



网站 新帖 搜索 帮助

快捷导航

请输入搜索内容

帖子

领取今日签到奖励

网站 【软件安全】 『脱壳破解区』

返回列表

1

2

... 12

1 / 12 页

[.NET逆向] [入门级]使用x64dbg暴打非托管壳（TMD，SE，EVB等） [复制链接]

wwh1004 2018-7-8 18:22



本帖最后由 wwh1004 于 2018-7-9 10:56 编辑

TMD=ThemIDA，SE=Safengine Shielden，EVB=Enigma Virtual Box为什么说是入门级，因为你需要会使用工具和使用x64dbg下断点，别的你都不需要会了，文章中演示的都属于简单情况。

什么？完全不会x64dbg？看看我上午发的这篇教程，只需要看如何下断点，2分钟包学会：

<https://www.52pojie.cn/thread-762711-1-1.html>

文章中用到的工具：

AssemblyRebuilder by wwh1004: <https://www.52pojie.cn/thread-699172-1-1.html>

ExtremeDumper by wwh1004: <https://www.52pojie.cn/thread-712611-1-1.html>

Universal Fixer by CodeCracker: <https://www93.zippyshare.com/v/wyCquy7N/file.html>

这个文件被包含在上面这个压缩包里（路径

ToolsArchive\_Part1.zip\PC\_CONTENTS\MY\_NET\_TOOLS\Unpackers\Universal\_Fixer.exe）。压缩包里有CodeCracker大神无偿分享的各种工具，.NET逆向必备神器级工具包

