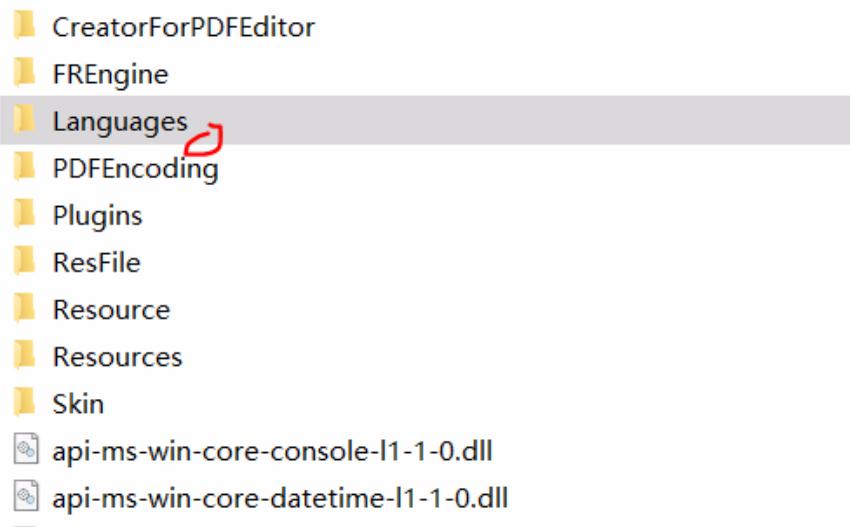
最开始我是拿的 6.0 版本破解的，破解完后发现，这个版本的 bug 特别严重，不能 编辑 pdf（也许是我改坏了， 6.4就没问题），于是后来又下了最新的 6.4. 老版的还有混淆，新版本连混淆都没有。为了总结更多的经验，先写 6.0 版本的破解，然后写 6.4 的破解。

