桌面应用保护

本地程序保护

`Windows .exe/.dll 保护`

`Linux 可执行程序/.so 保护`

`macOS 可执行程序/.dylib 保护`

功能

代码混淆、代码虚拟化、代码加密等多种方式保护代码逻辑

加密代码中使用的敏感字符串

校验程序内存完整性，防止代码被篡改，防止内存补丁

对程序的代码数据压缩加密，防止被 IDA Pro 等工具反编译

检测调试器，防止程序被调试分析

加密程序访问的配置、资源等文件，防止被窃取

应用场景

加密方案被破解

> 重要数据和文件的加密逻辑，如果不保护，很容易被篡改或窃取。

通讯加密被破解

> 通讯加密算法和协议，防止机器人和外挂的产生

核心代码被窃取

> 本地程序也可以被 `IDA Pro` 等工具反编译为 `C` 语言代码

资源、数据文件被窃取

> 程序访问的重要数据文件和资源如果不加密，可以被轻易窃取和修改。

代码逻辑被篡改

> 版权验证代码或重要的程序逻辑如果不保护，很容易被篡改，产生破解补丁或外挂。

核心技术

代码混淆

> 对原始汇编指令通过等价变量、指令拆分、间接调用、虚拟分支、立即数加密等手段，转换为更复杂的汇编指令，再配合单步寄存器检测、指令校验等方式检测并设置调试陷阱，阻碍逆向分析。

代码虚拟化

将原始汇编指令翻译为自定义的虚拟机指令，跳转到自定义的虚拟机中执行，每次保护生成的虚拟机指令随机，且对虚拟机解释器再度混淆，安全性极高。

内存校验

运行时对程序内存进行完整性校验，并提供 SDK 标签的方式，可对内存进行动态校验，防止程序被篡改。

压缩

对程序的代码或数据段整体压缩并加密，防止被反编译。

反调试

通过平台相关API、数据结构和寄存器，检测调试器，防止动态调试。

资源/文件加密

加密 PE 资源节，同时支持对程序访问的文件（包括配置文件、脚本、音视频等资源）加密，防止文件被窃取。

产品亮点

高安全性

多重保护，函数经过混淆、虚拟化保护后，无法被还原。

配置灵活

保护选项灵活可配，平衡安全性和性能。

简单易用

界面操作简单，并集成性能分析工具，方便过滤筛选方法。支持命令行自动化保护，方便 CI 集成。

高效服务

便捷、高效的技术支持。

效果展示

对比图

支持范围

Windows、Linux、macOS 可执行程序、动态库(.exe/.dll/.so/.dylib)

