

User Manual

Virbox Protector

Version 2.0



Copyright & Trademarks

The Virbox, Virbox LM, Virbox Elite 5, **Virbox Protector** with its technical documentation is copyrighted to present by ©Beijing SenseShield Technology Co., Ltd (SenseShield). All rights reserved.

The Virbox, Virbox LM, Virbox Elite 5, **Virbox Protector**, are Registered Trademarks of SenseShield in China and other countries.

All products referenced throughout this document are trademarks of their respective owners.

Disclaimer

All attempts have been made to make the information in this document complete and accurate. But we cannot guarantee everything is perfect, we will correct it in next version released in case some error has been found. SenseShield is not responsible for any direct or indirect damages or loss of business resulted from inaccuracies or omissions.

The specifications contained in this document are subject to change without notice.

Documentation Improvement

Any suggestion to this manual from you are welcome, We are glad to hear any feedback from you which will help us to continuously improve the documents quality and support and serve the developer to protect software products more efficiently.

Contact

Company: Beijing Senseshield Technology Co., Ltd

Address: Suite 510, Block C, Internet Innovation Center, Building 5, No.10, Xibeiwang East Road, Haidian District, Beijing China

Tel: +86-10-56730936

Fax: +86-10-56730936-8007

Sales: info@senselock.com;

Official Website: <https://lm-global.virbox.com/>

Virbox Developer Center (Virbox LM): <https://developer.lm-global.virbox.com/>

About this document

This document is designed to help Software Developer or Publisher to protect their Copyright or IP by protecting their software they would publish. And help the software resource supplier to protect their software resources.

Target User: The operation staff of Virbox Protector who is responsible for software copyright and IP protection.

Virbox Protector Update History:

Date	Version Number	Important Updates
2020.05	V 1.5	<ul style="list-style-type: none">▪ Add the function to support Dotnet Core3 program protection, including Windows, Linux, macOS▪ Add the function to support to protect Unity3D customized program set.▪ Support Anti-debug plugin function to Linux, ARM-linux, Android platform.
2020.10	V2.0	<ul style="list-style-type: none">▪ Add ARM-virtualization function to protect the ARM executable▪ Memory check function to PE and ELF program.▪ Add the compression function to DotNet DLL▪ Optimized the compression function to Dotnet executable▪ Support program protection to Java archive directly▪ Enhanced Anti-Runtrace function to ARM program.▪ Merged .vdata0 and .vadata1 segment▪ Support Control Flow Guard to PE program▪ Optimized the structure of the document

Table of Contents

1 Overview	7
1.1 Virbox Protector Introduction	7
1.2 Advanced and Secured Protection Technology	8
1.3 The program supported to be protected	8
2 Installation of Virbox Protector	11
2.1 Installation	11
2.2 License mode of Virbox Protector	12
2.2.1 License Verification with cloud license (For Trial User)	12
2.2.1.1 Verify license by Virbox User License Tool	12
2.2.1.2 Sign in by the Virbox Protector Interface	13
2.2.2 License Verification with soft license	15
2.2.2.1 Use Virbox Protector in online environment	15
2.2.2.2 Use Virbox Protector in offline environment	16
2.2.3 License Verification with EI5 dongle (For official user use dongle license)	21
3 Protection Function Introduction	23
3.1 Main Menu of Virbox Protector	23
3.2 Menu Bar	23
3.2.1 File	23
3.2.2 Protect	24
3.2.3 Plug-in	25
3.2.4 Log	26
3.2.5 Setting	26
3.2.6 Help	26
3.3 File Panel and Protection Panel	26
3.3.1 File Panel	26
3.3.2 Protection Panel	27
3.3.2.1 Basic Info	27
3.3.2.2 Function Options (function level protection)	27
3.3.2.3 Protection Options	37
3.3.2.4 Resource Encryption	43

3.3.2.5 Status bar	43
4 The Mechanism of software protection	45
4.1 Protect the PE program (application) and DLL.....	45
4.2 Protect the interpreter and code resource file (Python, PHP, etc.).....	46
4.3 Make the protection scheme for your software	46
5 Protection Example & Use Case	49
5.1 Protect the Local Executable.....	49
5.1.1 Fundamental protection to software	49
5.1.1.1 Import Table Protection	49
5.1.1.2 Resources encryption.....	49
5.1.1.3 Additional data extension	50
5.1.1.4 Compression	50
5.1.1.5 Memory Check:.....	52
5.1.2 Protect the critical Functions of software with following technology	53
5.1.2.1 Code obfuscation	53
5.1.2.2 Code Virtualization.....	56
5.1.2.3 Code encryption (Native)	57
5.1.3 Automatically protection to local executable files by using "Command line"	58
5.1.3.1 Generating & Using Map file.....	58
5.1.3.2 Using the SDK label API to mark the critical functions.....	62
5.1.3.3 Generate .ssp configuration file	65
5.1.3.4 Protect software with command line.....	65
5.2 Protect the .Net application.....	71
5.2.1 Protect the .NET application in fundamental.....	71
5.2.1.1 Name Obfuscation	72
5.2.1.2 Compression	73
5.2.1.3 JIT encryption.....	74
5.2.1.4 Remove Strong Name	75
5.2.2 Protect the critical Functions	75
5.2.2.1 Code encryption (.Net).....	76
5.2.2.2 Code Obfuscation.....	78
5.3 Java Program protection:	80
5.3.1 Protection background and introduction.....	80
5.3.2 Protect to Jar archive	81
5.3.2.1 Deployment.....	82
5.3.3 War archive protection:	84

5.4 Unity 3D Program Protection	89
5.4.1 Introduction	89
5.4.2 Protection Mechanism.....	89
5.4.3 Windows, Linux, macOS platform protection	90
5.4.3.1 Protect with Virbox Protector GUI	90
5.4.3.2 Using Command Line to protect the Unity3D program	92
5.4.4 Unity3D android application	93
5.4.4.1 Protect android application with GUI	93
5.4.4.2 Use command line to protect Unity3D apk.....	96
5.4.5 Unity3D program call Net dll plugin.....	96
5.4.6 Protection Comparison	99
5.5 Protect Android application	102
5.5.1 Normal apk application	102
5.6 Protect the python based application.....	102
6 Note	105
6.1 Known Issues.....	106
7 FAQ.....	107
7.1 What is the difference between the soft license edition and dongle edition?.....	107
7.2 What is the difference between the trial edition and standard edition?	107

1 Overview

1.1 Virbox Protector Introduction

Virbox Protector, is the latest Protector and wrap tool to software developer to protect their software copyright and IP which integrated with multi encryption and protection technology: Virtualization, Obfuscation, Smart compression, Code encryption, Data and resource protection, Detecting Hardware breakpoint, Detecting Memory breakpoint, Memory Integrity Check, etc. It is the powerful protector for software developer to protect their software and critical code, algorithm without additional coding, with easy to use and effortless feature.

Virbox Protector is suitable for the following scenarios and software developers:

1. Software developer has established the third party license system or self-developed license system; with Virbox Protector, Developer may enhance the security level of software and integrated with existed license system;
2. The software program needs to be protected and distributed to software users without licensing to software user. Developer just need use Virbox Protector to protect the software and distribute to targeted software user.

3. What is the difference between Virbox Protector LM and Virbox Protector?

Virbox Protector LM, a highly secured, easy to use and without code effort protection wrap tool, is one of critical component in Virbox LM solution, software developer use Virbox Protector LM to protect software and use Virbox LM (Virbox Developer Center) or Virbox Developer Utility to issue the license to the protected software and distribute the software and license to authorized software user.

So, software developer may choose either Virbox Protector LM or Virbox Protector to protect software according to the software applied scenario.

4. Virbox Protector will only protect the software program. And it will not have impact to the software execution or lib called.

Note: Following edition of Virbox Protector available for software developers to choose: Trial Edition, Virbox Protector Standard Edition, Virbox Protector .Net Edition, Virbox Protector Unity3D Edition, Virbox Protector Java Edition, Virbox Protector Android Edition, Virbox Protector ARM-Linux Edition.

Software developer may use trial edition to test and evaluate the project first, then to select the corresponding license according to your system and software environment respectively.



You can contact Virbox team to get above edition by following way:

Tel: +86-10-56730936

Fax: +86-10-56730936-8007

Sales: info@senselock.com;

Or visit official site and leave message to us, we will feedback as soon as possible.

<https://lm-global.virbox.com/detail/virboxProtector.html>

1.2 Advanced and Secured Protection Technology

- **Virtualization:** Code Virtualization & Secured VM function available, the native x86 code is converted into Secured Virtual Machine code and executed inside of VM; In combined with Obfuscation technology, it is effective way to defense static/dynamic analysis tools to debug, reverse engineering to your source code;
- **Advanced Obfuscation:** Advanced obfuscation functions supported to protect code, critical algorithm etc.
- **Smart Compression:** High efficiency Compression tools to developer with high performance, powerful shield to against hacker tools and effectively to prevents de-compilation of .NET, PE programs; effective to defense the crack tools and also keep small size of the program after protection.
- **Code Encryption:** Encrypt the function of your program and only the function is executed the function would be decrypted, with the SMC (Self-Modifying Code) technology.
- **Import Table Protection:** Hide the import table of the original program to protect the functions called by external program. In this way, to against the reverse engineering analysis and prevent the unpacking of the program.
- **Multi Encryption Scheme** to the selected functions, coding to be protected.

1.3 The program supported to be protected

The Operation System supported

- Windows: Windows 7 and above version
- Linux: CentOS, Ubuntu, Debian-9.4.0
- Mac: OX 10.4 and above version
- Android System (Protect Unity3D apk, .so library), 4.0 and above version supported
- ARM Linux (V7/V8 architecture)

The program/framework language

C, C++, .NET, .Net Core3, Java, Unity 3D, Unreal Engine 4, Delphi XE7 or above version, PB, BCB, C#, VB6.0, Python, Lua, Perl, R, Ruby, PHP, etc.

The plugin and framework supported

AutoCAD ARX, Revit

Development Tool supported

MATLAB, LabView

Executable file

32 bit/64 bit executable file and dynamic link library (DLL)

Elf and .so library

Resources protection supported

The software resources of the program developed based on Unity3D, UE4 engine, can be encrypted and prevent from being extracted illegally.