工具

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

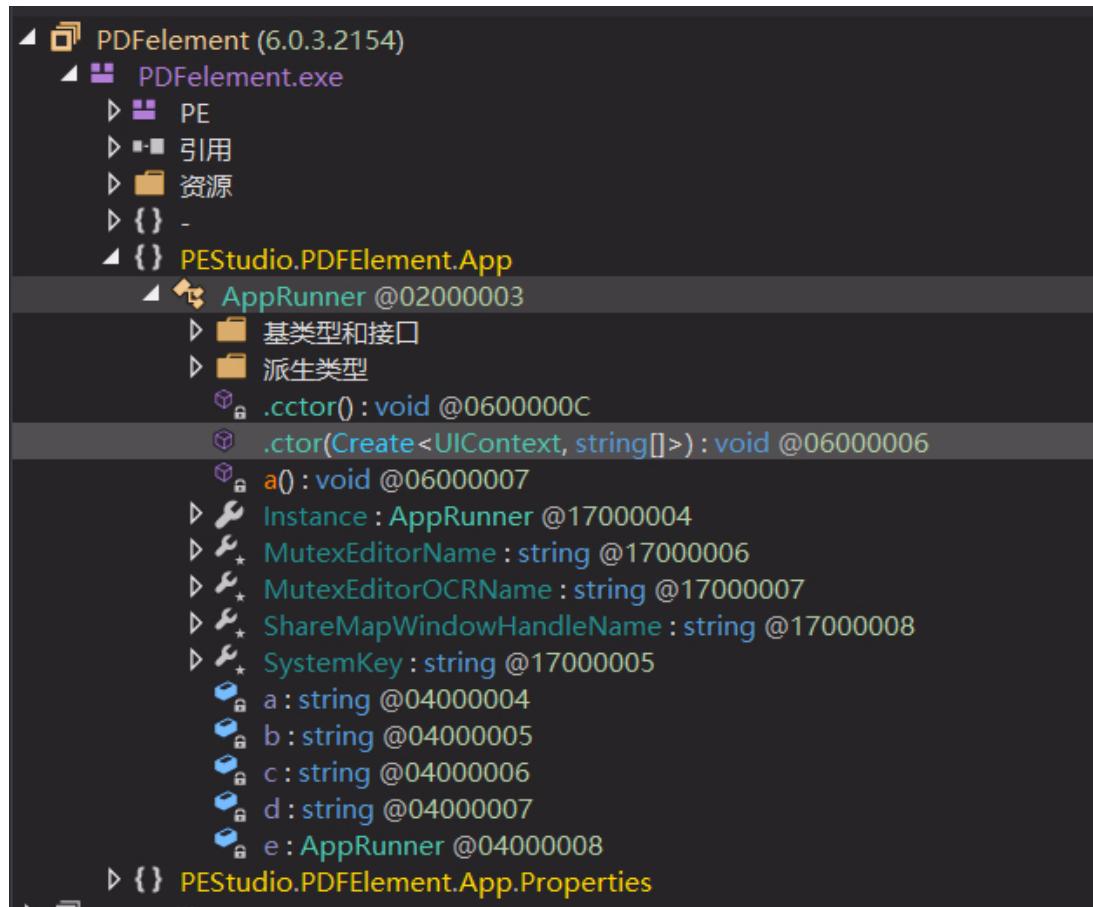
6.0版本破解

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。



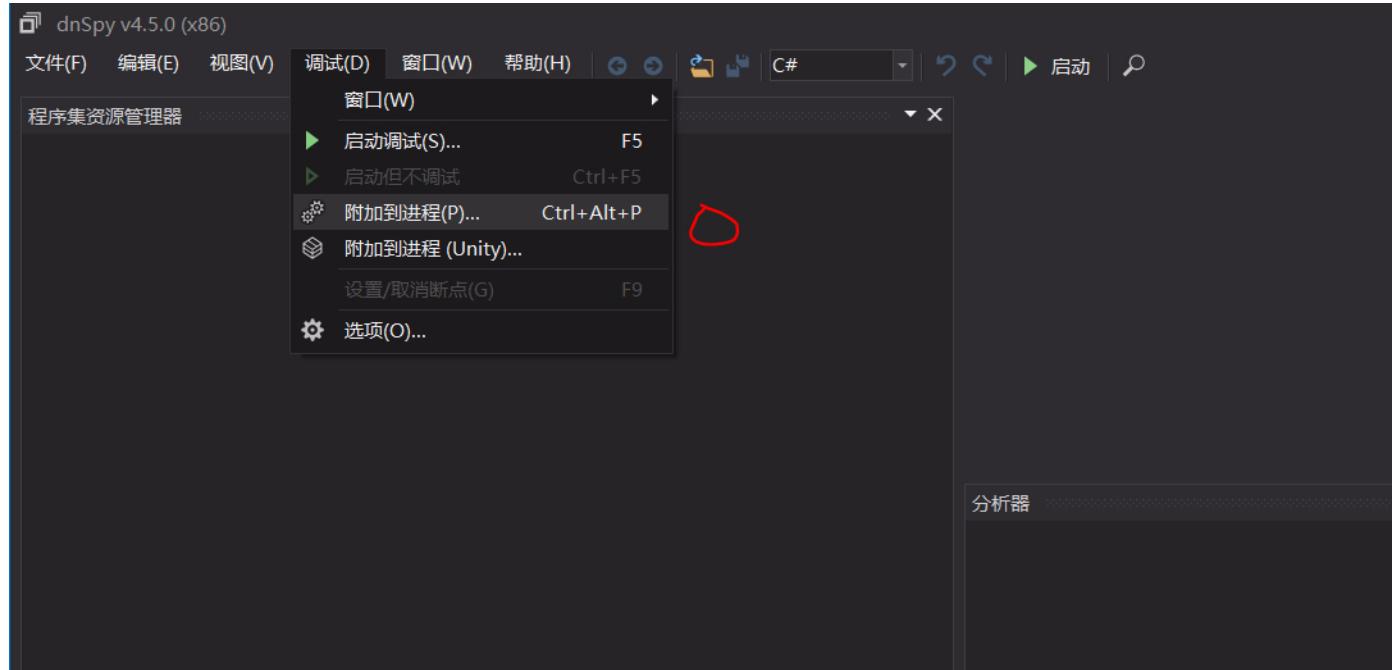
不过该程序的资源文件貌似不是文本格式的，故我放弃了这条路。

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

名称	已优化	Dynamic	InMemory	排序	版本	时间戳
PEStudio.Frameworks.Compacts.dll	是	否	否	3	6.0.3.0	2017/4/19 20:14:
System.dll	是	否	否	4	4.7.2556.0 built by: NET471REL1	2017/9/13 3:32:0
System.Drawing.dll	是	否	否	5	4.7.2556.0 built by: NET471REL1	2017/9/13 3:32:0
PEStudio.PDFElement.Customizations.dll	是	否	否	6	6.0.3.0	2017/4/19 20:14:
Resources.dll	是	否	否	7	6.0.3.0	2017/4/19 20:10:
WUL.Core.dll	是	否	否	8	2.1.0.100	2016/1/14 18:04:
PEStudio.PDFElement.Base.dll	是	否	否	9	6.0.3.0	2017/4/19 20:14:
PEStudio.PDFElement.UI.dll	是	否	否	10	6.0.3.0	2017/4/19 20:14:
PEStudio.PDFElement.Api.dll	是	否	否	11	6.0.3.0	2017/4/19 20:14:
System.Management.dll	是	否	否	12	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:1
BugSplatDotNet2.0.dll	是	否	否	13	6.0.3.0	2017/4/19 20:10:
System.Xml.dll	是	否	否	14	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:5

搜索 局部变量 模块 分析器 调用堆栈 书签

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

```

1 // PESTudio.PDFElement.Base (6.0.3.0)
2 // Token: 0x06000265 RID: 613 RVA: 0x0000855C File Offset: 0x00006750