Apple M1 设备原生程序

Go 语言程序

Electron 应用

Windows 驱动程序（.sys）【即将发布】

Windows ARM 架构支持【即将发布】

.NET 程序保护

`.NET .exe/.dll 保护`

`.NET Core 程序保护`

功能

混淆代码控制流，干扰逆向分析

加密 IL 代码，防止 DnSpy、ILSpy 等工具反编译

代码加密、JIT 加密、压缩等多重加密，防止 de4dot 等工具还原代码

混淆 .NET 类名、方法名

应用场景

核心代码被窃取

> .NET 程序由于包含丰富的元数据信息，如果不加保护，反编译后可以达到与源代码几乎相同的效果。

加密方案被破解

> 重要数据和文件的加密逻辑，如果不保护，很容易被篡改或窃取。

通讯加密被破解

> 通讯加密算法和协议，防止机器人和外挂的产生。

资源、数据文件被窃取

> 程序访问的重要数据文件和资源如果不加密，可以被轻易窃取和修改。

代码逻辑被篡改

> 版权验证代码如果不保护，很容易被篡改，产生破解补丁。

核心技术

代码混淆

通过虚拟分支、间接调用、立即数加密等方式，将 `IL` 代码转换为难以阅读的代码。

代码加密

加密方法的`IL`代码，运行时通过 `.NET` 动态方法技术反射调用方法，可对抗内存 `dump` 和 `de4dot` 等反混淆工具。

名称混淆

将 `.NET` 类名、方法名等修改为无意义字符，干扰逆向分析。

压缩

对代码段压缩加密，防止被反编译。

JIT 加密

加密 `.NET` 所有方法，仅在 `JIT` 编译过程解密，每次只解密一个方法，无法被整体 `Dump`。

产品亮点

高安全性

`IL` 函数级保护与 `JIT` 加密、压缩构成三重保护，安全性高。

配置灵活

保护选项灵活可配，平衡安全性和性能。

简单易用

界面操作简单，并集成性能分析工具，方便过滤筛选方法。支持命令行自动化保护，方便 `CI` 集成。

高效服务

便捷、高效的技术支持。

效果展示

对比图

支持范围

支持 .NET 2.0 及以上

支持 .NET Core 3.0 及以上（部分功能暂不支持 Linux 和 macOS）

支持 VB .NET

Java 程序保护

`桌面平台 .jar、.class、.war 保护`

功能

加密 jar/war/class）中的Java 方法，防止被 jg-gui 等工具反编译。

虚拟化保护 Java 中的 JVM 字节码，由自定义的Native解释器执行，防止任何工具破解。

保护 Java SDK，支持 IDE 中直接引用。

应用场景

核心代码被窃取

`Java` 程序由于包含丰富的类、方法、调试信息，如果不加保护，反编译后可以达到与源代码几乎相同的效果。

通讯加密被破解

通讯加密算法和协议如果不保护，容易被模拟调用，可能会对开发者造成损失。

代码逻辑被篡改

`Java` 代码如果不保护，很容易被篡改二次开发，损害开发者利益。

核心技术

方法加密

加密 `Java` 方法中的 `JVM` 字节码，仅在方法即时编译（`JIT`）过程中解密，每次仅解密一个方法在内存，无法整体 `Dump`。

Java 虚拟化

将 `JVM` 字节码转换为自定义的虚拟机指令，运行时跳转至 `Native` 虚拟机中执行，安全强度高，无法被任何已知工具还原出原始 `Java` 代码。

反调试

通过平台相关的反调试技术，防止进程被动态调试。

产品亮点

高安全性

高强度的虚拟化保护，可以对抗目前已知的任何工具还原代码。

配置灵活

保护选项灵活可配，可自由选择要保护的 `Java` 方法，平衡安全性和性能。

简单易用

界面操作简单，支持命令行自动化保护，方便 CI 集成。

多平台支持

支持所有主流操作系统，支持绝大部分容器环境。

高效服务

便捷、高效的技术支持。

效果展示

对比图

支持范围

支持 spring framework 等框架。

支持 tomcat。

支持所有主流操作系统和绝大部分窗口框架。

Unity 程序保护

`C# 程序集保护`

`il2cpp 保护`

`资源加密`

功能

保护 C# 程序集（Assembly-CSharp.dll），防止C#代码被反编译，防止 de4dot 等工具还原代码。

加密 global-meta-data.dat 文件，并通过多种保护技术防止 il2cppDumper 等工具 Dump .NET 元数据信息。

加密 AssetBundle 资源，防止 Asset Studio 等工具反编译。

通过平台相关技术，防止程序被动态调试。

应用场景

程序被反编译

Unity 程序集 DLL 实际为标准的 .NET 文件格式，可以直接反编译为 C# 代码，暴露代码逻辑，很容易滋生外挂。

资源被提取或篡改

Unity 程序 Asset Bundle 资源可以被提取重打包，修改场景，破坏程序正常功能。

程序集被篡改

程序集被反编译后，根据类名和方法名，可以很容易的定位并修改逻辑。

il2cpp 代码被篡改

il2cpp 虽然被编编译为 Native 模块，但可以通过 global-meta-data 文件轻松定位与 C# 代码关联的 Native 代码，也容易被篡改。

核心技术

程序集保护

对程序集中所有方法的 IL 代码加密，通过修改 Mono 引擎，运行时在 JIT 编译过程中解密，每次仅有一个方法被解密，并解密到随机的内存地址，保证加密后的程序无法被 Dump 出整个程序集。