Below table is the type of the file supported by Virbox Protector:

File Type	System Supported	Architecture	Programming language
.NET	Windows	x86, x64	VB, C#, etc
.NET Core3	Windows, Linux, macOS	x86, x64	C#, VB.net
PE	Windows	x86, x64	C/C++, Delphi, PB、BCB, etc
Unity3D	Windows, Linux, macOS, Android	x86, x64, ARM32	C#, etc
ELF	Linux, Android	x86, x64, ARM32、ARM64	C/C++, etc
Mach-O	macOS	x64	C/C++, Objective-C, Swift
java	windows	x86, x64	Java

Software Protection & Evaluation Process:

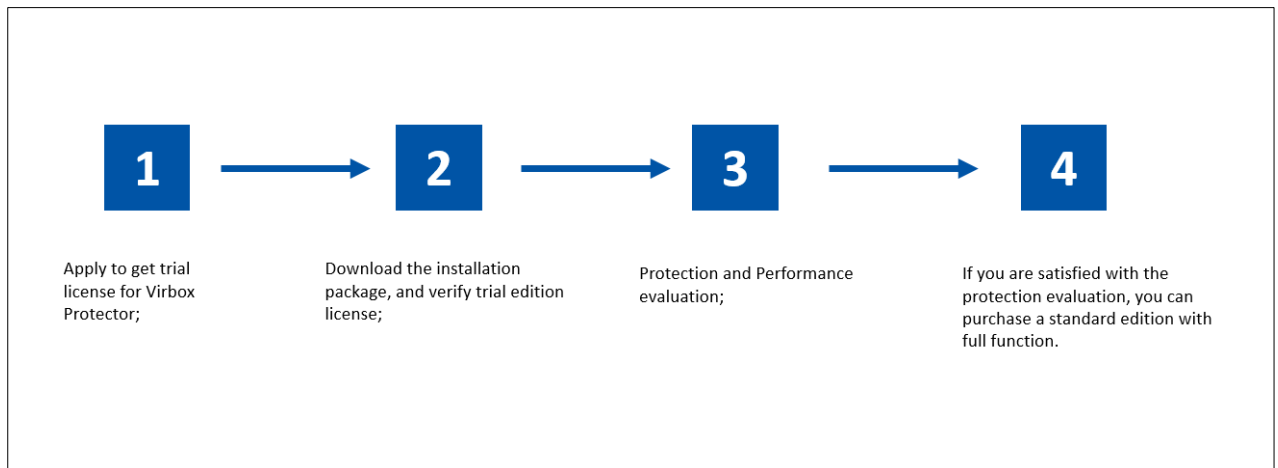


Figure 1-1

1. Apply to get trial license for Virbox Protector;
2. Download the installation package, and verify the license for trial edition in your computer;
3. **Protection and Performance evaluation:** protect your software or data resource with Virbox Protector to evaluate the protection scheme and performance according to the instruction of User Manual;
4. If you are satisfied with the protection evaluation, you can purchase a standard edition with full function.

The Limitation to trial edition:

Trial license for Virbox Protector will be valid within **30 days or 100 times usage**, the software protected by trial edition would be expired in **7** days, no limitation by standard edition;

2 Installation of Virbox Protector

2.1 Installation

For different software you want to encrypt/protect, please select the right license from Virbox.

Windows Edition, .Net Edition, Unity3D Edition, Java Edition, ARM-Linux Edition, Android Edition available.

Download the corresponding installation package and install on your computer.

After the installation of the Virbox Protector, you will get two software installed in your computer: **Virbox Protector & Virbox User License Tool**. Virbox User License Tool is the tool to verify the Virbox Protector License. You need to activate your Virbox Protector license and verify the license via Virbox User License Tool before start to protect your software/program.



Figure 2-1

The following chart shows the Virbox Protector installation path:

```
└─bin
| └─virboxprotector.exe
| └─virboxprotector_con.exe
|   └─dsprotector_con.exe
└─example
|   └─plugin
|       └─demo
|       └─src
|       └─sdk
└─help
└─plugin
└─anti
```

2.2 License mode of Virbox Protector

Virbox Protector supports following license mode to software developer to choose when they apply trial and evaluate Virbox Protector performance or purchase Virbox Protector later:

Trial License: Cloud based license; which is the easiest way for software developer to apply and get the trial license quickly. The Virbox Protector's trial license can be get by providing your email, we will issue the trial license into your email, use your email and password to login the **Virbox User License Tool** to start the trial.

Trial license for Virbox Protector will be valid within **30 days or 100 times usage**, the software protected by trial edition would be expired in **7** days, no limitation by standard edition;

Free trial license apply:

<https://lm-global.virbox.com/detail/virboxProtector.html>

For official Virbox Protector user (software developer), they can select either "Soft license" or "dongle based license" according to their requirement;

Soft license: Support for account based license both in online/offline environment;

Dongle License: Use Virbox EL5 (hardware dongle, order separately) to be the License container of Virbox Protector. Developer can use Virbox Protector at designated computer which have plugged in EL5 Dongle;

All of the license modes support subscription and perpetual license for Virbox Protector.

2.2.1 License Verification with cloud license (For Trial User)

2.2.1.1 Verify license by Virbox User License Tool

Open Virbox User License Tool, sign in your account with your email (the email you provided to Virbox) to verify the Virbox Protector License, then you can open and use Virbox Protector to start the testing and evaluation

Note: After we issued the license into your email account, a password will be sent to your email. You can sign in the Virbox User License Tool with your email account. The password is the password you received in your email box.

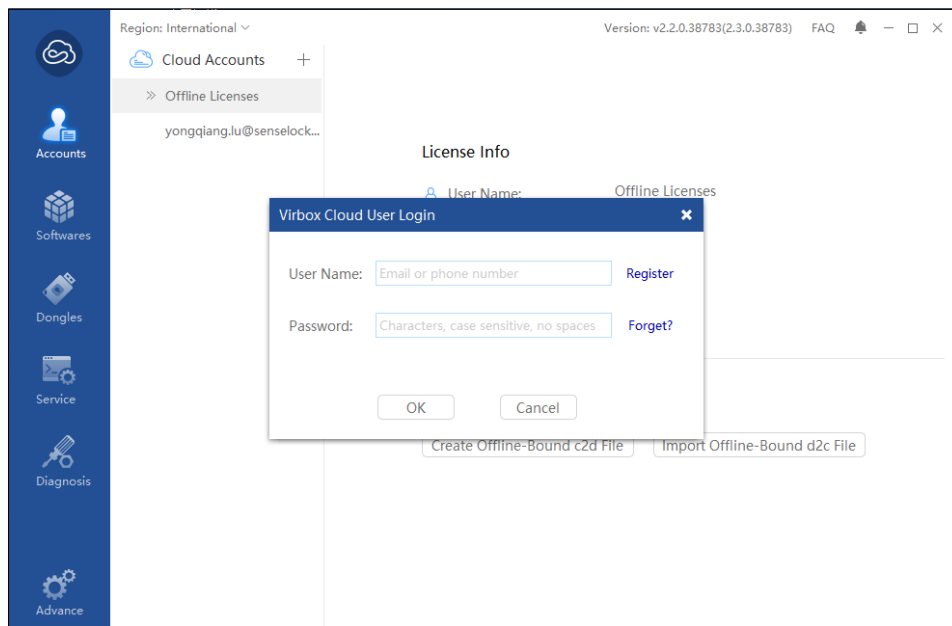


Figure 2-2

After sign in the account, you can check the detail information of the license here showing in the picture:

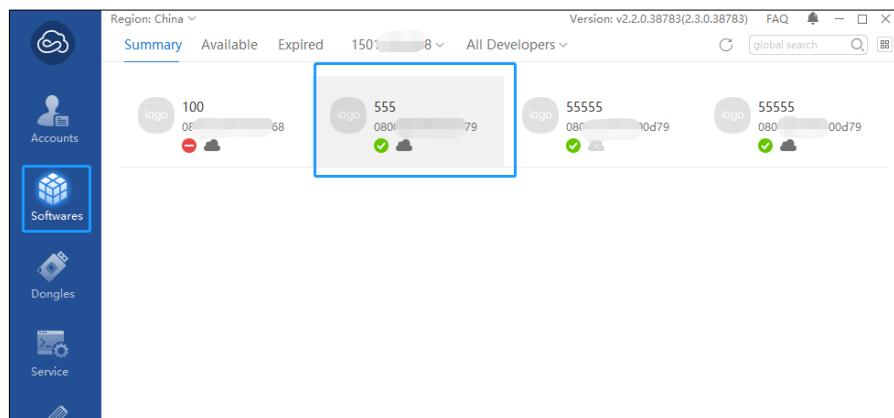


Figure 2-3

2.2.1.2 Sign in by the Virbox Protector Interface

You can also sign in from the Virbox Protector interface:

Sign in the authorized account:

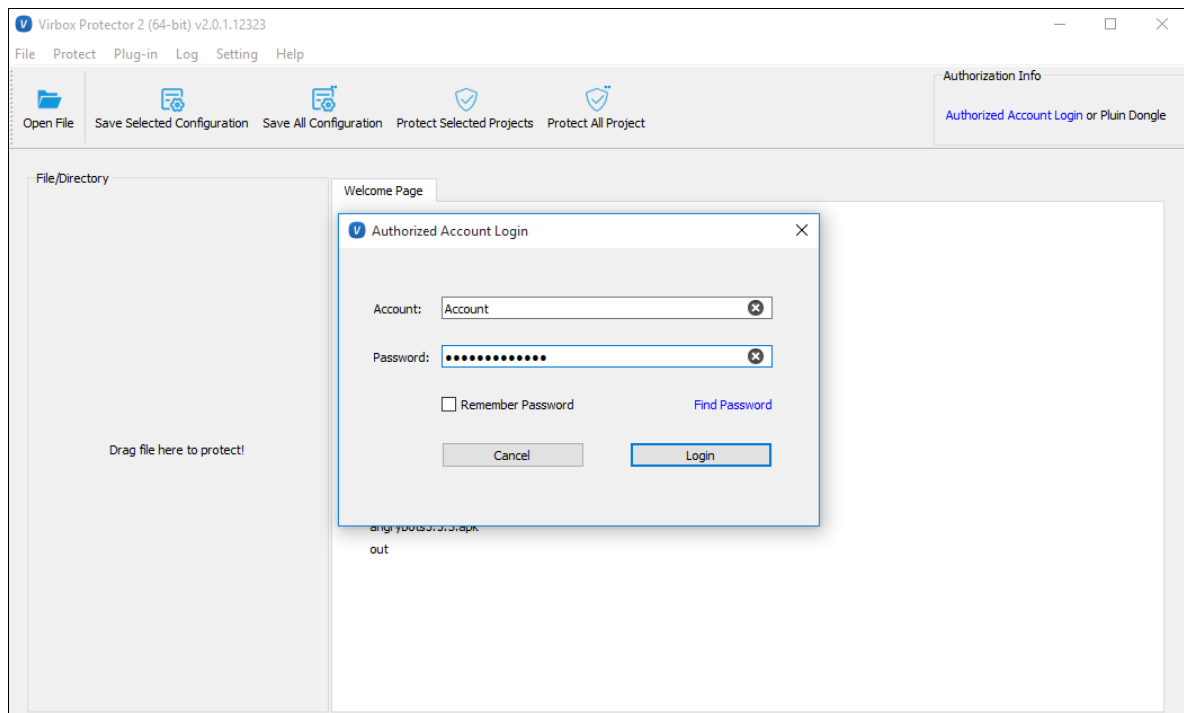


Figure 2-4

You can check the license detailed information by clicking here:

