



MaQaQ

博客园 首页 新随笔 联系 管理 订阅 写作

随笔- 34 文章- 0 评论- 26 阅读- 65149

昵称: MaQaQ
园龄: 4年
粉丝: 2
关注: 6
+加关注

探究——C# .net 代码混淆/加壳

背景:

保密。

过程:

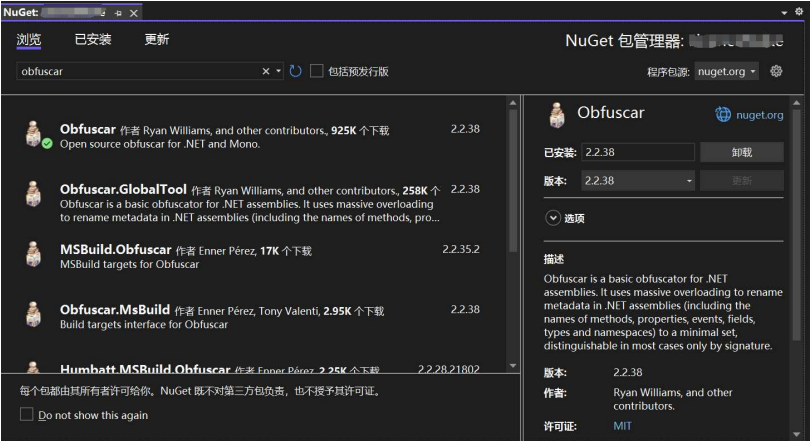
先查询一下常见的加壳工具:

- Dotfuscator, 官方自带, 据说免费版混淆程度不高
- ConfuserEx, 只支持.NET Framework 2.0/3.0/3.5/4.0/4.5/4.6/4.7/4.8, 不支持.NET Core
- Virbox Protector, 很好很优秀, 但是收费
- NET Reactor, 最新6.9版收费, PJ版到4.9不支持.NET Core
- Obfuscator, 开源, 可以用dotnet tool或项目构建的方式进行使用

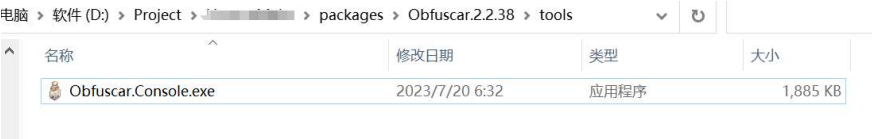
那就先用Obfuscator试试水 (官方文档: <https://docs.obfuscator.com/getting-started/configuration>)。

方式一: nuget安装 (推荐, 这种方式可以针对性下载各.net版本对应工具包)

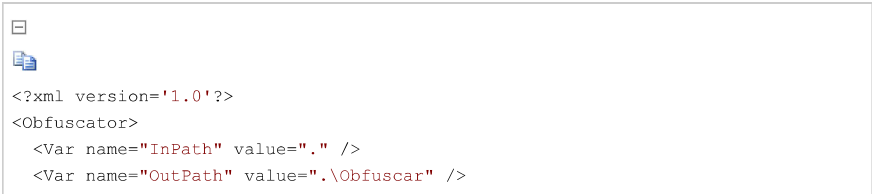
1、在项目中使用nuget安装obfuscator



2、在项目根目录下找到packages\Obfuscator.2.2.38\tools, 将Obfuscator.Console.exe拷到要加密文件的文件夹中



3、新建一个Obfuscator.xml 文件放到相同目录下, 内容如下:



2025年7月						
日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

备忘(13)
工具(10)
C#(7)
windows(4)
git(4)
WPF(3)
探究(3)
数据库(3)
错误记录(3)
通信(2)
更多

随笔分类

技术(3)