资源加密

加密 Unity Asset Bundle 资源，在 unity 引擎中解密，防止 Asset Bundle 资源被反编译打包，支持资源热更新。

global-meta-data 加密

加密 il2cpp 的 global-meta-data 文件，并对内部结构进行混淆处理，防止运行时在内存中直接解析。

il2cpp 保护

对 il2cpp.so 或 GameAssembly.dll 处理，去除导出函数，并对内存进行完整性校验，防止篡改。

反调试

通过平台相关API、数据结构和寄存器，检测调试器，防止动态调试。

产品亮点

高安全性

高强度的虚拟化保护，可以对抗目前已知的任何工具还原代码。

配置灵活

保护选项灵活可配，可自由选择要保护的 Java 方法，平衡安全性和性能。

简单易用

界面操作简单，支持命令行自动化保护，方便 CI 集成。

快速迭代

快速的产品迭代，保障产品的安全性，适配 Unity 最新版本。

高效服务

> 便捷、高效的技术支持。

效果展示

对比图

支持范围

支持 Unity 5.4 及以上。

支持 Unity 2017 及以上。

支持 il2cpp 编译方式。

不支持 Debug 模式编译的程序。

脚本语言保护

`python/php/javascript加密`

功能

> 能做什么

通过透明加解密的方式，加密脚本语言文件（.py/.php/.js/.html 等）

结合反调试等功能，防止内存调试。

应用场景

> 解决的问题，带来的好处

核心技术

> 技术手段亮点

效果展示

对比图

移动应用保护

安卓应用保护

`DEX 加密、虚拟化保护`

`apk 资源加密、保护so库`

`apk 反调试、ptrace 防注入`

功能

对 DEX 文件加密并隐藏，防止被 jadx, jeb 等工具反编译。

对 DEX 中的方法进行虚拟化等技术保护，保护后的代码无法被还原，防止逆向分析。

加密 APK 中的图片、配置、脚本等资源文件。

保护 APK 中的 SO 库，对代码段加密，隐藏导入导出符号。

对开发者签名校验，防止 APK 二次打包。

防止 IDA Pro、gdb、jeb 等工具调试对 APK 进行 Java 层调试和 Native 调试。

防止 ptrace 注入和附加进程，防内存 dump。

应用场景

解决的问题，带来的好处

核心技术

DEX 加密

> 对 DEX 文件整体加密并隐藏，防止反编译。

DEX 虚拟化

> 将 DEX 方法中的字节码转换为自定义的虚拟机指令，由自定义解释器解释执行，保护后无法被还原。

资源加密

> 加密 apk 中的图片，配置，脚本等资源文件，防止被窃取。

ptrace防注入

> 通过双进程 ptrace 守护技术，防止其它进程对 APK 进程附加调试或注入。

反调试

> 多种系统相关的检测技术检测调试器，发现调试器后清场退出。

SO 库保护

> 对 SO 库中的代码段压缩加密，隐藏导入导出函数。

产品亮点

高安全性

> DEX 虚拟化与指令级混淆虚拟化两重保护，再结合反调试、防注入等运行时防护功能，安全性高。

配置灵活

> 保护选项灵活可配，可自由选择要保护的 Java 方法，平衡安全性和性能。

简单易用

> 界面操作简单，支持命令行自动化保护，方便 CI 集成。

离线保护【】

> 保护过程可完全离线，不会上传任何信息，保护后的程序不包含任何与保护无关的代码。

高效服务

> 便捷、高效的技术支持。

效果展示

对比图

安卓 SDK 保护

`安卓 jar/aar 保护`

功能

对 JAR 中的方法进行虚拟化等技术保护，保护后的代码无法被还原，防止逆向分析。

保护 AAR 中的 SO 库，对代码段加密，隐藏导入导出符号，防止反编译。

应用场景

核心技术

Java 虚拟化保护

> 将 `Java` 方法中的 `JVM` 字节码转换为自定义的虚拟机指令，由自定义解释器解释执行，保护后无法被还原。

SO 库保护

> 对 `SDK` 中 `SO` 库的代码段压缩加密，隐藏导入导出函数。

效果展示

对比图

安卓 SO 保护

`so 库 x86/arm 混淆、虚拟化`=====

`so 库完整性校验、文件加壳压缩`

功能

指令级代码混淆、代码虚拟化、代码加密等多种方式保护代码逻辑

加密代码中使用的敏感字符串

校验程序内存完整性，防止代码被篡改

对程序的代码数据压缩加密，防止被反编译

检测调试器，防止程序被调试分析

应用场景

解决的问题，带来的好处

核心技术

代码混淆

对原始汇编指令通过等价变量、指令拆分、间接调用、虚拟分支、立即数加密等手段，转换为更复杂的汇编指令，再配合单步寄存器检测、指令校验等方式检测并设置调试陷阱，阻碍逆向分析。

代码虚拟化

将原始汇编指令翻译为自定义的虚拟机指令，跳转到自定义的虚拟机中执行，每次保护生成的虚拟机指令随机，且对虚拟机解释器再度混淆，安全性极高。

内存校验

运行时对程序内存进行完整性校验，并提供 SDK 标签的方式，可对内存进行动态校验，防止程序被篡改。

压缩

对程序的代码或数据段整体压缩并加密，防止被反编译。

反调试

通过平台相关API、数据结构和寄存器，检测调试器，防止动态调试。

资源/文件加密

加密资源，同时支持对程序访问的文件（包括配置文件、脚本、音视频等资源）加密，防止文件被窃取。

产品亮点

高安全性

指令级代码虚拟化技术，安全性更高，同时支持 x86/x64/arm32/arm64 架构，保护后的代码无法被还原。

配置灵活

保护选项灵活可配，平衡安全性和性能。

简单易用

无需配置开发环境，界面操作简单，支持命令行自动化保护，方便 CI 集成。

高效服务

便捷、高效的技术支持。

效果展示

安卓/iOS Unity 保护

`C# 程序集保护、il2cpp 保护`

`资源加密、防注入`

功能

保护 C# 程序集（Assembly-CSharp.dll），防止C#代码被反编译，防止 de4dot 等工具还原代码。

加密 global-meta-data.dat 文件，并通过多种保护技术防止 il2cppDumper 等工具 Dump .NET 元数据信息。

加密 AssetBundle 资源，防止 Asset Studio 等工具反编译。

通过平台相关技术，防止程序被动态调试。

防调试器附加，防注入，防内存 dump。

应用场景

核心技术

程序集保护

> 对程序集中所有方法的 `IL` 代码加密，通过修改 `Mono` 引擎，运行时在 `JIT` 编译过程中解密，每次仅有一个方法被解密，并解密到随机的内存地址，保证加密后的程序无法被 `Dump` 出整个程序集。