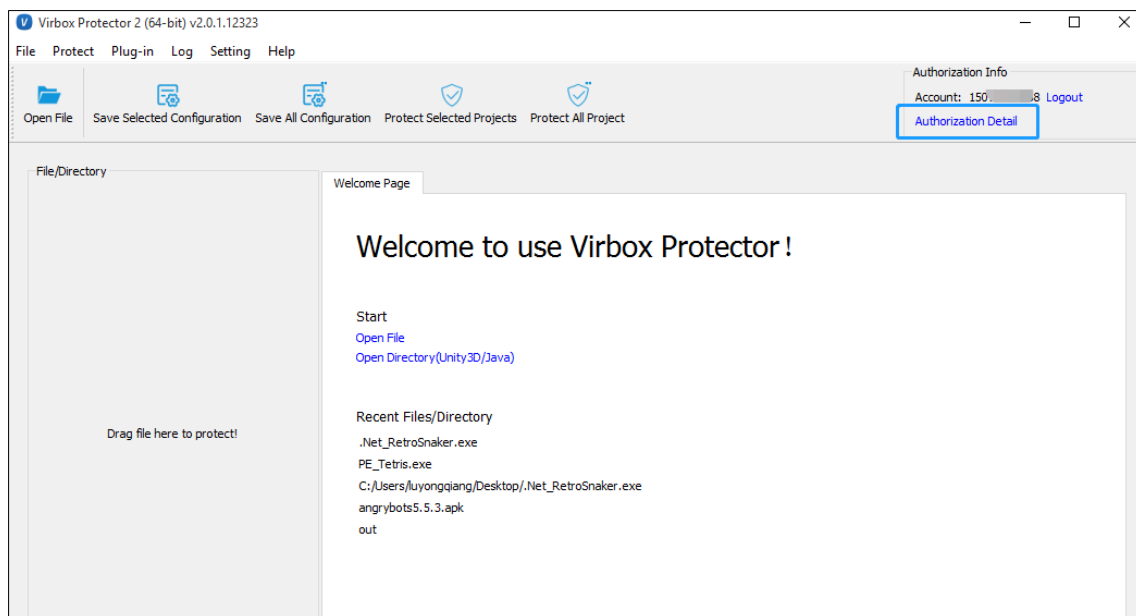


Figure 2-5

You can logout by clicking the “logout” button.

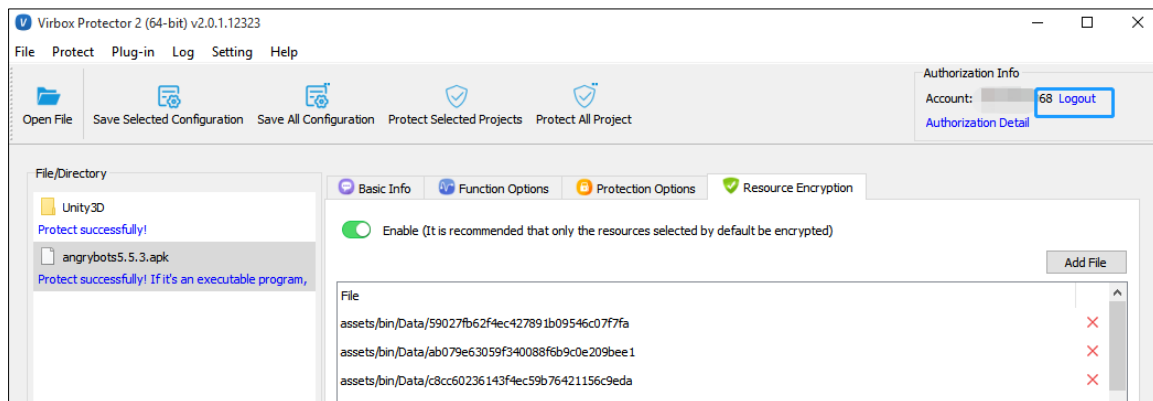


Figure 2-6

2.2.2 License Verification with soft license

2.2.2.1 Use Virbox Protector in online environment

When you use the Virbox Protector in online environment, you can sign in the account that have already issued license. After you start Virbox Protector, the software license will bind to your hardware machine automatically.

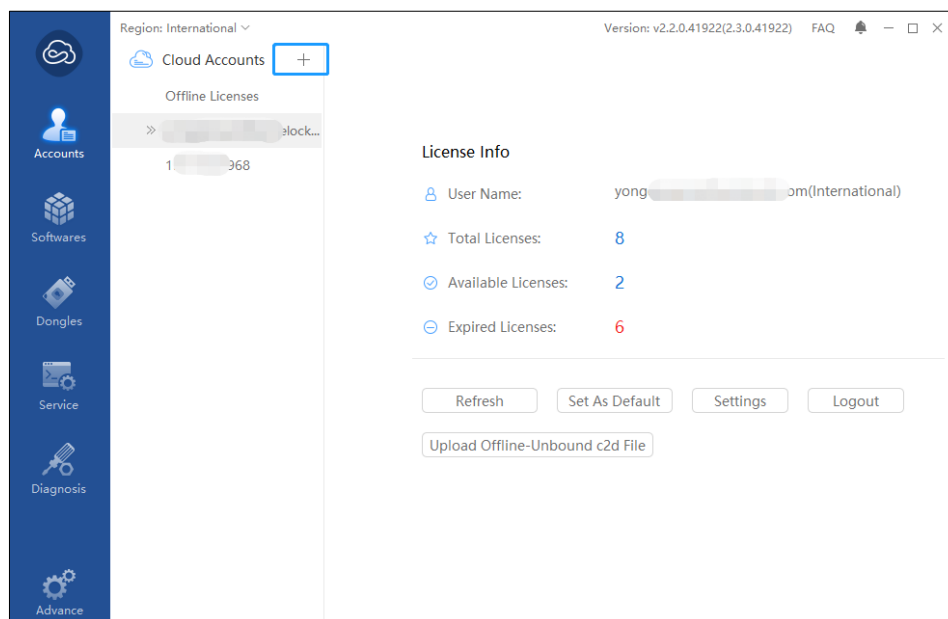


Figure 2-7

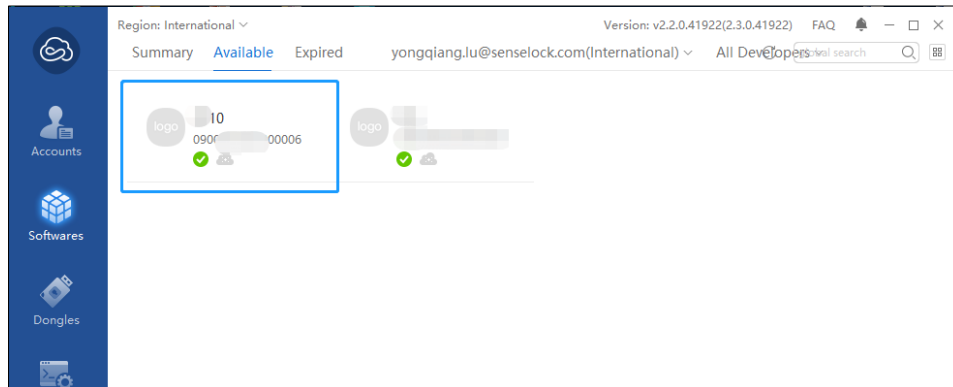


Figure 2-8

2.2.2.2 Use Virbox Protector in offline environment

If the Virbox Protector is used in offline environment, you need to use the following step to activate the license of your offline machine with a computer that can connect to internet (Online computer). Both computer (Online and offline computer) need to install Virbox User License Tool.

- **Generate c2d file on the Offline computer**

Open Virbox User License Tool, click “**Accounts**”,

Click “Offline”,

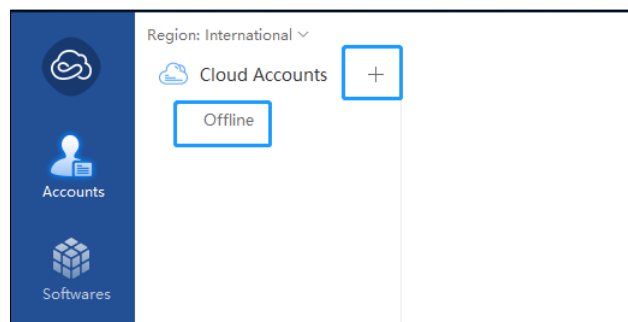


Figure 2-9

Generate offline bind c2d file, and save the .c2d file.

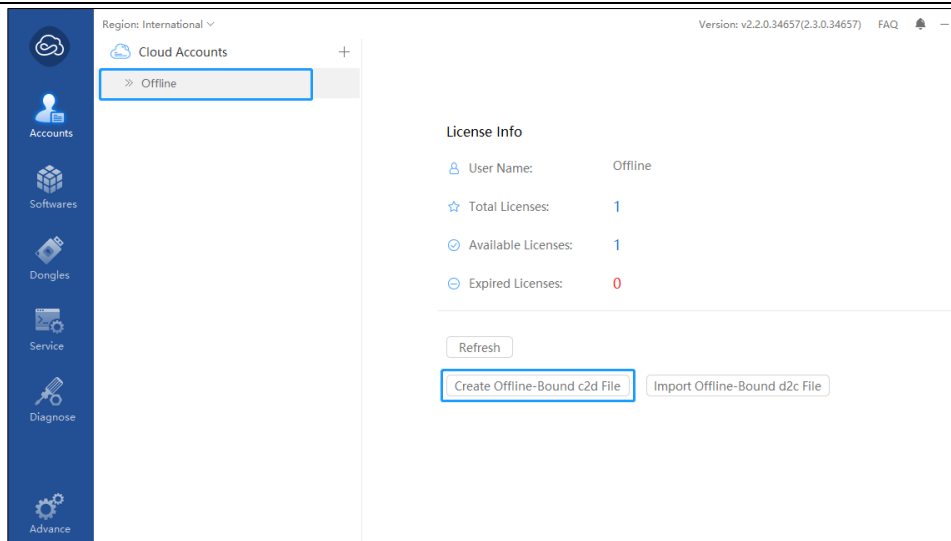


Figure 2-10

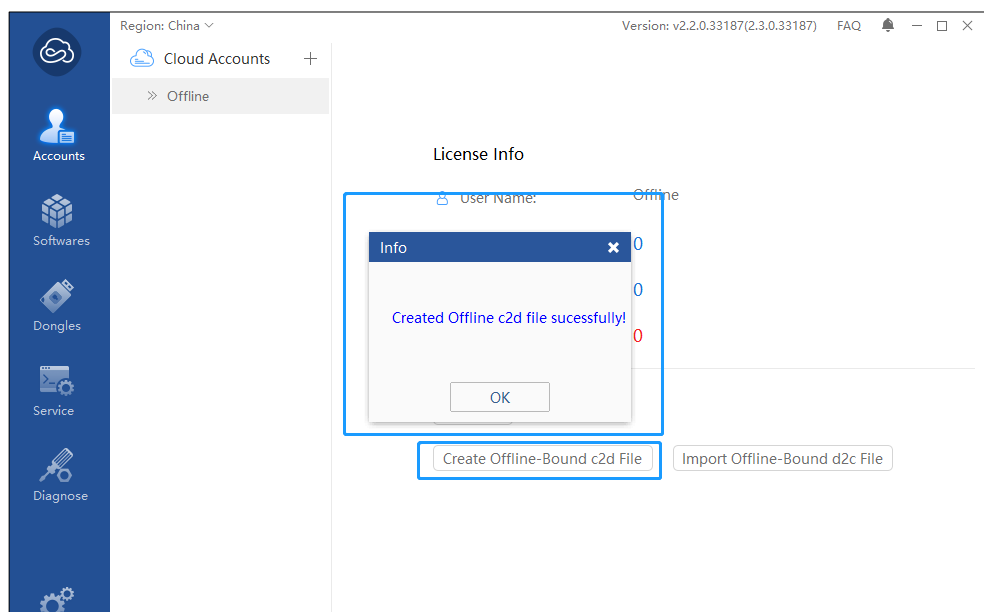


Figure 2-11

After you have created c2d file successfully. You need to copy this c2d file to the online computer.