x64dbg: <https://github.com/x64dbg/x64dbg/releases>

注意装上插件SharpOD并按SharpOD的推荐设置设置好

dnSpy by 0xd4d: <https://github.com/0xd4d/dnSpy/releases>

视频在线观看: <https://www.bilibili.com/video/av26414595>

视频下载: <https://pan.baidu.com/s/1LToTluQ2-TtbRjJQLcVcCg> 密码: ffc4

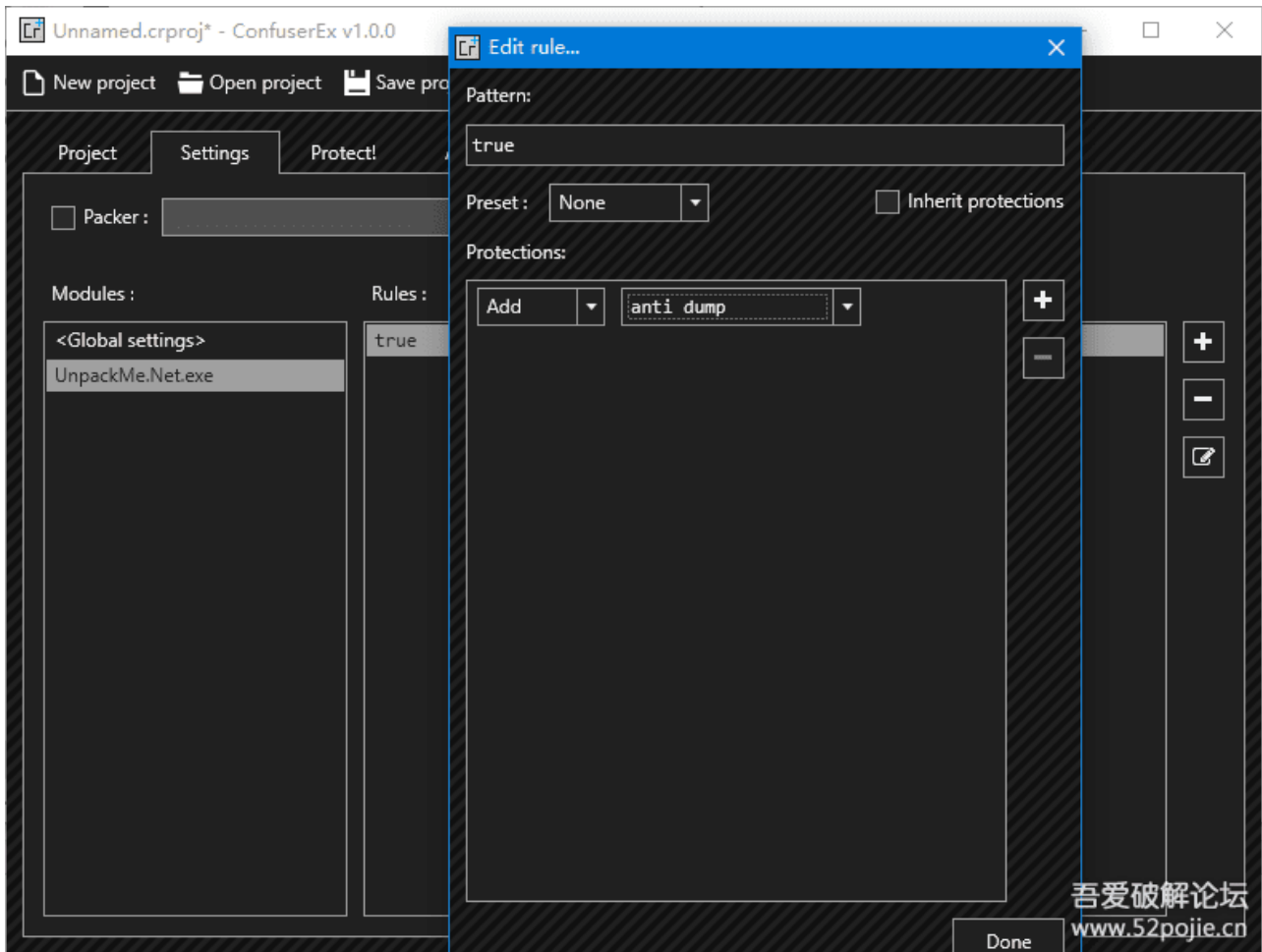
文章看不懂的地方务必手动转到视频对应位置查看

下方还有红字标明的原理，算是核心内容了

补充一个实战，脱画眉大神的"[UnPackMe]【.Net UnpackMe】免费的壳，也可以加出别人脱不掉的效果！不服来战，坐等被虐。”，<https://www.52pojie.cn/thread-459231-1-1.html>。

