



会员

周边

众包

新闻

博文

闪存

赞助商

Chat2DB

代码改变世界



.NET100

.NET开发学习交流

博客园

首页

新随笔

联系

订阅

管理

用这些C#代码混淆器保护你的代码安全

最近有群友问，怎么保护发布的C#代码不被别人反编译，这就需要C#代码混淆组件。C#是一种强大的编程语言，可以用于开发各种类型的应用程序，包括桌面应用程序、Web应用程序和移动应用程序等。但是，由于C#程序易于反编译，使得代码的安全性和保护性受到了威胁。为了保护C#代码，开发人员可以使用代码混淆器对代码进行混淆和保护。本文将介绍几种常用的C#代码混淆器。

ConfuserEx

ConfuserEx是一款免费、开源的C#代码混淆器，它支持多种混淆技术，如控制流平坦化、字符串加密、模拟器代码等。此外，ConfuserEx还提供了插件系统，可以通过插件实现更多的混淆技术。

GitHub地址是：github.com/yck1509/ConfuserEx。

Obfuscator

Obfuscator是一种免费、开源的C#代码混淆器，它可以对整个程序集进行混淆，支持多种混淆技术，如重命名、控制流混淆、字符串混淆等。Obfuscator还支持命令行为和MSBuild集成，可以轻松地与构建过程集成。

GitHub地址是：github.com/obfuscator/obfuscator。

Dotfuscator Community Edition

Dotfuscator Community Edition是一款免费的代码混淆器，由PreEmptive Solutions开发。它提供了多种混淆技术，如命名混淆、控制流混淆、字符串混淆等。Dotfuscator Community Edition也可以与Visual Studio集成，并且提供了丰富的文档和教程。

文档地址是：

preemptive.com/dotfuscator/4.43/doc/en/index.html

Eazfuscator.NET Community Edition

Eazfuscator.NET Community Edition是一款免费的C#代码混淆器，它支持多种混淆技术，如命名混淆、控制流混淆、字符串混淆等。此外，Eazfuscator.NET Community Edition还提供了加壳功能和资源加密功能，可以更好地保护代码的安全性。Eazfuscator.NET的

文档地址是：

gazelle.gitbooks.io/eazfuscator-net/content/。

Babel Obfuscator

Babel Obfuscator是一款免费的、易于使用的C#代码混淆器，它提供了多种混淆技术，如重命名、控制流混淆、字符串混淆等。Babel Obfuscator还提供了对ASP.NET应用程序和WPF应用程序的支持，并提供了Visual Studio插件，可以轻松地与构建过程集成。

文档地址是：babelfor.net/docs/babel-obfuscator。

结语

什么组件都是有利弊的，代码混淆器也一样，使用代码混淆器不能完全保护程序不被破解，只能增加反编译的难度。同时，使用代码混淆器可能会对程序的性能、可读性和调试能力造成影响，需要进行权衡利弊来决定是否使用。以上的c#代码混组件大家根据组件的文档并结合自己项目情况使用，切勿盲目选择。希望本文对你有所收获，欢迎留言吐槽。

公告

昵称：翔星
园龄：9年1个月
粉丝：56
关注：0
[-取消关注](#)

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

C#知识(20)
C#(19)
.NET(19)
.NET Core(7)
Entity Framework(3)
.net求职(3)
ChatGPT(2)
人工智能(2)
.NET7(2)
winui(1)
[更多](#)

合集

[开源软件\(1\)](#)
[C#知识\(1\)](#)

随笔档案

[2024年5月\(2\)](#)
[2024年4月\(2\)](#)
[2024年1月\(2\)](#)

来源公众号：DotNet开发跳槽

标签: C#知识

好文要顶

已关注

收藏该文

微信分享



翔星

粉丝 - 56 关注 - 0

10

升级成为会员

我在关注他 取消关注

« 上一篇: C#异步方法async/await的三种返回类型

» 下一篇: 使用这些.NET代码生成器, 开发效率飙升

posted @ 2023-05-18 12:43 翔星 阅读(3774) 评论(1) 收藏 举报

评论列表

默认|按时间|按支持数

#1楼 2023-05-25 09:35 lanwah

干得漂亮，整理的挺好👍

支持(0) 反对(0)

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：

- 每个.NET开发都应掌握的C#异常处理知识点
- C#代码重构的几个典型案例
- 保护C#代码的艺术：深入浅出代码混淆技术

- 2023年12月(4)
- 2023年11月(3)
- 2023年10月(2)
- 2023年9月(4)
- 2023年8月(7)
- 2023年7月(8)
- 2023年6月(7)
- 2023年5月(12)
- 2023年4月(12)
- 2023年3月(11)
- 2023年2月(4)
- 2020年11月(1)
- 更多

文章分类

- .NET(1)
- 数据库(1)
- 无题(1)

阅读排行榜

1. lambda表达式查询经验：IN 和groupby的使用(15168)
2. C#遍历指定文件夹中所有文件的3种方法 (8812)
3. Visual Studio 2017 RC 初探安装(8336)
4. C#List的3种排序方法(6764)
5. 最全.NET Core 、.NET 5、.NET 6和.NET 7简介和区别(6032)

评论排行榜

1. .NET使用EF批量插入数据，一行代码性能飙升！(7)
2. 每个.NET开发都应该知道的10个.NET库(6)
3. C#中的var究竟是强类型还是弱类型？ (4)
4. 避坑：.NET内存泄露的几种情况(4)
5. 为什么C#要采用顶级语句？ (3)

推荐排行榜

1. 每个.NET开发都应掌握的C#网络编程知识点(5)
2. 三种方法让.NET轻松实现Excel转PDF(5)
3. 每个.NET开发都应该知道的10个.NET库(5)
4. 在.NET Core使用 HttpClient 的正确方式(5)
5. 用上这几种.NET EF Core性能调优，查询性能飙升(5)

最新评论

1. Re:简单到复杂：C#拷贝文件的3种方法
@lanwah 是的，应该写成 new FileInfo(sourceFile).CopyTo(targetFile, true);...
--东经115
2. Re:三种方法让.NET轻松实现Excel转PDF
也使用微软自己的Microsoft.Office.Interop.Excel系列，word、excel等都可以直接转pdf。 bool result = false; Excel.XlFixedFor...
--老马识途远
3. Re:C#如何用最简单方法调用Python？
如果通过print输出日志数据的话，当print数据量大的时候，python就不会实时输出了，他会自行决定什么时候输出。也就是隔一两分钟没有输出，然后哗的一下子出来很多。 所以为了实现实时输出效果，需...
--猫叔Vincent
4. Re:除了Visual Studio, Rider也能开发.NET（文中有料）

https://www.cnblogs.com/xbhp/p/17411590.html

2/3

- 代码混淆加密保护我们的C#程序
- C#代码混淆工具

非常感谢

--?+.史莱姆|

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

5. Re:一文读懂.NET WebAPI中FromRoute、FromQuery、FromBody的关键角色与用法

不错学习到了
.Net兼职社区

--Metadata科技屋

博客园 © 2004-2025



浙公网安备 33010602011771号 浙ICP备2021040463号-3