- **Create d2c file on the computer Online**

Also need open Virbox User license Tool on online computer.
Click “+” to login your account that have already have license.

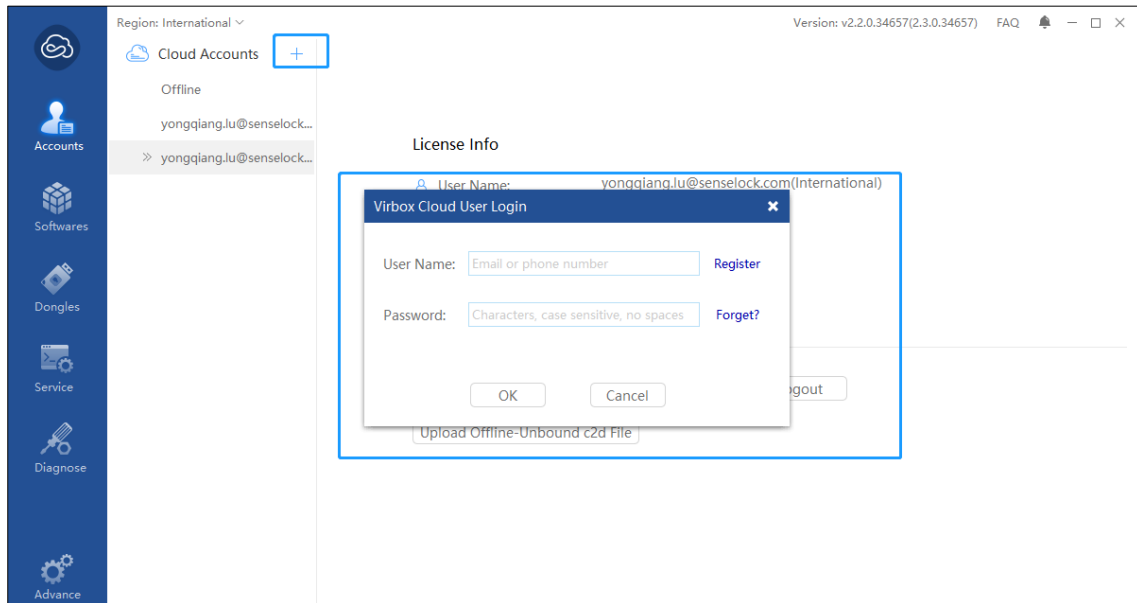


Figure 2-12

Click **“Software”**,

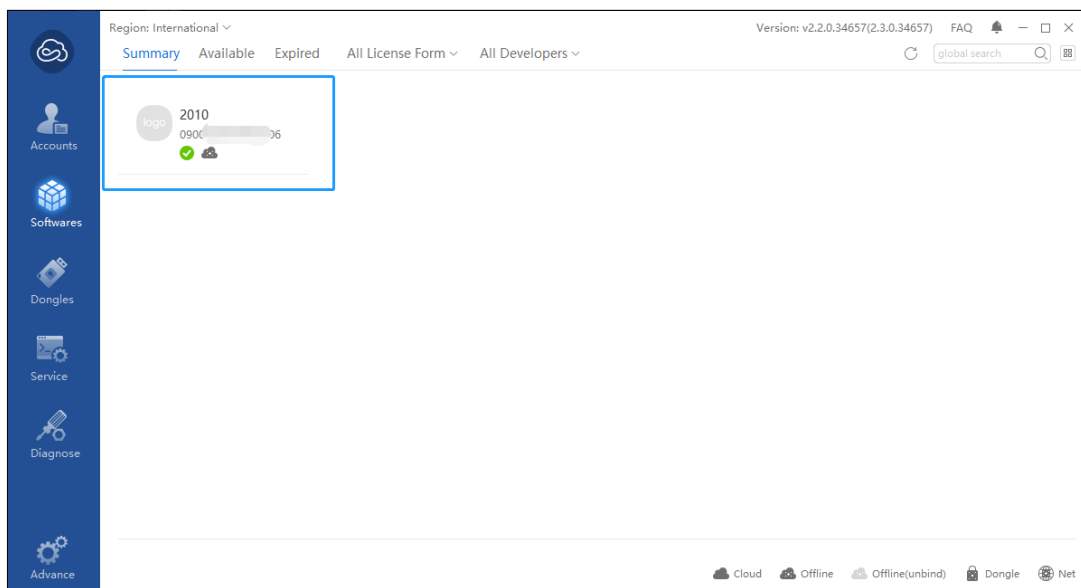


Figure 2-13

Double click the license, the detail information of the license will show:

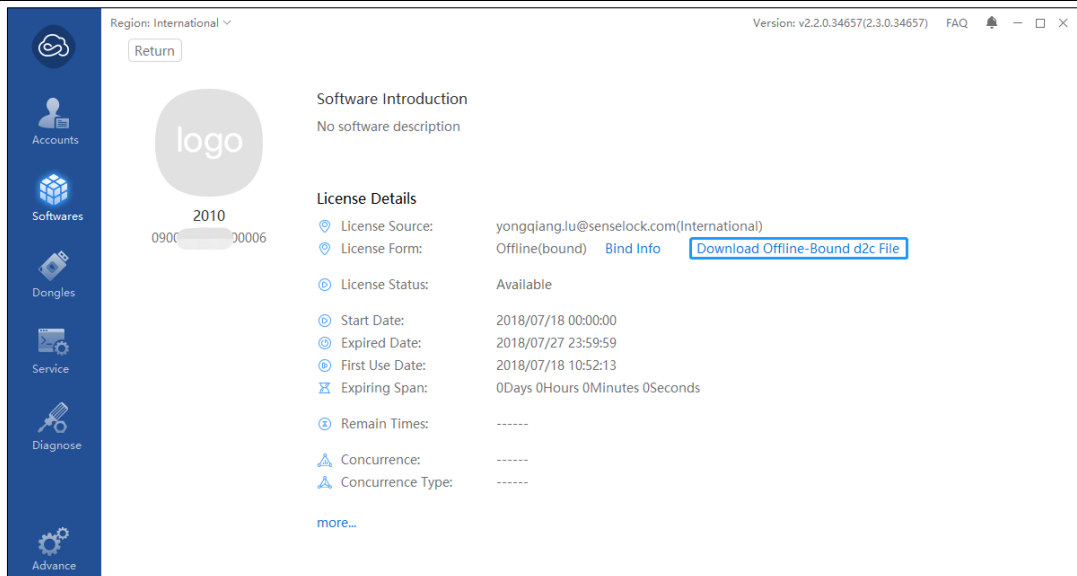


Figure 2-14

Click **“Download Offline Bound d2c file”**

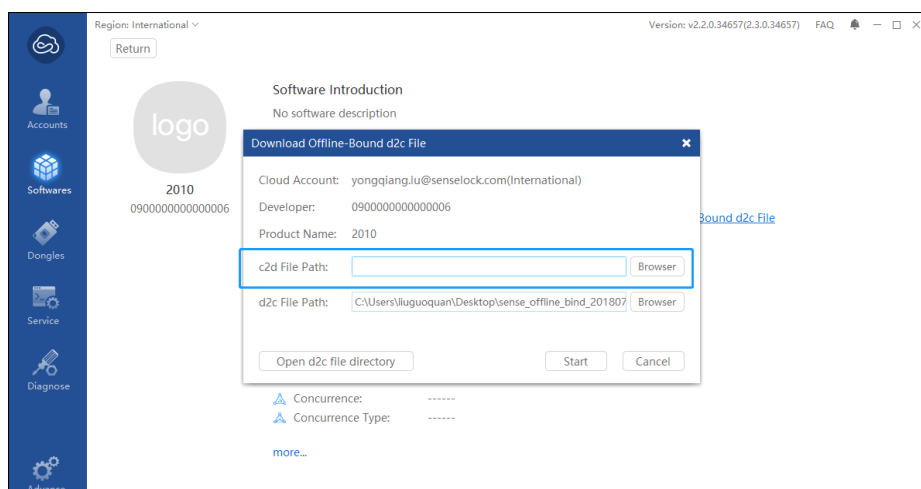


Figure 2-15

To generate a D2C file, you need to import the c2d file you generated from the offline computer in last step, Click **“Start”**,

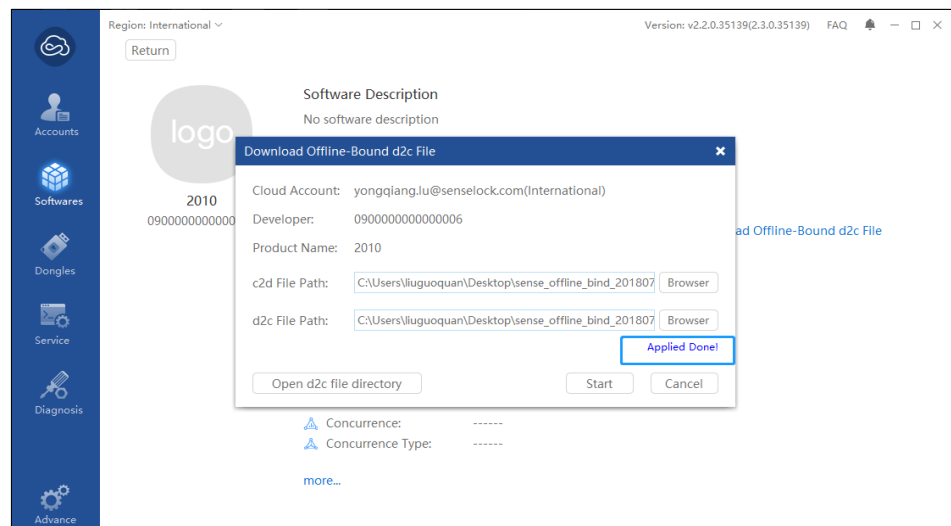


Figure 2-16

You can select the path to generate d2c file, here I put it to desktop. If the file is generated successfully, it will show “**Applied Done**”, such as in the picture above showed.

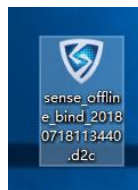


Figure 2-17

This is the d2c file generated.

Note:

1. Click “**open d2c file directory**”, also will show the path of the generated file.
2. *The valid time of this d2c package is 24 hours, please complete binding process in time.*

▪ Verify d2c file on the Offline computer

Now we need to copy d2c file from online computer to the computer offline and complete license verification.

Copy the **d2c** file generated from the computer (Online computer).

Import it in to the offline computer.