视频在线观看: <https://www.bilibili.com/video/av26458567>

视频下载: <https://pan.baidu.com/s/1mM3TervBoyXqdCrK-D6zEA> 密码: 8xt8

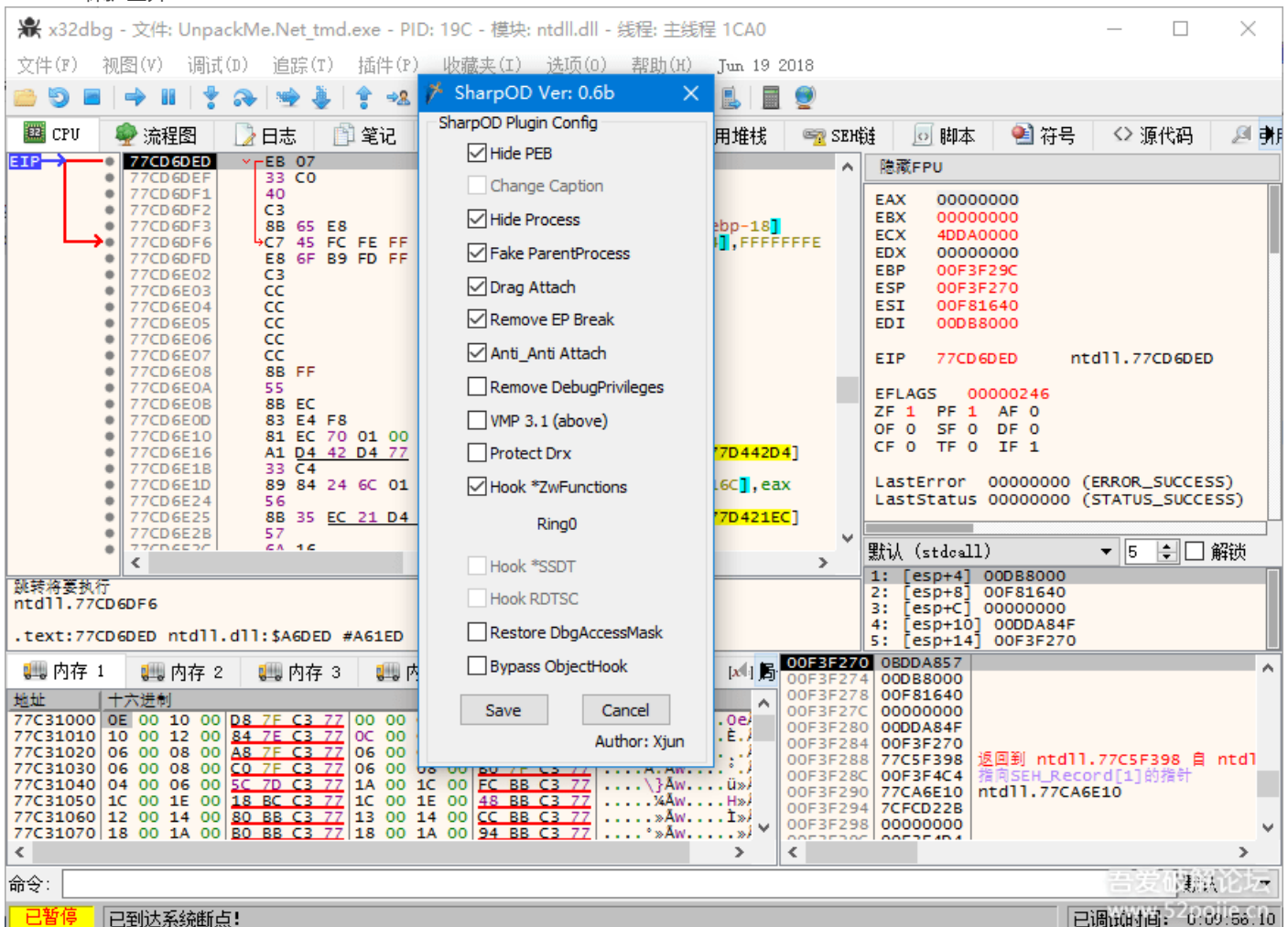


先用ConfuserEx加一层壳，只是演示如何脱非托管壳，所以ConfuserEx只设置加anti dump

然后开始演示Themida：