3 private bool a(string A_0, string A_1, string A_2, string A_3)
4 {
5     bool result;
6     try
7     {
8         result = WSProductReg.Verify(Convert.ToInt32(A_0), Convert.ToInt32(A_1), A_3, A_2);
9     }
10    catch (Exception)
11    {
12    }
13    if (true)
14    {
15        return result;
16    }
17 }
18
19 return result;
20
21

```

名称	已优化	Dynamic	InMemory	排序	版本	时间戳	地址
PEStudio.Frameworks.Compacts.dll	是	否	否	3	6.0.3.0	2017/4/19 20:14:34	07A60000-0
System.dll	是	否	否	4	4.7.2556.0 built by: NET471REL1	2017/9/13 3:32:08	0C160000-0
System.Drawing.dll	是	否	否	5	4.7.2556.0 built by: NET471REL1	2017/9/13 3:32:0	08E70000-0
PEStudio.PDFElement.Customizations.dll	是	否	否	6	6.0.3.0	2017/4/19 20:10:46	08870000-0
Resources.dll	是	否	否	7	6.0.3.0	2017/4/19 20:10:46	08870000-0
WUL.Core.dll	是	否	否	8	2.1.0.100	2016/1/14 18:04:06	08530000-0
PEStudio.PDFElement.Base.dll	是	否	否	9	6.0.3.0	2017/4/19 20:14:44	09F60000-0
PEStudio.PDFElement.UI.dll	是	否	否	10	6.0.3.0	2017/4/19 20:14:37	092B0000-0
PEStudio.PDFElement.Api.dll	是	否	否	11	6.0.3.0	2017/9/13 3:31:18	094E0000-0
System.Management.dll	是	否	否	12	4.7.2556.0 built by: NET471REL1	2017/4/19 20:10:57	08860000-0
BugSplatDotNet2.0.dll	是	否	否	13	6.0.3.0	2017/4/19 20:10:57	08860000-0
System.Xml.dll	是	否	否	14	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:59	09F80000-0