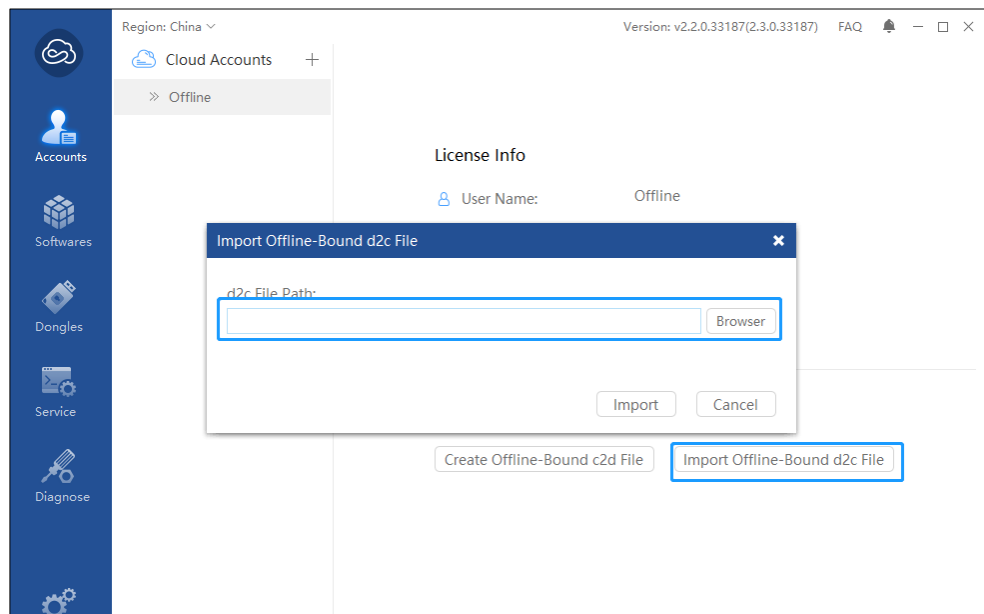


Figure 2-18

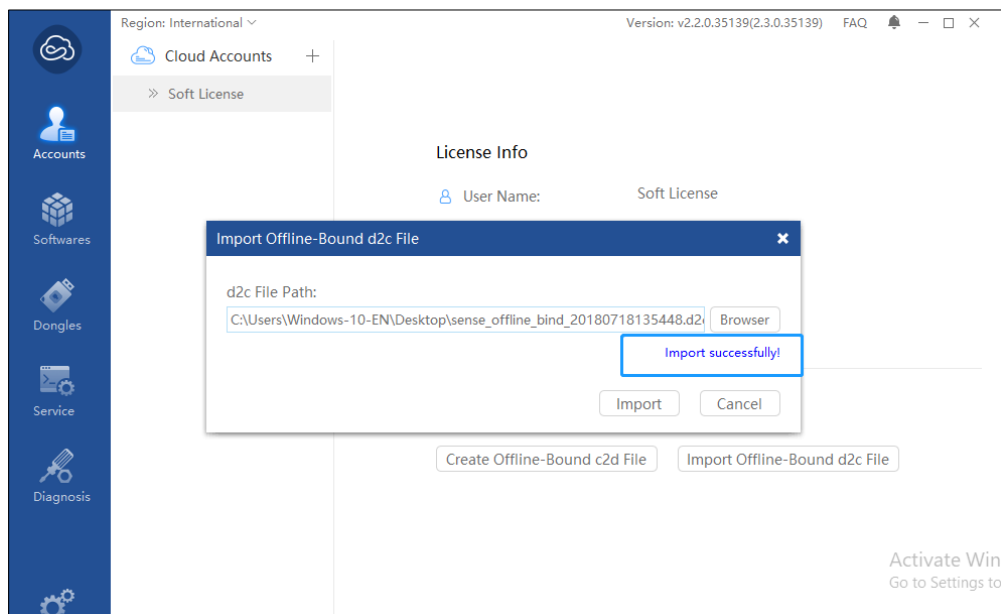


Figure 2-19

It will popup “**Import successfully**” message after you imported the file successfully.

Then you have activated the license on the offline computer successfully, and you can use it to encrypt your program now.

2.2.3 License Verification with EI5 dongle (For official user use dongle license)

If you purchased the Virbox Protector with a Virbox EI5 dongle, after installation you need to insert the dongle

on your PC for license verification. Then you can use Virbox User License Tool to check the license you have subscribed. As the figure shown below:

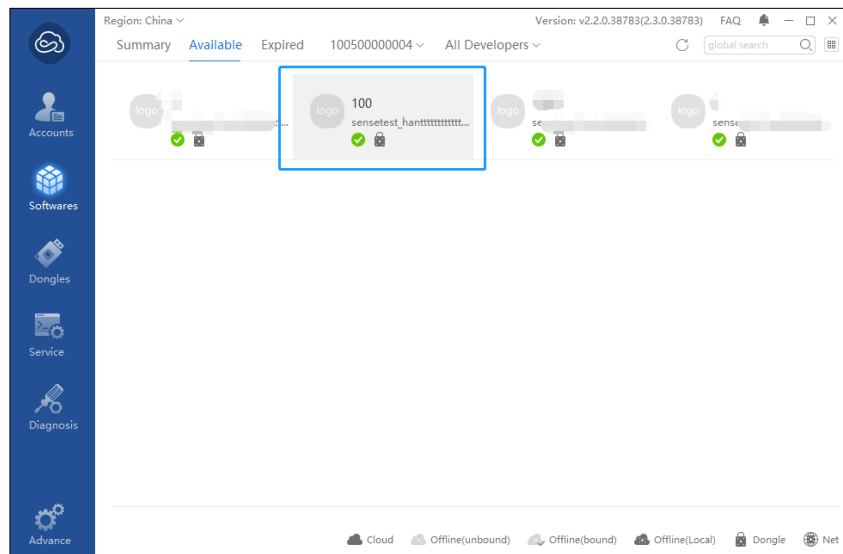


Figure 2-20

You can double click that dongle icon for license detail information.

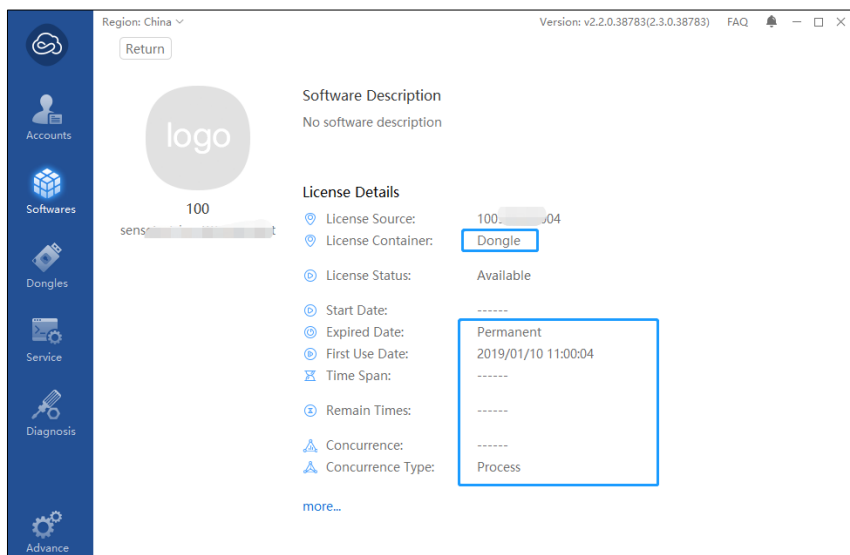


Figure 2-21

3 Protection Function Introduction

3.1 Main Menu of Virbox Protector

The main menu shown as below: includes 3 areas:

Menu Bar: consist of: *File/Protect/Plug-in/Log/Setting/Help* functions;

Tool Bar: *Open File/ Save Selected Configuration/Save All Configuration/Protect Selected Projects/Protect All Projects*;

File/Directory Panel and Protection Panel

These functions and options in the menu, includes the Tab and Panels will be introduced and explained in this chapter.

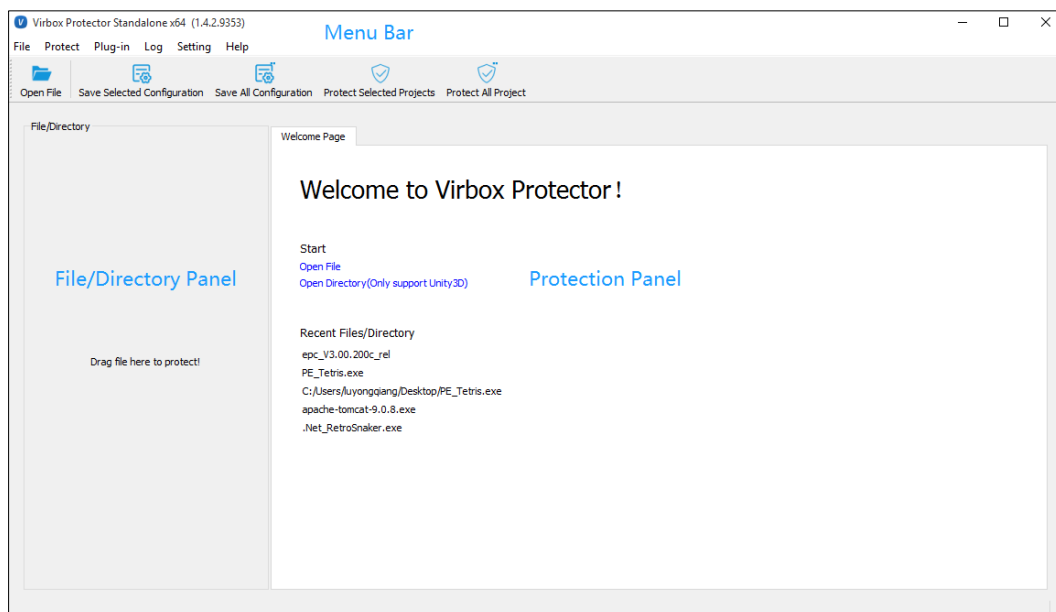


Figure 3-1

3.2 Menu Bar

3.2.1 File

Open File: Click "*Open File*", you can select .Net, PE, ELF, MachO, Arm Linux and Android .so lib file. And load the selected files into the left panel, the File/Directory Panel.

Note: If the xxx.map file located in the same directory of the program file being protected, these xxx.map file will be loaded with the program file automatically. And the name of the functions will also be loaded and listed in the File Panel. The map file generated by VS, VC, BCB, Delphi compiler is supported at present.

Open Directory: Here you can open a whole directory, to open Unity3D directory only.

Recent/Batch Projects: Here you can reopen the recent protection project quickly, or batch file protection project. Up to 5 recent projects can be recorded.

The recent protected program would be loaded and listed into the File Panel.

If you want to save the project setting and path of multiple file, you can save those configuration into a project file **xxx.vbpsln** by clicking **"Save Batch/project"**. Then you can reload this project file for next protection.

Save Batch Projects: You can use this function to save all of path of the file, but the configuration would not be saved. If you have changed the configuration and want to save them, you need to click "save the selected configuration" or "save all of the configuration".