随笔档案

2025年3月(3)
2024年6月(1)
2024年5月(1)
2024年4月(1)
2023年11月(3)
2023年10月(3)
2023年9月(2)
2023年8月(2)
2023年5月(1)
2022年11月(4)
2022年9月(1)

```
<Var name="KeepPublicApi" value="true" />
<Var name="HidePrivateApi" value="true" />
<Var name="HideStrings" value="true" />
<Var name="UseUnicodeNames" value="true" />
<Var name="ReuseNames" value="true" />
<Var name="RenameFields" value="true" />
<Var name="RegenerateDebugInfo" value="true" />

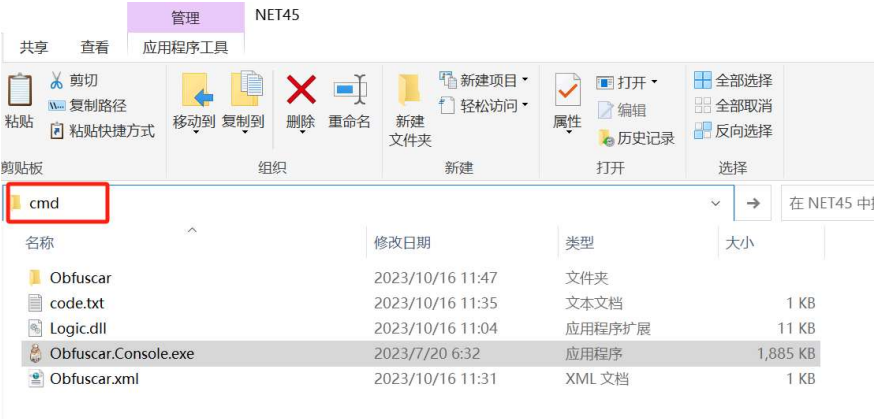
<Module file="$ (InPath) \Logic.dll" />

</Obfuscator>
```

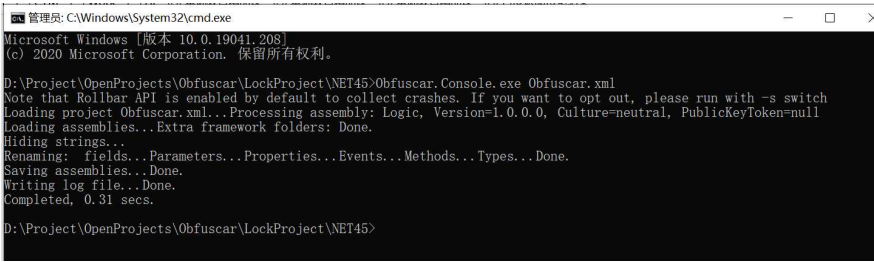
下图中，Logic.dll是要加密的类库：



4、用命令提示符cmd进入到目录下（可以在cmd里用cd指令跳转，也可以直接打开目标文件夹，然后在上方的文件路径那里直接替换成cmd后enter）



5、执行>Obfuscator.Console.exe Obfuscator.xml



6、在生成的Obfuscator文件夹中可以找到被加壳后的同名Logic.dll



上述得到的Logic.dll即可被其他项目直接引用，加密后类似：

- 2022年8月(1)
- 2022年7月(1)
- 2022年6月(1)
- 2022年3月(4)
- 2021年8月(2)
- 2021年7月(3)

阅读排行榜

- 1. 使用NSSM将.exe程序安装成windows服务(8022)
- 2. C# 数字转字符串(5924)
- 3. 探究——C# .net 代码混淆/加壳(5835)
- 4. Android Studio Giraffe安装与gradle配置(5508)
- 5. C# DataGridView绑定数据源后不显示问题(4351)