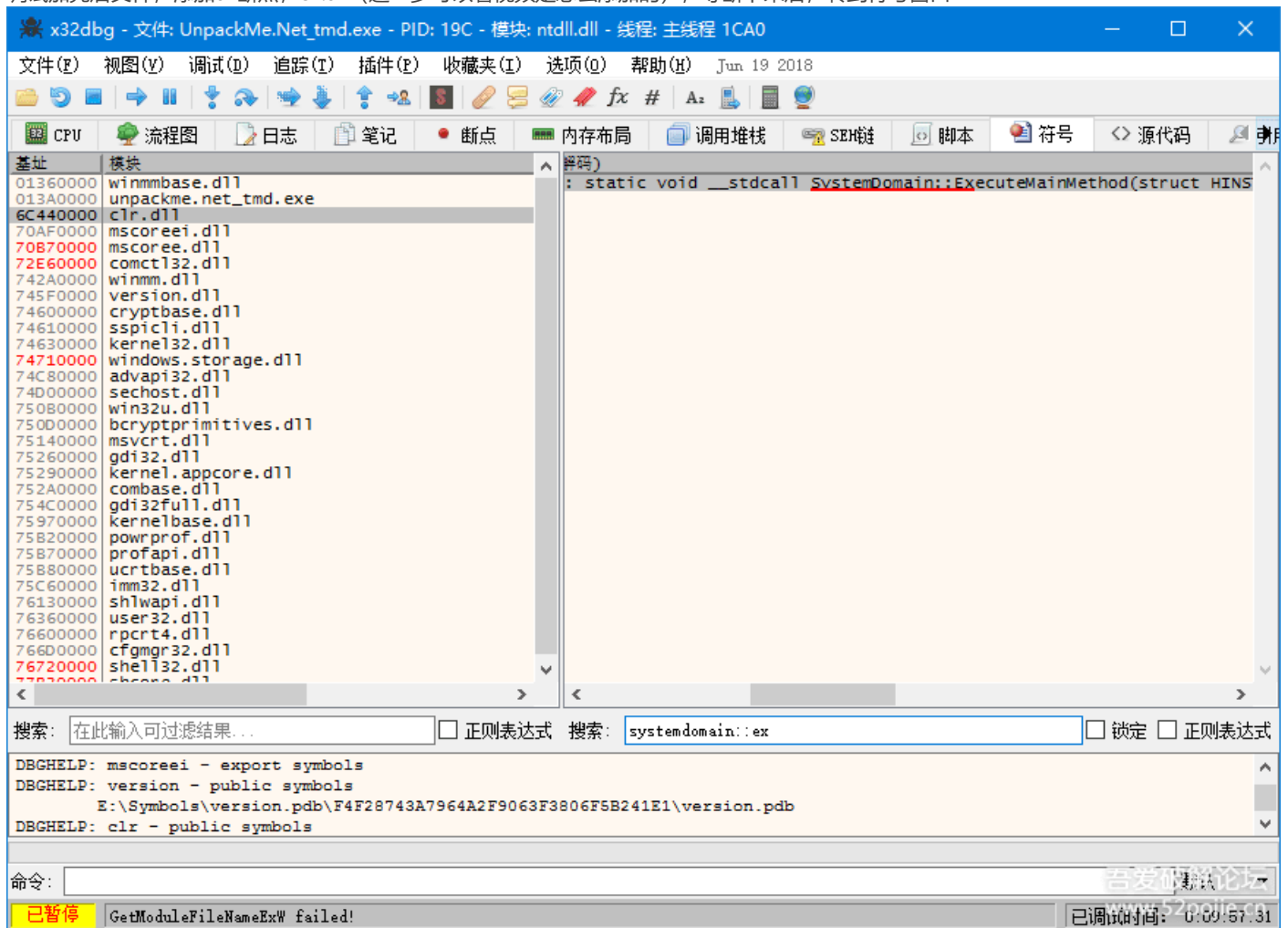


Themida保护全开

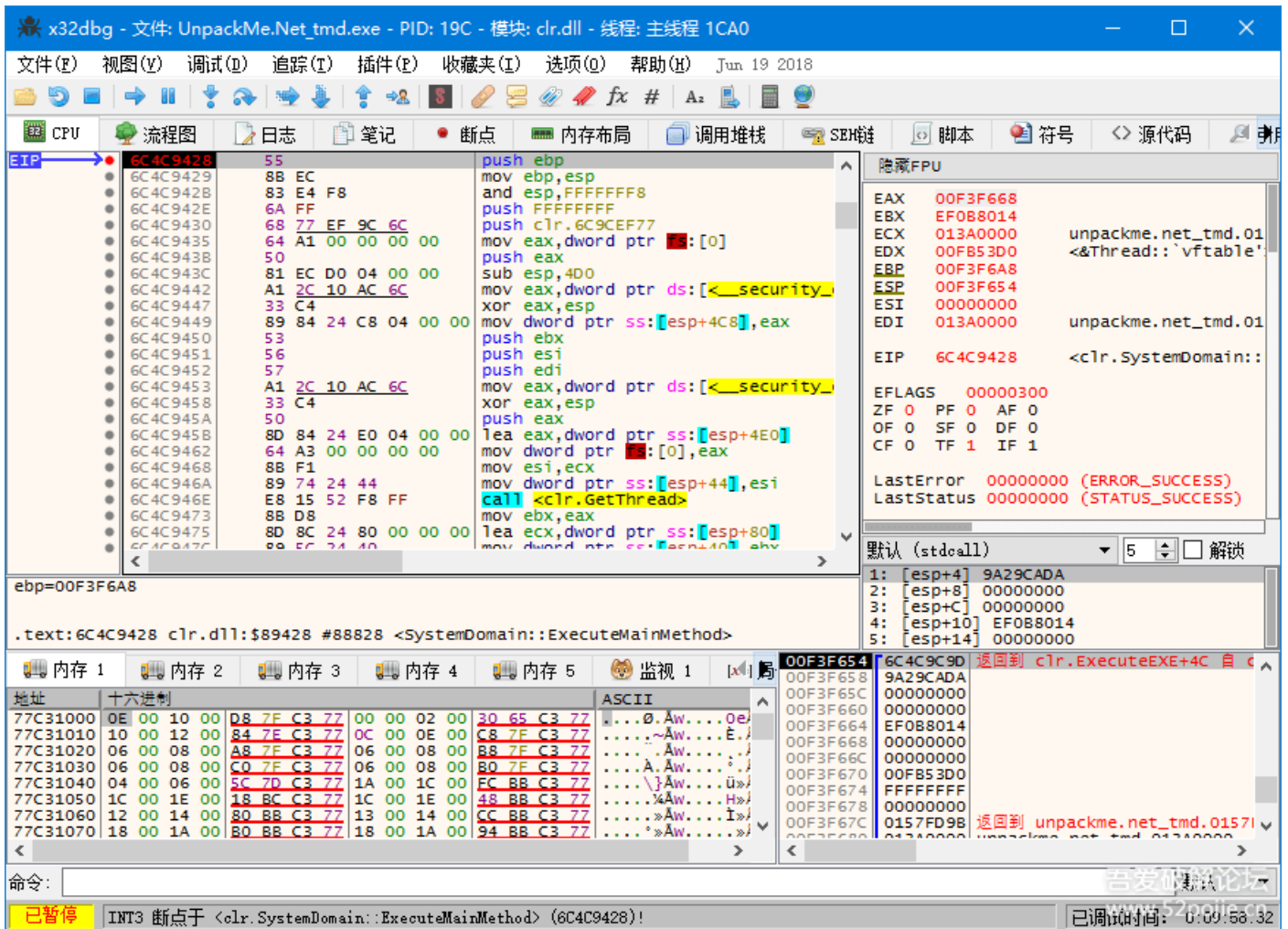


配置好x64dbg

调试加壳后文件, 添加dll断点, clr.dll (这一步可以看视频是怎么添加的), 等断下来后, 转到符号窗口

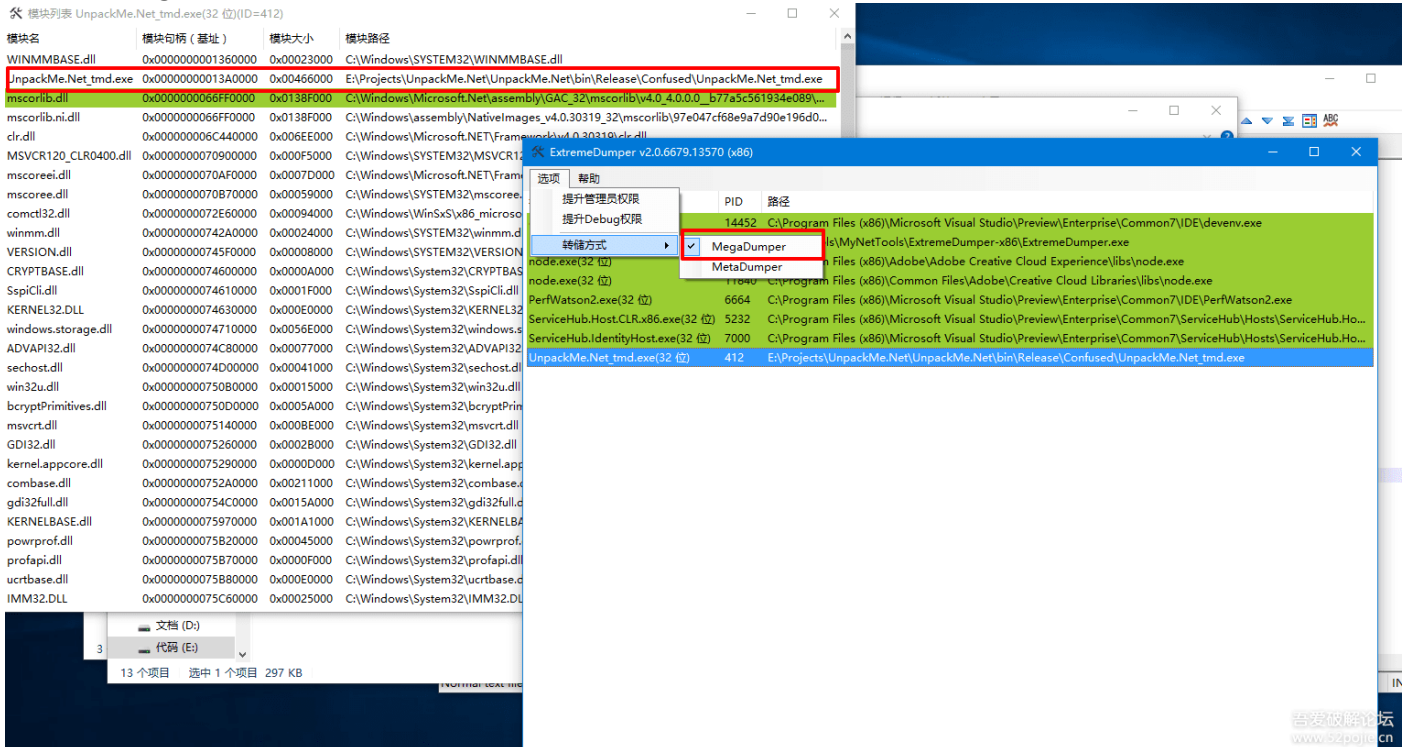


点击clr.dll (右键加载符号文件, 我的已经下载过了, 没下载过的可能要等10分钟+), 