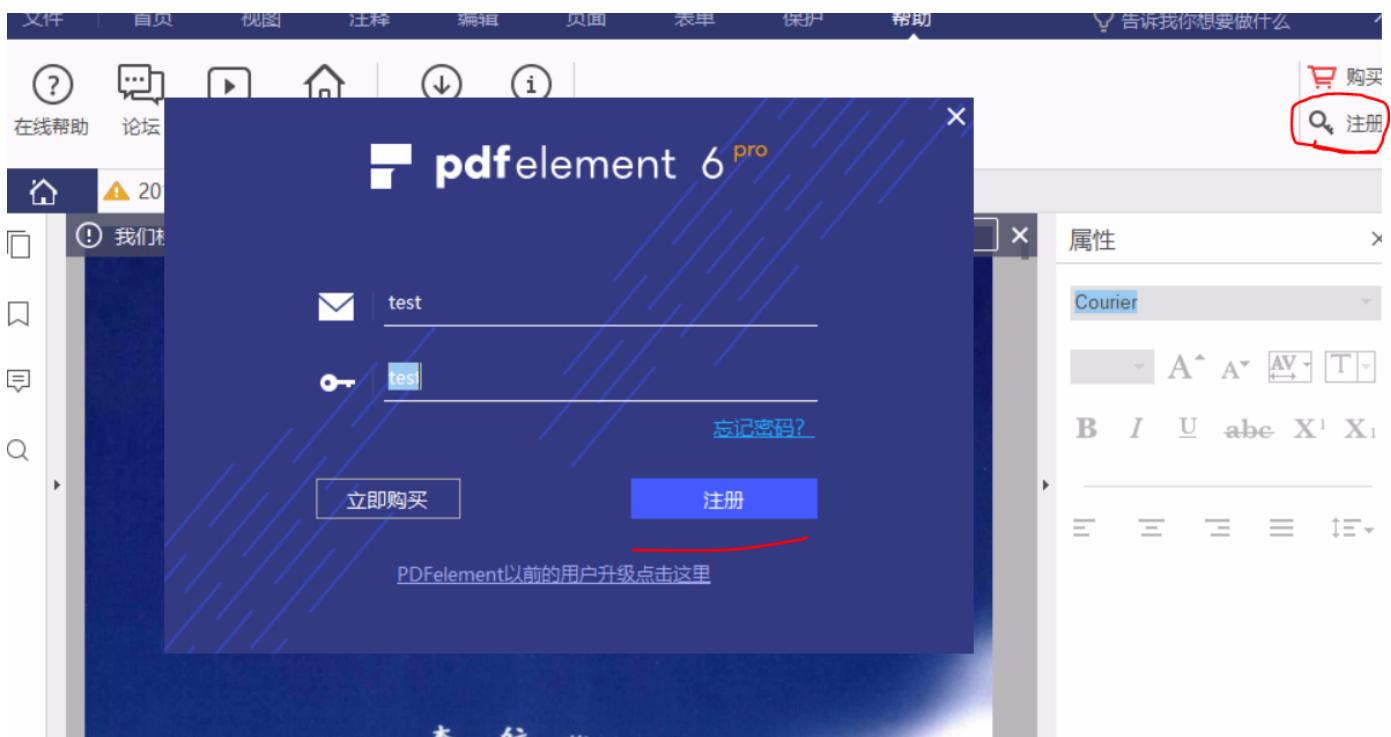
看这些函数名，估计这里是在注册时，用来校验的函数了。在这里下个断点

```

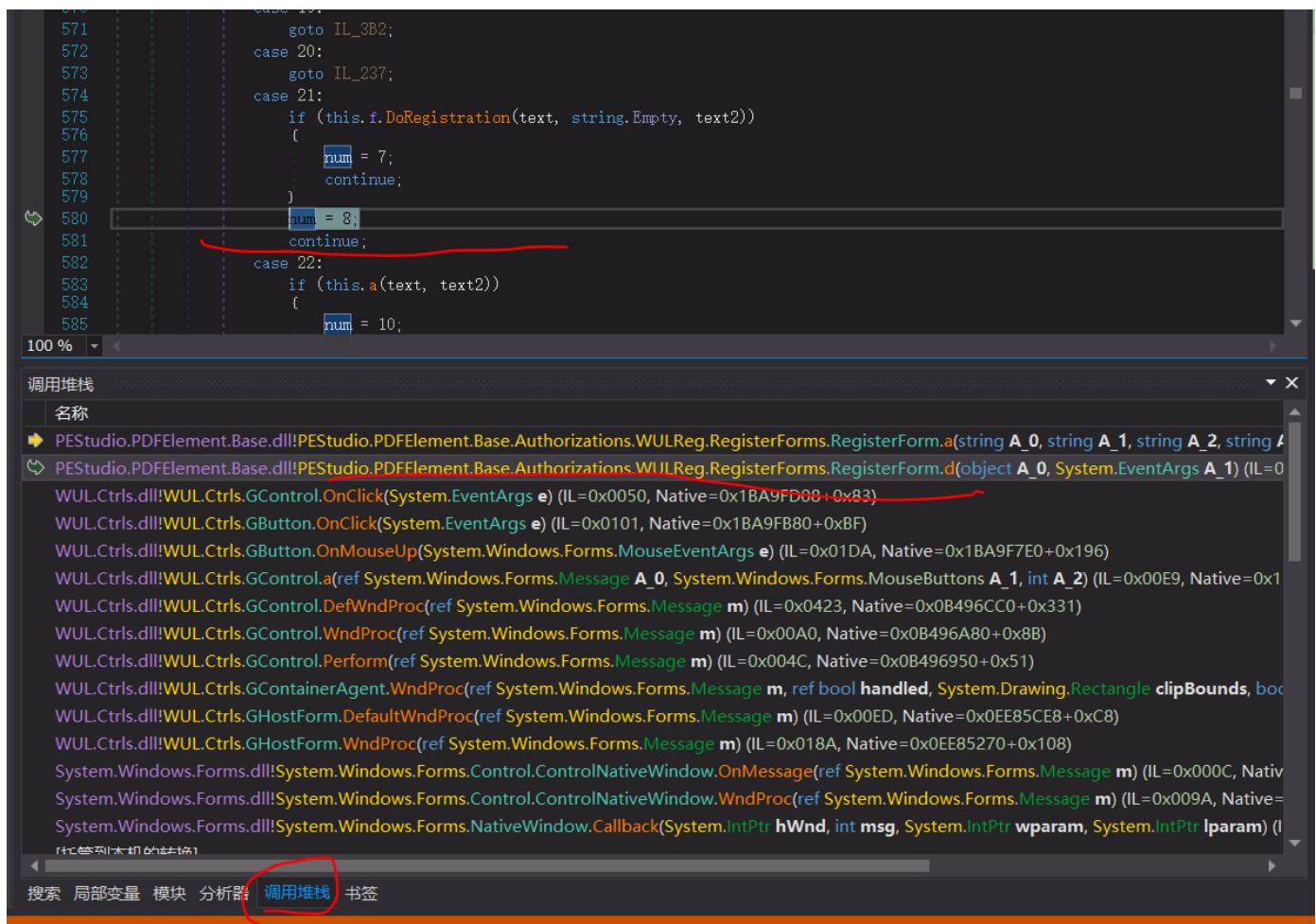
2 // Token: 0x06000265 RID: 613 RVA: 0x0000855C File Offset: 0x00006750
3 private bool a(string A_0, string A_1, string A_2, string A_3)
4 {
5     bool result;
6     try
7     {
8         result += WSProductReg.Verify(Convert.ToInt32(A_0), Convert.ToInt32(A_1), A_3, A_2);
9     }
10    catch (Exception)
11    {
12        // 在新标签页中打开(T) Ctrl+单击
13        // 启动调试 F5
14        // 添加断点(A) F9
15        // 编辑方法 (C#)... Ctrl+Shift+E
16        // 编辑类 (C#)... 
17        // 添加类 (C#)... 
18        // 与程序集合并...
19        // 编辑IL指令(S)
20    }
21 }