评论排行榜

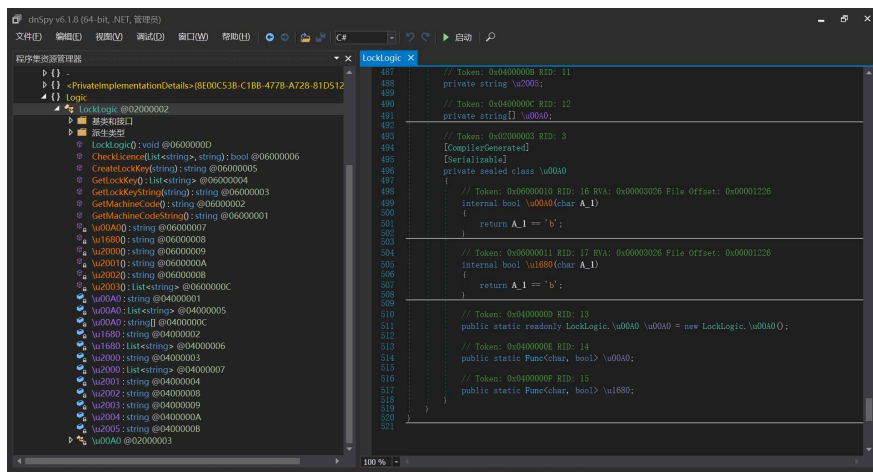
- 1. C# 从零开始使用Layui.Wpf库开发WPF客户端(8)
- 2. 探究——C# .net 代码混淆/加壳(7)
- 3. C#通过FTP获取服务端文件(3)
- 4. 使用NSSM将.exe程序安装成windows服务(2)
- 5. C# 串口通信(2)

推荐排行榜

- 1. 探究——C# .net 代码混淆/加壳(12)
- 2. C# 从零开始使用Layui.Wpf库开发WPF客户端(8)
- 3. 使用NSSM将.exe程序安装成windows服务(7)
- 4. win10离线安装.net3.5失败的解决方案(4)
- 5. windows下测试TCP/UDP端口连通性(3)

最新评论

- 1. Re:C# List<T>复制
复制List<T>，莫名其妙的数据都乱了，这篇文章帮我了一个大忙，也彻底理解了值类型和引用类型，多谢博主。
--包子789654
- 2. Re:C#通过FTP获取服务端文件
直接在iis里开ftp不是更方便吗？我直接在web服务业里开一段文件下载，这样连配置都不用了。
--flysun027
- 3. Re:C# 从零开始使用Layui.Wpf库开发WPF客户端
@花落心语 本来就是工具的问题，最后一个个重新自绘了。就比如一个表格单元格里有N个元素的搭积木方式，就是有性能瓶颈...
--eveloper
- 4. Re:C# 从零开始使用Layui.Wpf库开发WPF客户端
@eveloper @花落心语 再深入了解，数据驱动的模式，调试和优化都很麻烦 有的麻烦，不是因为工具的问题，是人的问题。当绝大多数人都觉得某个工具好用，但自己却觉得不好用的时候，就应该反思下自身的问...
--花落心语
- 5. Re:C# 从零开始使用Layui.Wpf库开发WPF客户端
@花落心语 再深入了解，数据驱动的模式，调试和优化都很麻烦...
--eveloper



方式二：dotnet tool (.net 6)

- 1、新建一个Obfuscator.xml 文件放到类库所在目录，内容如下：



其中，Module对应填入想要加壳的类库，可以添加多行，AssemblySearchPath根据自己.net的路径进行配置。

- 2、在cmd中进入到上述目录中，执行命令：

```
dotnet tool install --global Obfuscator.GlobalTool
```

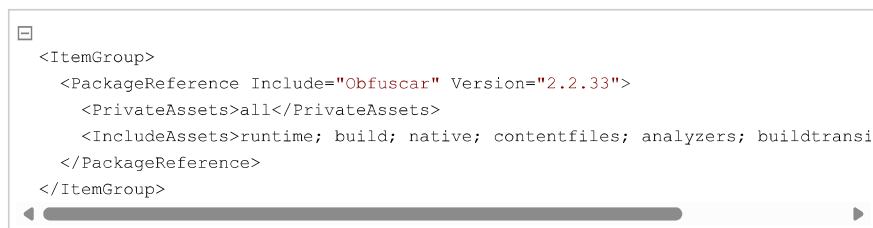
- 3、在cmd中执行命令： obfuscator.console Obfuscator.xml