资源加密

> 加密 `Unity Asset Bundle` 资源，在 `unity` 引擎中解密，防止 `Asset Bundle` 资源被反编译打包。

global-meta-data 加密

> 加密 `il2cpp` 的 `global-meta-data` 文件，并对内部结构进行混淆处理，防止运行时在内存中直接解析。

il2cpp 保护

> 对 `il2cpp.so` 或 `GameAssembly.dll` 处理，去除导出函数，并对内存进行完整性校验，防止篡改。

反调试

> 通过平台相关API、数据结构和寄存器，检测调试器，防止动态调试。

防注入

> 通过双进程 `ptrace` 守护技术，防止其它进程对附加调试或注入。

产品亮点

高安全性

```

```

配置灵活

```

```

简单易用

```
界面操作简单，支持命令行自动化保护，方便 CI 集成。
```

快速迭代

```
快速的产品迭代，保障产品的安全性，适配 Unity 最新版本。
```

高效服务

```
便捷、高效的技术支持。
```

效果展示

对比图

支持范围

支持 Unity 5.4 及以上。

支持 Unity 2017 及以上。

支持 il2cpp 编译方式。

不支持 Debug 模式编译的程序。

---

iOS 应用保护

`代码混淆虚拟化保护`

`内存完整性校验、反调试`

功能

代码混淆、代码虚拟化等多种方式保护代码逻辑，支持非 Bitcode 编译的Native指令。

校验程序内存完整性，防止代码被篡改

检测调试器，防止程序被调试分析

应用场景

解决的问题，带来的好处

核心技术

代码混淆

> 对原始汇编指令通过等价变量、指令拆分、间接调用、虚拟分支、立即数加密等手段，转换为更复杂的汇编指令，再配合单步寄存器检测、指令校验等方式检测并设置调试陷阱，阻碍逆向分析。

代码虚拟化

> 将原始汇编指令翻译为自定义的虚拟机指令，跳转到自定义的虚拟机中执行，每次保护生成的虚拟机指令随机，且对虚拟机解释器再度混淆，安全性极高。

内存完整性校验

> 运行时对程序内存进行完整性校验，防止程序被篡改。

反调试

> 通过系统层函数和数据，检测调试器，防止动态调试。

效果展示

对比图

IoT 应用保护

ARM Linux 应用保护

`so 库 x86/arm 混淆、虚拟化`

`so 库完整性校验、文件加壳压缩`

`数据文件、资源、脚本加密`

功能

指令级代码混淆、代码虚拟化、代码加密等多种方式保护代码逻辑。

加密代码中使用的敏感字符串。

校验程序内存完整性，防止代码被篡改。

对程序的代码数据压缩加密，防止被反编译，保护后几乎对性能无影响。

检测调试器，防止程序被调试分析。

加密程序访问的数据文件、资源、脚本等，防止被窃取。

应用场景

解决的问题，带来的好处

核心技术

代码混淆

对原始汇编指令通过等价变量、指令拆分、间接调用、虚拟分支、立即数加密等手段，转换为更复杂的汇编指令，再配合单步寄存器检测、指令校验等方式检测并设置调试陷阱，阻碍逆向分析。

代码虚拟化

将原始汇编指令翻译为自定义的虚拟机指令，跳转到自定义的虚拟机中执行，每次保护生成的虚拟机指令随机，且对虚拟机解释器再度混淆，安全性极高。

内存校验

运行时对程序内存进行完整性校验，并提供 SDK 标签的方式，可对内存进行动态校验，防止程序被篡改。

压缩

对程序的代码或数据段整体压缩并加密，防止被反编译。

反调试

通过平台相关API、数据结构和寄存器，检测调试器，防止动态调试。

资源加密

加密程序访问的数据文件、资源、脚本等，防止文件被窃取。

产品亮点

高安全性

首家指令级代码虚拟化技术，安全性更高，同时支持 x86/x64/arm32/arm64 架构，保护后的代码无法被还原。

配置灵活

保护选项灵活可配，平衡安全性和性能。

简单易用

无需配置开发环境，界面操作简单，支持命令行自动化保护，方便 CI 集成。

高效服务

便捷、高效的技术支持。

效果展示

对比图

安卓设备应用保护

安卓 so

功能

能做什么

应用场景

解决的问题，带来的好处

核心技术

技术手段

产品亮点

效果展示

对比图