搜索SystemDomain::ExecuteMainMethod, 在这里下断点



## 成功断下

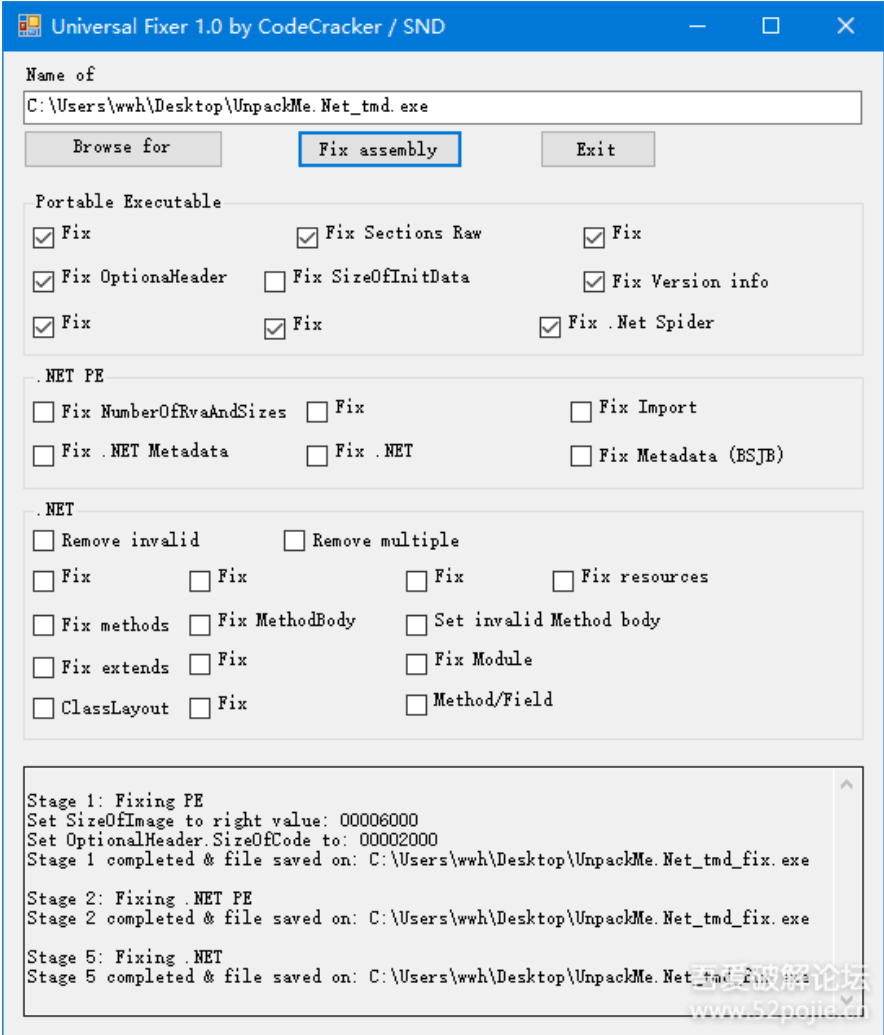
接下来没x64dbg什么事情了，是不是非常简单，完全只需要用工具



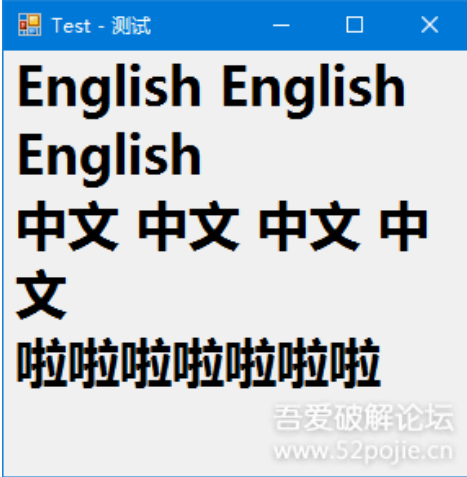
把ExtremeDumper的转储方式设置为MegaDumper（用MegaDumper也可以，只不过觉得那个界面太难看了，而且还很卡...）