```

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。

```

3  using System.Runtime.CompilerServices;
4  using System.Windows.Forms;
5  using PEStudio.PDFElement.Base.UI;
6  using PEStudio.PDFElement.Base.UI.Controls;
7  using WUL.Ctrls;
8  using WUL.Reg;
9
10 namespace PEStudio.PDFElement.Base.Authorizations.WULReg.RegisterForms
11 {
12     // Token: 0x0200008D RID: 141
13     public partial class RegisterForm : BaseDialogForm
14     {
15         // Token: 0x06000265 RID: 613 RVA: 0x0000855C File Offset: 0x0000675C
16         public bool a(string A_0, string A_1, string A_2, string A_3)
17         {
18             bool result;
19             try

```

代码 描述 文件 行

- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.cs 16
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 28
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 54
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 60
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 10
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 11
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 11
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 14
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 15
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 15
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 16
- CS0102 The type 'RegisterForm' already contains a definition for 'a' main.g.cs 17
- CS0102 The type 'RegisterForm' already contains a definition for 'd' main.g.cs 18
- CS0102 The type 'RegisterForm' already contains a definition for 'c' main.g.cs 18
- CS0102 The type 'RegisterForm' already contains a definition for 'b' main.g.cs 18
- CS0102 The type 'RegisterForm' already contains a definition for 'h' main.g.cs 19
- CS0102 The type 'RegisterForm' already contains a definition for 'g' main.g.cs 19
- CS0102 The type 'RegisterForm' already contains a definition for 'f' main.g.cs 19

main.cs 编译 取消

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

方法集资源管理器

```

.method _9(string, string, string, string)...
    .ctor() : void @06000244
    .bgnOrder_Click(object, EventArgs) : void @06000253
    .btnRegister_Click(object, EventArgs) : void @06000254
    .BuyNowClick() : void @06000258
    .Dispose(bool) : void @06000263
    .edtCode.KeyPress(object, KeyPressEventArgs) : void @0600024E
    .edtCode.TextChanged(object, EventArgs) : void @06000250
    .edtEmail.KeyPress(object, KeyPressEventArgs) : void @0600024B
    .edtEmail.TextChanged(object, EventArgs) : void @0600024F
    .gbbutton_0_Click(object, EventArgs) : void @06000257
    .gbbutton_2_Click(object, EventArgs) : void @06000256
    .gbbutton_3_Click(object, EventArgs) : void @06000255
    .gbpanel_4_MouseDown(object, MouseEventArgs) : void @06000259
    .lblError_LinkClicked(object, LinkLabelLinkClickedEventArgs) : void @060002
    .lnklblForgotKey_LinkClicked(object, LinkLabelLinkClickedEventArgs) : void @0600024A
    .lnklblUpgrade_LinkClicked(object, LinkLabelLinkClickedEventArgs) : void @0600024B
    .method_0() : void @06000247
    .method_1(bool) : void @06000248
    .method_10() : void @06000264
    .method_2() : void @06000249
    .method_3(char) : bool @0600024C
    .method_4(char) : bool @0600024D
    .method_5() : void @0600025C
    .method_6(string) : bool @0600025D
    .method_7(string, string) : bool @0600025E
    .method_8(string, string, string) : bool @0600025F
    .method_9(string, string, string, string) : bool @06000260
    .OnLoadEventArgs() : void @0600024A
    .OnLoadEventArgs : void @06000246
    .RegisterFormShowDialog(IWin32Window, bool) : DialogResult @06000245
    .RegisterForm.KeyUp(object, KeyEventArgs) : void @06000258
    .RegisterForm.SizeChanged(object, EventArgs) : void @06000251
    .smethod_0(string, string, string) : string @06000261
    .btnOrder_GButton @04000243
    .btnRegister_GButton @04000242
    .edtCode_GEdit @0400023B
    .edtEmail_GEdit @0400023C
    .gbbutton_0_GButton @04000244