When you reopen the Virbox Protector, you can drag in xxx.vbpsln to open the project, the saved file and configuration would be loaded if the location of the file haven't changed.

Exit:

Close Virbox Protector and exit.

3.2.2 Protect

Parse selected project (File):

Select one or multi file which listed in the File Panel, you can parse these file by clicking **"parse selected project"** button. The file need to be parsed correctly before protection.

Parse all project

Parse all of the files in the project, no matter how much you have selected.

The purpose of parsing is to reload the configuration status you saved.

Save selected configuration

Configuration means the function options, protection options, which you selected to the protected file,

You can save the configuration of the Function options, Protection options, Message by clicking "save selected configuration"

Save All configuration

Save all of the protection configuration of the project, no matter how many file you have selected. Corresponding error report or error code will show, if the configuration is not correct and you can't save the configuration.

Protect selected project

You can protect the selected file in the file list by clicking this option. If the configuration is not correct, it will remind you corresponding error report or error code.

Protect All project:

No matter how much file you selected in the file list, you can protect all the file by clicking this option.

These function are also available in tool bar, you can also use those function from the tool bar.

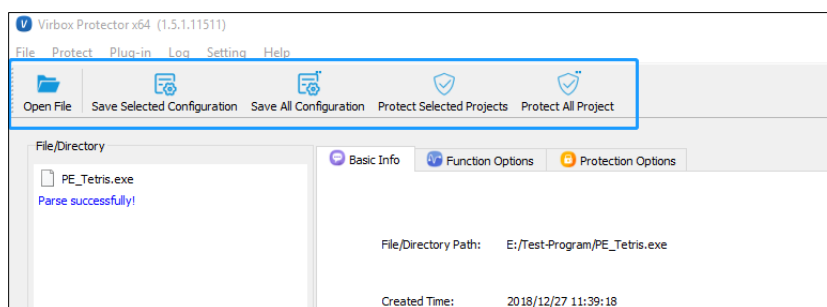


Figure 3-2

3.2.3 Plug-in

Open "DSProtector", which is the plugin tool for data resources protection, such as jar archive, .py file, pyc file etc.

"DSProtector"(Hereinafter referred to as DS Protector) is the plugin unit to protect the data resources provided by Virbox, software developer may use DS Protector to protect data file and encrypt related data resources together with protected software program.

Please noted that DSProtector does not support the data resources protection which from Linux and Mac system currently.

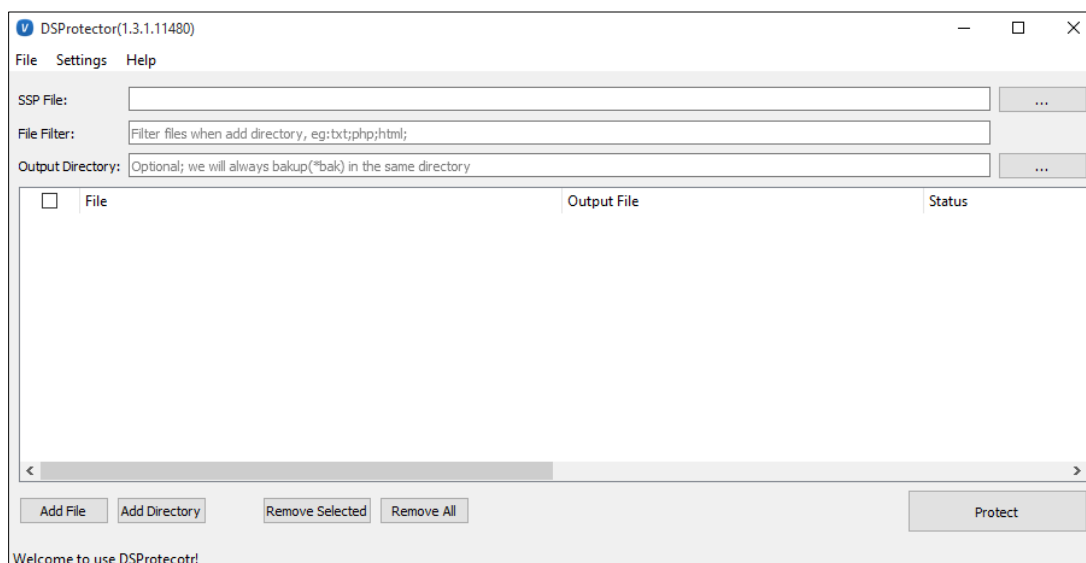


Figure 3-3

3.2.4 Log

Show log dialog

Log dialog will show the log file when you are protecting the software. You can save the log by clicking “save”, to save the log to other directory.

Open local log directory: Open the log directory.

3.2.5 Setting

Language setting:

Both Chinese and English are supported. To change the language of the interface of the software you need to restart the software. You can restart instantly or next time you open the software.

3.2.6 Help

About: It will show you the technical support email and website.

3.3 File Panel and Protection Panel

3.3.1 File Panel

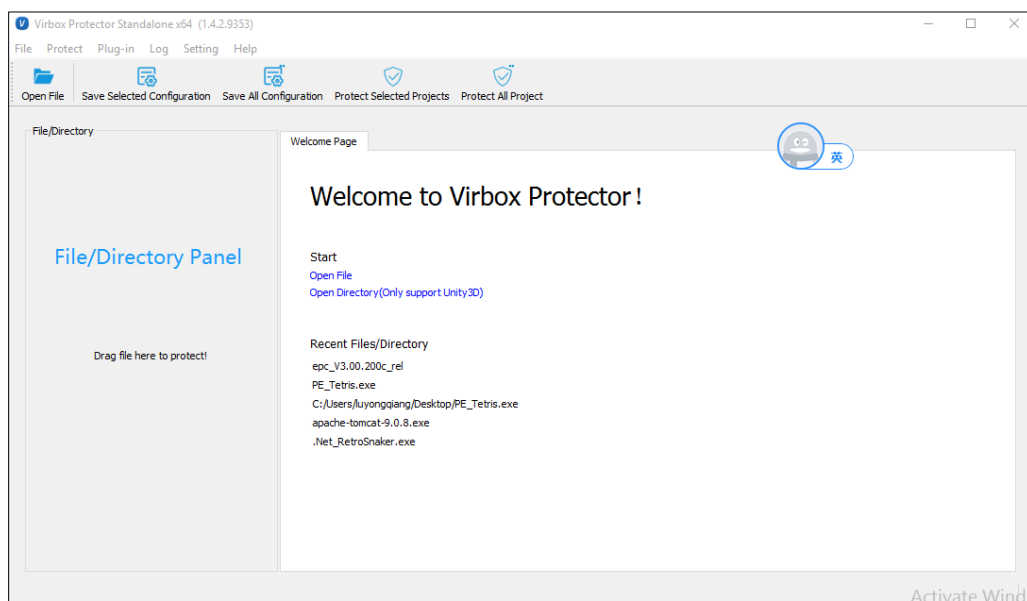


Figure 3-4

After you drag the software to be protected in to the file panel, the software basic information will show in the

You can select one or more software and right click the software to select the corresponding function.

- **Parse (software or file)**
- **Save configuration**
- **Protect software**
- **Show containing folder**
- **Set output directory for protection**
- **Copy the protection option of this file to other selected files:**

Select 2 more software, right click software and choose “copy the protection option of this file to other selected files”, and use this setting to other software, the other software will have the same configuration.
- **Close project:**

Right click the selected software, and choose close, you can close the program with saving the current configuration and also can exit without saving, or cancel the operation.

3.3.2 Protection Panel

- Basic info
- Function Option
- Protect Option
- Resources Encryption

3.3.2.1 Basic Info

Basic info will show you the basic information of the loaded software, File/directory path, file creation time, Last configuration Modified time, Last Accessed Time, Application Type (PE or .Net, etc).

3.3.2.2 Function Options (function level protection)

Virbox Protector supports to protect the software application to the functions level and provides several protection mode to developer select to protect the critical functions.

Add the Functions to be protected

Function Option page lists all the functions in your application, you can select and protect the critical functions in your application or program in this page, you can select **No Protected**, **Virtualization**, **Code Obfuscation**, and **Code encryption** mode to protect the selected functions.

When you click the function in the software, the protection mode, function name, function address and assemble code will show in this page. And the quantity of total functions, total protected functions and the quantity of every function protection type will be shown in this page.

As shown the figure below:

This tag will list all of the function module which have been parsed (There are little difference between managed code and un-managed code), you can select the corresponding protection option.

- Managed code: The function name is “Name Space + Class Name+ Function Name”
- Un-managed code: The function name is the “va” value of the function.