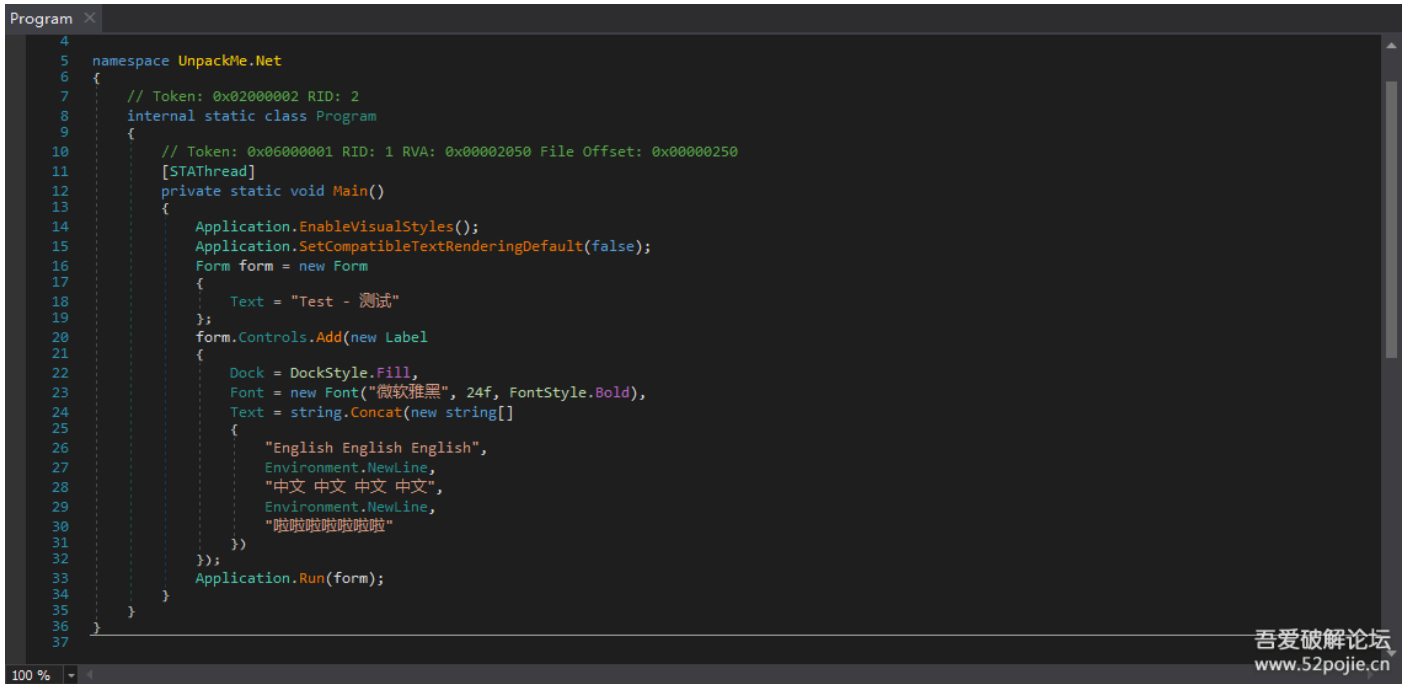
右键, 查看模块, Dump主模块





使用Universal Fixer修复，一般把我选中的这些勾上，别的都不选就可以了  
运行正常，反编译也正常：

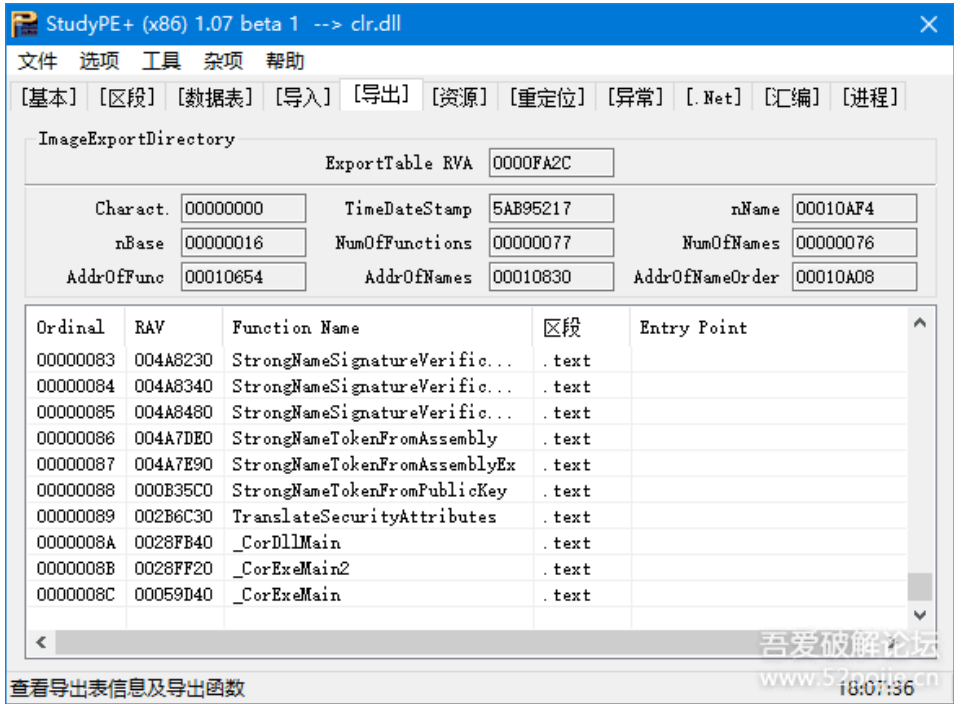




### 原理：

在SystemDomain::ExecuteMainMethod处下断点，是一个非常折中的方案，不会过于深入clr内部，也不会随随便便被hook住运行到这里的时候主程序集已经被这些非托管壳还原到对应位置为什么这说，请看下图