```

模块

名称	已优化	Dynamic	InMemory	排序	版本	时间戳	地址
PEStudio.PDFElement.Customizations.dll	是	否	否	6	6.0.3.0	2017/4/19 20:14:55	07800000-0000-0000-0000-000000000000
Resources.dll	是	否	否	7	6.0.3.0	2017/4/19 20:10:46	08580000-0000-0000-0000-000000000000
WUL.Core.dll	是	否	否	8	2.1.0.100	2016/1/14 18:04:06	08270000-0000-0000-0000-000000000000
PEStudio.PDFElement.Base.dll	是	否	否	9	6.0.3.0	2017/4/19 20:14:35	08E20000-0000-0000-0000-000000000000
PEStudio.PDFElement.Ui.dll	是	否	否	10	6.0.3.0	2017/4/19 20:14:44	093A0000-0000-0000-0000-000000000000
PEStudio.PDFElement.Api.dll	是	否	否	11	6.0.3.0	2017/4/19 20:14:37	08F60000-0000-0000-0000-000000000000
System.Management.dll	是	否	否	12	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:18	09200000-0000-0000-0000-000000000000
BugSplatDotNet2.0.dll	是	否	否	13	6.0.3.0	2017/4/19 20:10:57	08580000-0000-0000-0000-000000000000
System.Xml.dll	是	否	否	14	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:59	60E80000-6000-0000-0000-000000000000
System.Configuration.dll	是	否	否	15	4.7.2556.0 built by: NET471REL1	2017/9/13 3:31:52	6E060000-6E00-0000-0000-000000000000
System.Core.dll	是	否	否	16	4.7.2600.0 built by: NET471REL1LAST	2017/10/6 21:57:15	6A5C0000-6
Accessibility.dll	否	否	否	17	4.7.2556.0 built by: NET471REL1	2017/9/13 3:20:55	09390000-0000-0000-0000-000000000000

设下断点，断下来后往上回溯

```
205
206
207     else
208     {
209         if (this.method_6(text, text2))
210         {
211             this.lblError.Visible = true;
212             this.lblError.Text = CommonLanguages.Info_5xRegTip;
213             this.lblError.Tag = "Upgrade";
214             base.Height = this.int_1 + this.int_0;
215             return;
216         }
217         if (AgentMgr.ProductType == ProductType.Professional && this.method_7(text, text2))
218         {
219             this.lblError.Visible = true;
220             this.lblError.Text = CommonLanguages.Info_StandardCodeTip;
221             this.lblError.Tag = "Professional";
222             base.Height = this.int_1 + this.int_0;
223             return;
224         }
225         if (AgentMgr.ProductType == ProductType.Standard && this.method_8(text, text2))
226         {
227             ...
228         }
229     }
230 }
```

堆栈

简称

```
EStudio.PDFElement.Base.dll!PEStudio.PDFElement.Base.Authorizations.WULReg.RegisterForms.RegisterForm.method_9(string string_5, string string_6)
EStudio.PDFElement.Base.dll!PEStudio.PDFElement.Base.Authorizations.WULReg.RegisterForms.RegisterForm.btnRegister_Click(object sender, System.EventArgs e)
VUL.Ctrls.dll!WUL.Ctrls.GControl.OnClick(System.EventArgs e) (IL=0x0050, Native=0x0B022F88+0x83)
VUL.Ctrls.dll!WUL.Ctrls.GButton.OnClick(System.EventArgs e) (IL=0x0101, Native=0x02FD0448+0xBE)
VUL.Ctrls.dll!WUL.Ctrls.GButton.OnMouseUp(System.Windows.Forms.MouseEventArgs e) (IL=0x01DA, Native=0x02FD01A0+0x196)
VUL.Ctrls.dll!WUL.Ctrls.GControl.a(ref System.Windows.Forms.Message A_0, System.Windows.Forms.MouseButtons A_1, int A_2) (IL=0x00E9, Native=0x02FD01A0+0x196)
VUL.Ctrls.dll!WUL.Ctrls.GControl.DefWndProc(ref System.Windows.Forms.Message m) (IL=0x0423, Native=0x128A1888+0x331)
```