- 4、在上述目录中找到自动生成的Obfuscator文件夹，加壳后的类库就存放在里面，拷贝出来即可使用。

PS：.net6的带WebAPI的exe好像加壳失败，待测试。

方式三：项目构建

- 1、在csproj 项目文件中添加安装Obfuscator的代码：



- 2、在项目中添加一个Obfuscator.xml 文件，内容跟方式一的类似，再在csproj 项目文件中设置更新



3、在csproj 项目文件中设置自动构建

 View Code

PS：这种方式还没测试过，待测试

总结：

- 1、使用手册：<https://docs.obfuscar.com/getting-started/configuration>
- 2、可以用skip自定义跳过混淆的内容
- 3、如果混淆后，部分涉及到正则表达式的内容无法运行，那么考虑将internal的属性改为public

本文来自博客园，作者：[MaQaQ](#)，转载请注明原文链接：<https://www.cnblogs.com/magicMaQaQ/p/17702951.html>
本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文链接，否则保留追究法律责任的权利。

标签：[工具](#)，[探究](#)

好文要顶

关注我

收藏该文

微信分享



[MaQaQ](#)
粉丝 - 2 关注 - 6

[+加关注](#)

120

[升级成为会员](#)

« 上一篇：[win10的开机启动文件夹](#)
» 下一篇：[错误记录-FileStream访问被拒绝](#)

posted @ 2023-10-17 11:50 [MaQaQ](#) 阅读(5836) 评论(7) [收藏](#) [举报](#)

评论		默认 按时间 按支持数 ↕
# 1楼	2023-10-18 09:15 iamsunrise	回复 引用
在容器中怎么做事情呢，都持续集成了，没有windows环境了		支持(0) 反对(0)
# 2楼	2023-10-19 11:19 收破烂 	回复 引用
这个混淆的工具都是开源的，混淆算法都开源的。是不是也就是说 不安全呢？可能被人反混淆回源代码		支持(0) 反对(0)
# 3楼	[楼主] 2023-10-19 15:39 MaQaQ	回复 引用
@收破烂 我个人感觉吧，他是把源代码里面有意义的命名，给随机替换成类似短字母这种看不出含义的命名，混淆后的代码反编译出来后的可读性很低。		支持(0) 反对(0)
# 4楼	2024-06-19 14:50 Patrick-Rex	回复 引用
@收破烂 没有不能破解的程序，只是增加破解的成本，如果对方觉得不值得就不会破解了		支持(0) 反对(0)
# 5楼	2024-07-10 14:03 keke7106	回复 引用
不是有AOT模式吗，如果想保密就改成AOT。		支持(0) 反对(0)
# 6楼	[楼主] 2024-07-19 15:49 MaQaQ	回复 引用
@keke7106 AOT好像是.net7开始有的？我用的.net6好像没看见有，感觉是挺好用的。		支持(0) 反对(0)





@keke7106
net7发布成单个exe文件，AOT模式也没法用

支持(0) 反对(0)


刷新评论 刷新页面 返回顶部

发表评论 升级成为园子VIP会员

编辑 预览

B    

支持 Markdown

 自动补全

提交评论 退出 订阅评论 我的博客

[Ctrl+Enter快捷键提交]

- 【推荐】100%开源！大型工业跨平台软件C++源码提供，建模，组态！
- 【推荐】2025 HarmonyOS 鸿蒙创新赛正式启动，百万大奖等你挑战
- 【推荐】博客园的心动：当一群程序员决定开源共建一个真诚相亲平台
- 【推荐】开源 Linux 服务器运维管理面板 1Panel V2 版本正式发布
- 【推荐】轻量又高性能的 SSH 工具 IShell：AI 加持，快人一步



相关博文：

- 使用IL_DASM反编译exe
- C# 从零开始使用Layui.Wpf库开发WPF客户端
- 探究——C# .net 代码混淆/加壳
- 探究——C# .net 代码混淆/加壳
- C# 代码混淆

阅读排行：

- 推荐 5 款实用的 Docker 可视化管理工具，工作效率翻倍！
- 瞧瞧别人家的接口重试，那叫一个优雅！
- 独立开发第二周：构建、执行、规划
- 【EF Core】框架是如何识别实体类的属性和主键的
- 简单记录下最近2个月完成的线上系统迁移工作