这是主线程的调用堆栈（我在主线程里运行了消息泵，所以暂停住了）  
可以看到底端是不知名的地址，应该是TMD之类的壳生成的  
再向上看，有个clr\_CorExeMain和clr\_CorExeMainInternal，clr\_CorExeMain是导出函数



有被hook的风险，所以在这里下断点不适合，clr\_CorExeMainInternal其实也可以，但是感觉不太好（没测试过）  
然后是

clr.RunMain  
clr.Assembly::ExecuteMainMethod

```
clr.SystemDomain::ExecuteMainMethod
clr.ExecuteEXE
.....
```

这4个应该都可以，都是比较安全的，几乎不可能被hook住，直接dump下这时的主模块就没啥问题了  
不过我只测试了clr.SystemDomain::ExecuteMainMethod，测试了3个壳（2个强壳，1个打包器），都是可以的

Shielden：  
Dump前操作一样，只不过Dump之后要用CFF Explorer手动清理导入表  
至于为什么要清理，看@brezeer 的<https://www.52pojie.cn/thread-439154-1-1.html>  
或者也可以使用我的AssemblyRebuilder，重建程序集，程序体积完全恢复正常，一切OK（如果还有强壳包裹了就不OK了，AssemblyRebuilder会让接下来的脱壳无法进行，这时请考虑手动清理导入表）

Enigma Virtual Box：  
没啥说的了，会了TMD SE，这个也就会了

接下来我放出之前加壳的文件，大家可以自己尝试一下  
链接: [https://pan.baidu.com/s/1CjtqeTTEmMi\\_F1TR7hO8aQ](https://pan.baidu.com/s/1CjtqeTTEmMi_F1TR7hO8aQ) 密码: 3261

免费评分

	吾爱币	热心值	理由	收起
 gogame	+ 1	+ 1	非常感谢教程及工具，请教有一个TMD加壳的.net，加载不到clr.dll模块怎么办.	
 灵魂行者	+ 1	+ 1	谢谢@Thanks!	
 XXTK	+ 2	+ 1	受益匪浅!	
 78054465	+ 1	+ 1	感谢发布原创作品，吾爱破解论坛因你更精彩! .NET大神	
 nio	+ 1	+ 1	热心回复!	
 给力的小黄瓜	+ 1	+ 1	谢谢@Thanks!	
 一个老妖	+ 1	+ 1	谢谢@Thanks!	
 mlwy	+ 1	+ 1	谢谢@Thanks!	
 華冷枫	+ 1	+ 1	我很赞同!	
 netle8	+ 1	+ 1	谢谢@Thanks!	
 Shock	+ 3	+ 1	用心讨论，共获提升!	
 L.S. LzSkyline	+ 2	+ 1	鼓励转贴优秀软件安全工具和文档!	
 陪伴时光		+ 1	谢谢@Thanks!	
 hxx26	+ 1	+ 1	谢谢@Thanks!	
 何为人生888	+ 1	+ 1	我很赞同!	
 zzm1223	+ 1	+ 1	谢谢@Thanks!	
 winooxx	+ 1	+ 1	谢谢@Thanks!	
 御剑	+ 3	+ 1	谢谢@Thanks!	
 zejax	+ 1	+ 1	热心回复!	
 tchivs	+ 2	+ 1	谢谢@Thanks!	
 a5606495	+ 1	+ 1	用心讨论，共获提升!	



	sunnylds7	+ 1	+ 1	用心讨论，共获提升！
	smile1110	+ 2	+ 1	我很赞同！
	azahod	+ 1	+ 1	热心回复！
	yyhf	+ 1	+ 1	用心讨论，共获提升！
	pj5008	+ 1	+ 1	谢谢@Thanks！
	Lsflhd	+ 1		我很赞同！
	Ravey	+ 1	+ 1	谢谢@Thanks！
	凉游浅笔深画眉	+ 1	+ 1	用心讨论，共获提升！
	MaxMadcc	+ 1	+ 1	谢谢@Thanks！
	xjun	+ 3	+ 1	学习学习
	依旧沉沉	+ 1	+ 1	有没有像我一样，连标题都没看懂的
	fanvalen	+ 1	+ 1	感谢发布原创作品，吾爱破解论坛因你更精彩！
	zhczf	+ 1	+ 1	我很赞同！
	zhaotianrun	+ 2	+ 1	太好了！！
	m4n0w4r	+ 1	+ 1	谢谢@Thanks！
	独行风云	+ 1	+ 1	谢谢@Thanks！
	jgs	+ 1	+ 1	谢谢@Thanks！
	vipcrack	+ 1	+ 1	录像文件能不能百度一份，谢谢
	xindong8	+ 1	+ 1	谢谢@Thanks！
	qaz003	+ 1	+ 1	6666666666
	pk8900	+ 1	+ 1	谢谢分享.net教程，这块一直没研究过，慢慢学习。
	zeknight	+ 1	+ 1	这个非常牛逼了~~楼主的经验太有用了
	yhz	+ 1	+ 1	热心回复！
	青春ノ易逝	+ 1	+ 1	谢谢@Thanks！
	QB56	+ 1	+ 1	热心回复！
	xinkui	+ 1	+ 1	谢谢@Thanks！
	adingtao11	+ 1	+ 1	好久没看到如此详细的教程了
	freesoft00	+ 1	+ 1	我很赞同！