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;
}

```

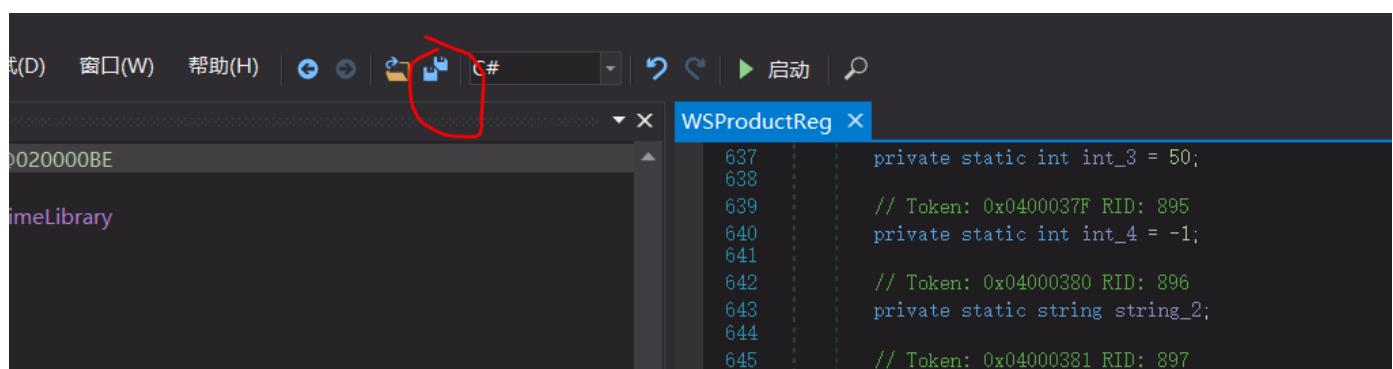
WSProductReg.Class12.smethod_0 才是真正的校验函数，于是修改掉他的返回值。

```

// 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_coupleload=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>

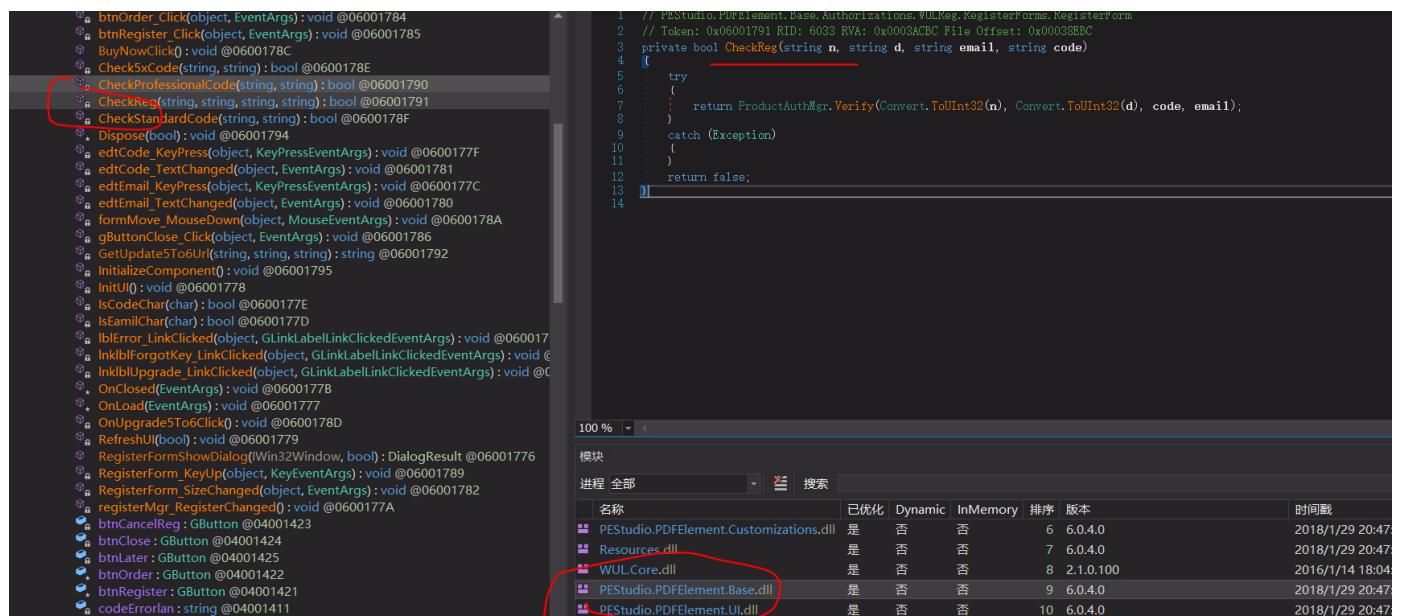
```

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 + 1000));
    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>