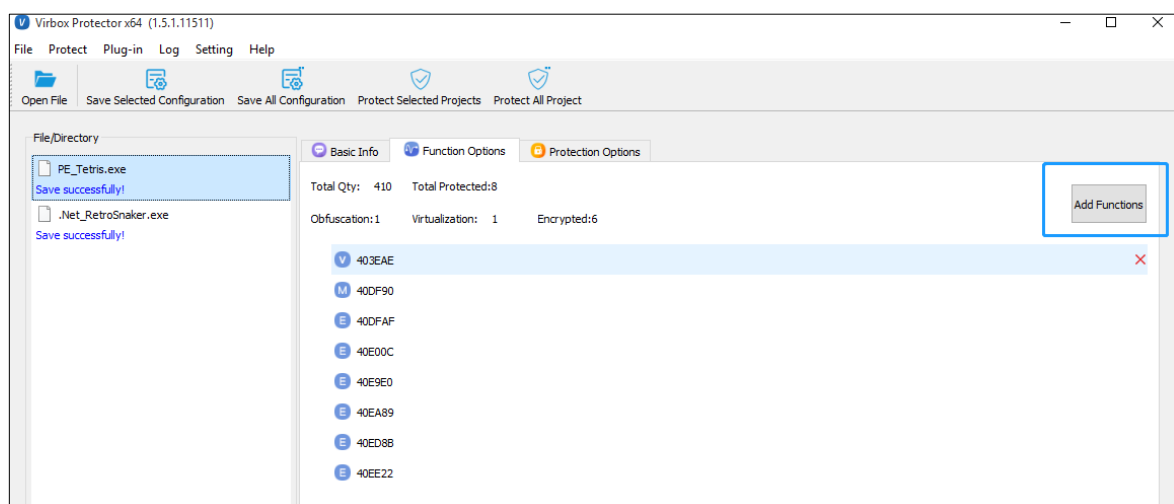


Figure 3-5

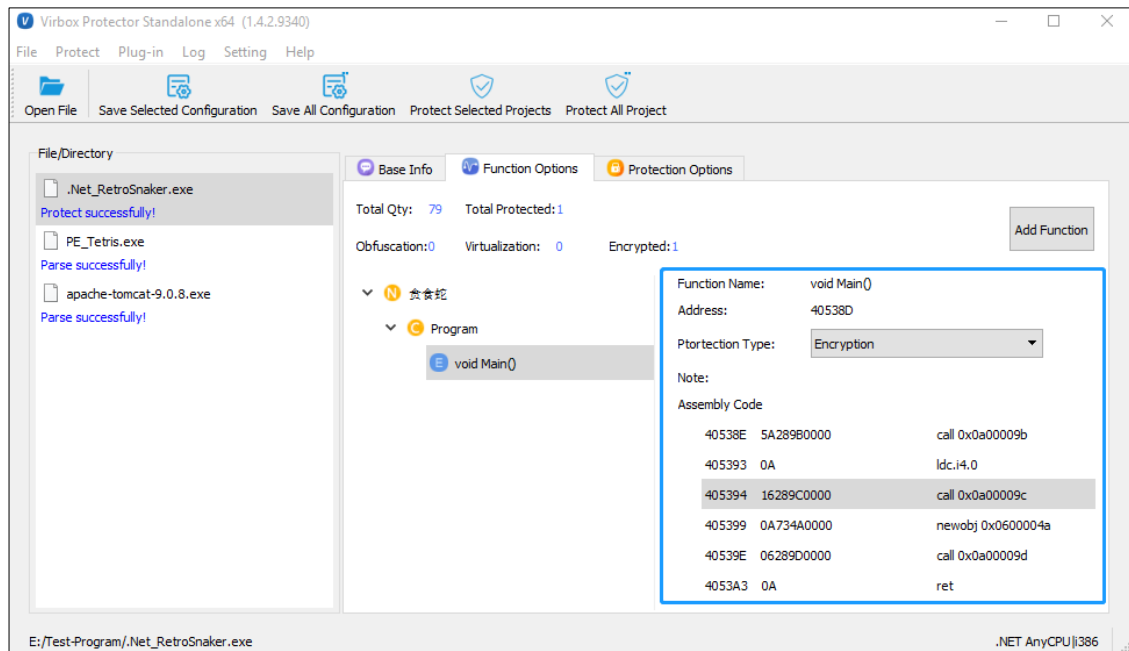


Figure 3-6

Following steps will describe how to set and select options for above setting:

[Note: If xxx.map file existed in the same folder of software be protected, Virbox Protector will load this map file automatically and list functions in the main menu, currently the map file support be protected includes the map file created by VS, VC, BCB, Delphi compilers.]

Note: Usually, software developer need to balance the software execution performance and protection level before software protection. Be careful to select and protect these frequently called functions, since it will decrease software execution performance after protection/encryption.

Click "Add Function" (See picture attached), you can enter the "Add Function" window.

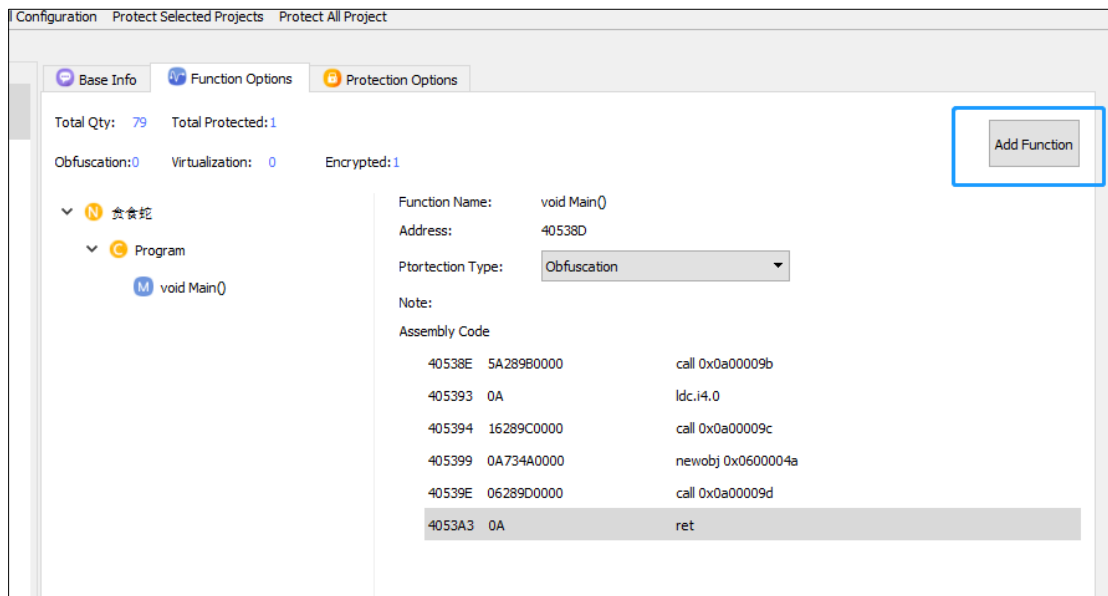


Figure 3-7

The **Virbox Protector** will list all of the functions used in this software in the left panel.

Click "OK" to confirm the protection and the function you selected will show up in the function list:

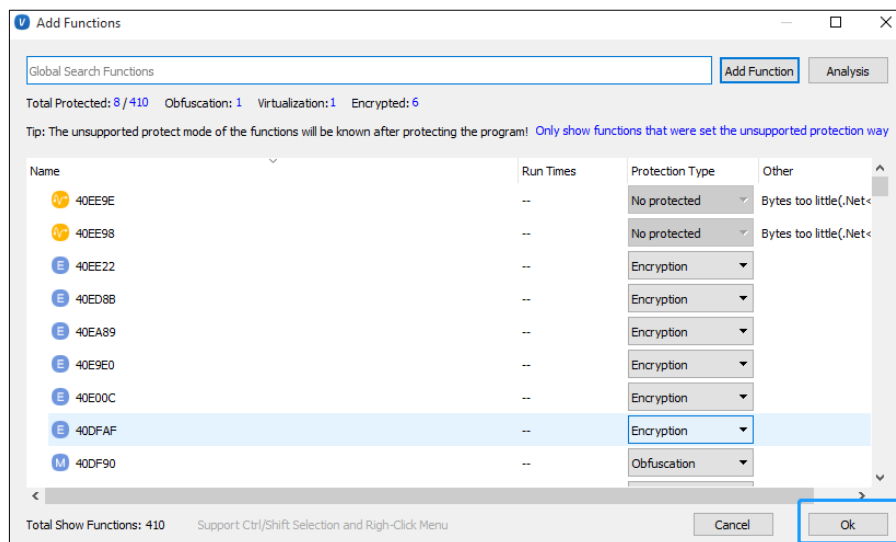


Figure 3-8

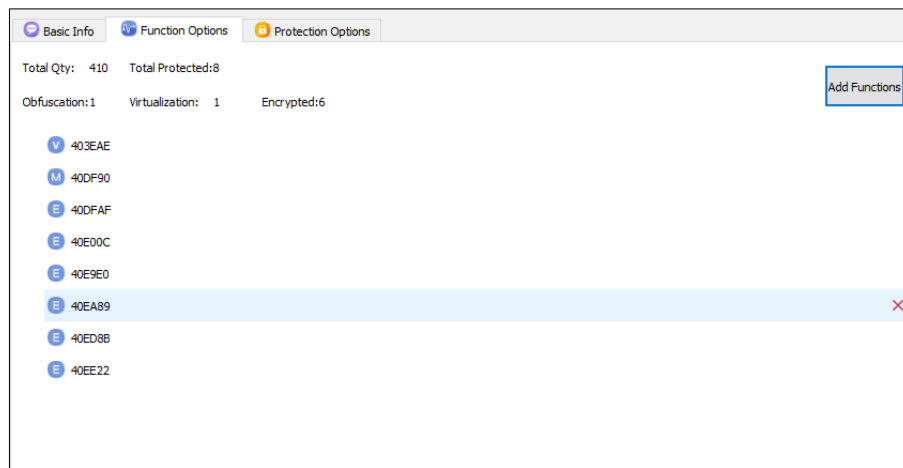


Figure 3-9

Select the function to be protected/encrypted with different protection mode:

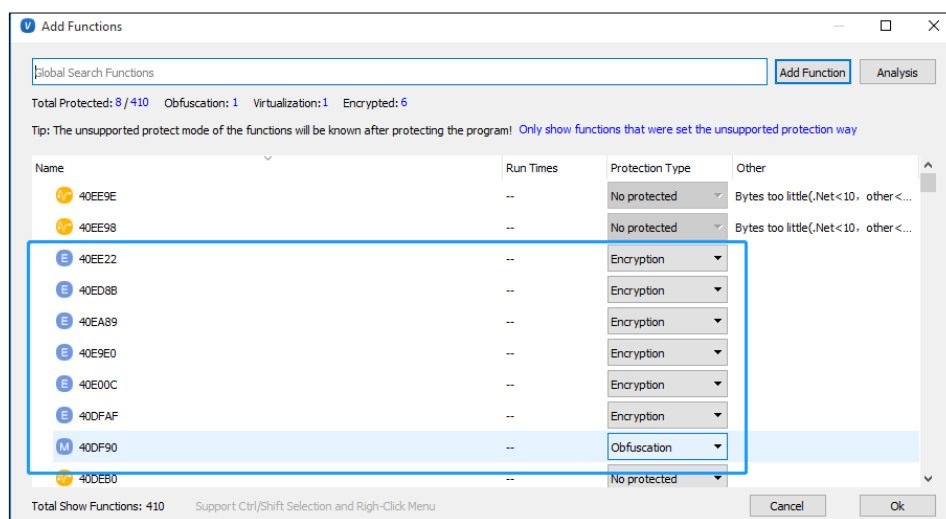
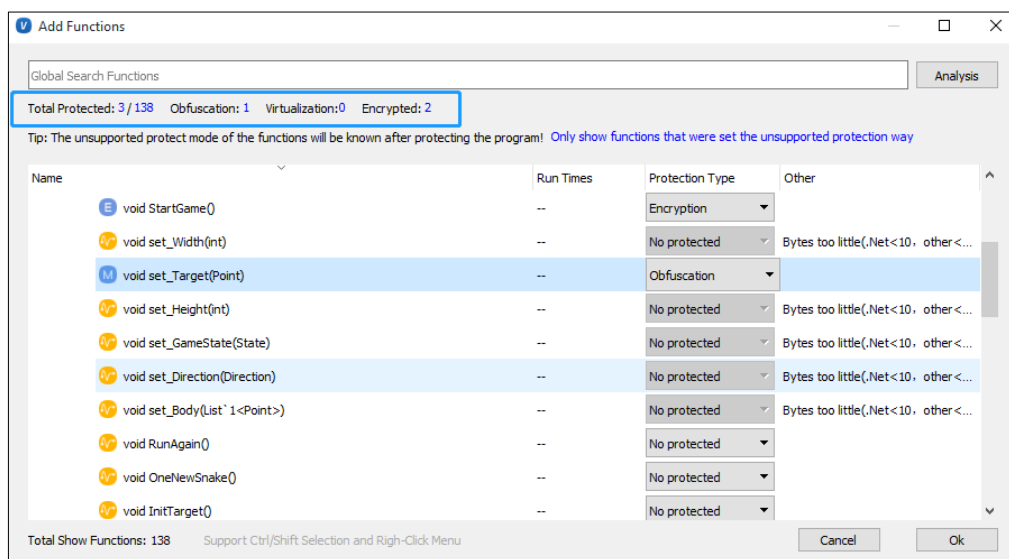


Figure 3-10

Also the total protection option you selected would be counted and show in the interface:



If the protection process failed, Virbox Protector will prompt “Some of protected functions doesn't support the protection mode you selected”, you need to change to other protection mode to protect the function.