[查看全部评分](#)

本帖被以下淘专辑推荐:

· [学习及教程](#) | 主题: 1306, 订阅: 986· [Aarow](#) | 主题: 1680, 订阅: 275

发帖前要善用【论坛搜索】功能，那里可能会有你要找的答案或者已经有人发布过相同内容了，请勿重复发帖。

回复

举报



SCL 2018-7-12 18:54

推荐

楼主真厉害。。。。

【吾爱破解论坛总版规】 - [让你充分了解吾爱破解论坛行为规则]

回复

支持 0

免费评分 举报



zxzt 2019-4-18 21:25

推荐

吾爱破解论坛没有任何官方QQ群，禁止留联系方式，禁止任何商业交易。

dump下了，更之前的代码一样，是我哪里操作错了

An exception occurred when decompiling this method (06000267)

ICSharpCode.Decompiler.DecompilerException: Error decompiling smgw.ModuleLicense smgw.AuthHelper::License()

---> System.NullReferenceException: 未将对象引用设置到对象的实例。

在 ICSharpCode.Decompiler.ILAst.ILAstOptimizer.IntroducePropertyAccessInstructions(ILExpression expr, ILExpression parentExpr, Int32 posInParent) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILAstOptimizer.cs:行号 1589

在 ICSharpCode.Decompiler.ILAst.ILAstOptimizer.IntroducePropertyAccessInstructions(ILNode node) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILAstOptimizer.cs:行号 1579

在 ICSharpCode.Decompiler.ILAst.ILAstOptimizer.Optimize(DecompilerContext context, ILBlock method, AutoPropertyProvider autoPropertyProvider, StateMachineKind& stateMachineKind, MethodDef& inlinedMethod, AsyncMethodDebugInfo& asyncInfo, ILAstOptimizationStep abortBeforeStep) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILAstOptimizer.cs:行号 244

在 ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(IEnumerable`1 parameters, MethodDebugInfoBuilder& builder) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:行号 123

在 ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context, AutoPropertyProvider autoPropertyProvider, IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb, MethodDebugInfoBuilder& stmtsBuilder) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:行号 88

--- 内部异常堆栈跟踪的结尾 ---

在 ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context, AutoPropertyProvider autoPropertyProvider, IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb, MethodDebugInfoBuilder& stmtsBuilder) 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:行号 92

在 ICSharpCode.Decompiler.Ast.AstBuilder.<>c\_\_DisplayClass89\_1.<AddMethodBody>b\_\_0() 位置

C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstBuilder.cs:行号 1531

如何升级？如何获得积分？积分对应解释说明！

回复

支持

免费评分 举报



dwzh007 2018-7-8 18:50

4#

《站点帮助文档》有什么问题来这里看看吧，这里有你想知道的内容！

大神请分享 SharpOD 插件，谢谢

免费评分

	吾爱币	理由	收起
 wwh1004	+ 1	百度比我回复快得多www.52pojie.cn/thread-628837-1-1.html	
<a href="#">查看全部评分</a>			

呼吁大家发布原创作品添加吾爱破解论坛标示！

回复支持

免费评分 举报

 **dwh007** 2018-7-8 19:095#

dwh007 发表于 2018-7-8 18:50

大神请分享 SharpOD 插件，谢谢

谢谢，找到了，

如何快速判断一个文件是否为病毒！

回复支持

免费评分 举报

 **冥界3大法王**👑 2018-7-8 19:486#

插件还是自己汉化过的给力啊~~提示也更像人话儿。

回复支持


免费评分 举报

 **fandongjie** 2018-7-8 20:337#

谢谢分享，学习了，高难度啊

回复支持

免费评分 举报

 **170077000** 2018-7-8 20:498#

这个可以有 X32还没用过 听说比OD好很多 可是使用的视频好像很少

回复支持

免费评分 举报

 **kilkilo502** 2018-7-8 20:539#

学习了多谢老师的教程

回复支持

免费评分 举报

 **梦幻110110** 2018-7-8 21:1410#

楼主能不能发下你配置好的x64dbg谢谢

回复支持

免费评分 举报

 **wwh1004**👑 2018-7-8 21:2411#


梦幻110110 发表于 2018-7-8 21:14

楼主能不能发下你配置好的x64dbg谢谢

下载原版x64dbg，装上SharpOD，按SharpOD压缩包里的图片设置，自己动手吧

回复支持

免费评分 举报



QB56

2018-7-8 21:33

用心的教程，热心奉上

12#

回复支持

免费评分 举报

下一页 »

返回列表

1

2

... 12

1 / 12 页

	高级模式

验证问答  换一个

发表回复

警告：本版块禁止灌水或回复与主题无关内容，违者重罚！

☐ 回帖并转播

☐ 回帖后跳转到最后一页

免责声明：  
吾爱破解所发布的一切破解补丁、注册机和注册信息及软件的解密分析文章仅限于学习和研究目的；不得将上述内容用于商业或者非法用途，否则，一切后果请用户自负。本站信息来自网络，版权争议与本站无关。您必须在下载后的24个小时之内，从您的电脑中彻底删除上述内容。如果您喜欢该程序，请支持正版软件，购买注册，得到更好的正版服务。如有侵权请邮件与我们联系处理。

Mail To:Service@52PoJie.Cn