Protection Mode:

To protect the specified functions of the software, following functions protection mode can be selected: **No protected**, **Code Obfuscation**, and **Virtualization**, **Code Encryption**.

- ◆ For the functions which is called frequently, select "**No Protection**" option, since if you protect the functions which is called frequently, it will decrease software's running performance when software is executed;

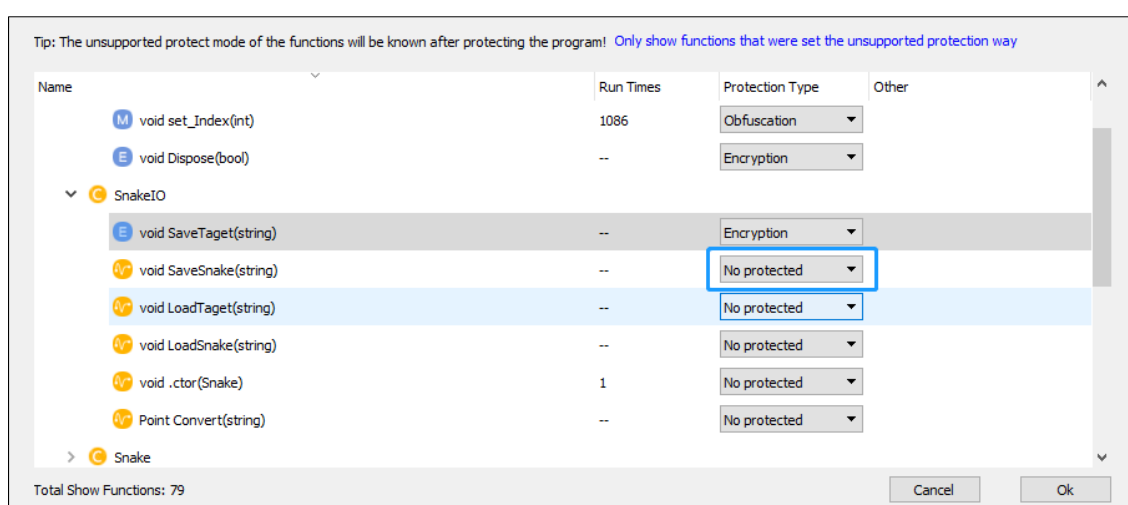


Figure 3-11

- ◆ Select "**Obfuscation**": Virbox Protector will translate the code instruction into a stream of pseudo-code that neither the machine nor the human can identify it. When the pseudo-code

executed, the software will translate and interpret to restore the code into the original code. To let it be executable.

Virbox Protector support the obfuscation for x86/ARM .Net il serial instruction.

Code Protection Mechanism:

Interference the original instruction and prevent the code from being static analyzed.

The Benefit:

Prevent from anti-compiled and make it more difficult to analysis the code.

The Weakness:

Partial Impact to the execution performance.

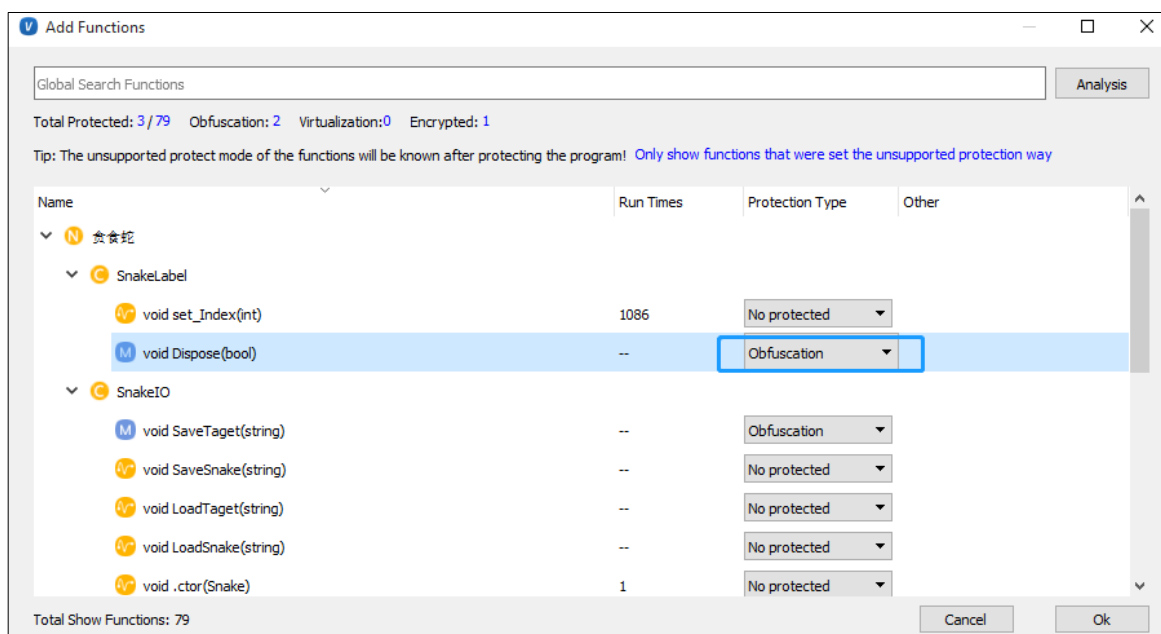


Figure 3-12

- ◆ Select "Virtualization": Virbox Protector will compiles original instructions into virtual instruction and run them in the specified virtual machine. There are certain format requirements and limitation for instructions, and some functions may not be protected;

Protection Mechanism:

Hide the original instruction, prevent the code logic from being analyzed.

The Benefit:

Highly secured protection mode, the original code logic almost can't be get by analysis.

The Weakness:

Performance impact to software execution

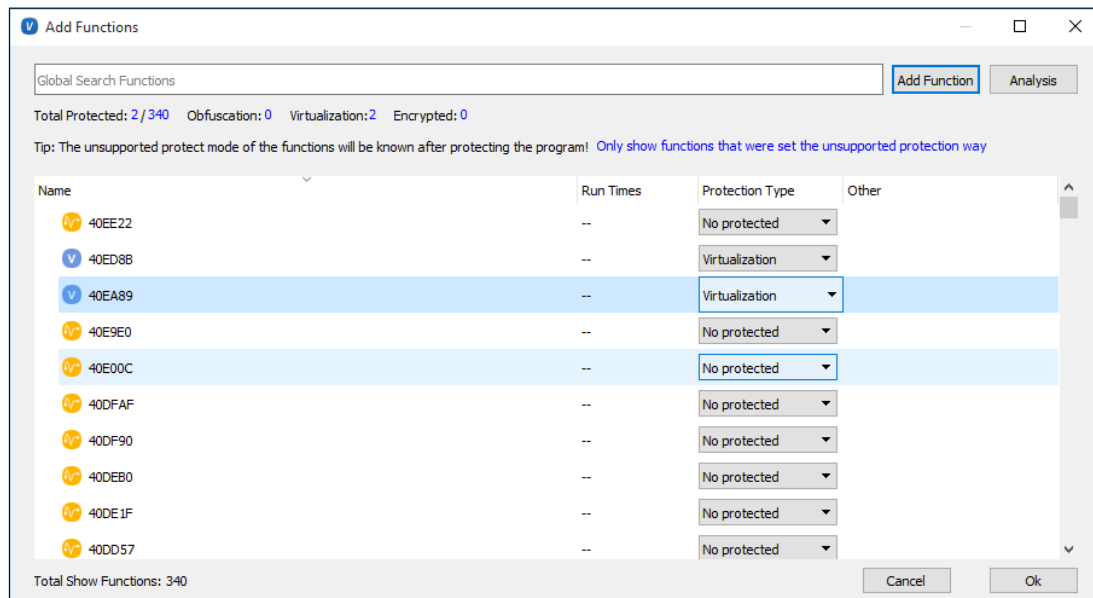


Figure 3-13

◆ Code Encryption:

Code encryption, it encrypt the original function of the program by SMC (Self-Modifying Code) technology and the function will be decrypted only when the program executed.

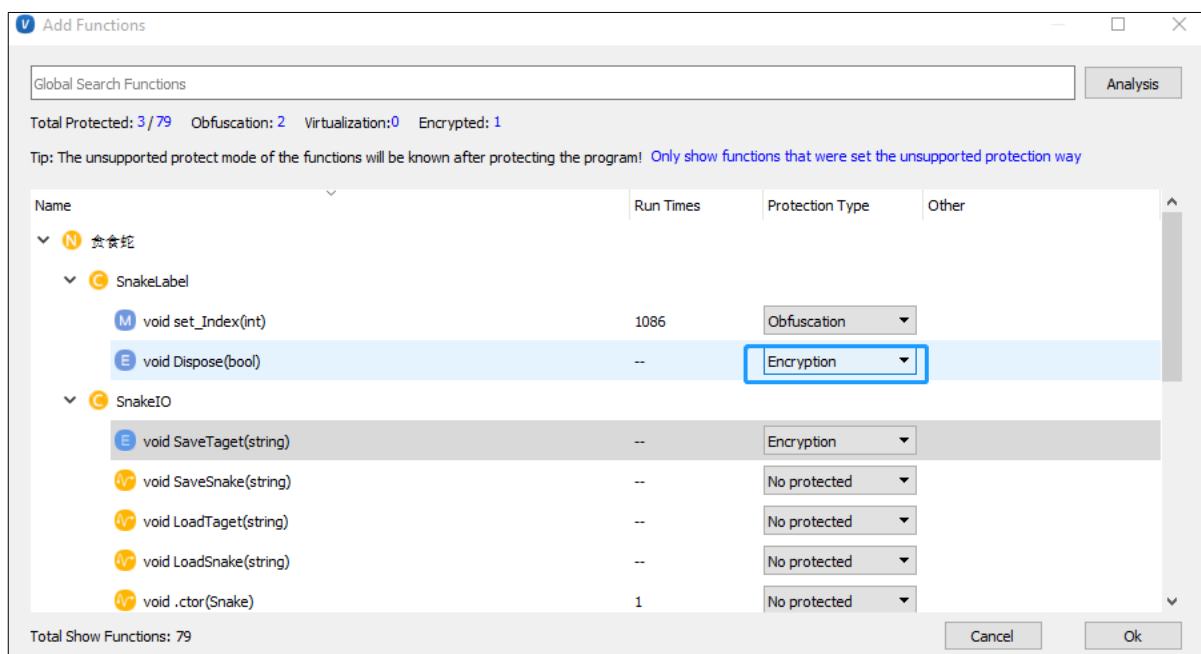


Figure 3-14

Note: For .Net Programs, Support function protection options includes: **No protected, Obfuscation, Encryption;**

For **Other** Programs (PE or native program): Support Function protection options: **No protected, Obfuscation and Virtualization, Encryption.**

Protection Mechanism:

Prevent from being unpacking, and prevent the program from being dumped directly.

The Benefit:

Almost no impact to software performance.

The Weakness:

Low Security: It is possible be decrypted and by analyzed to the protected functions.

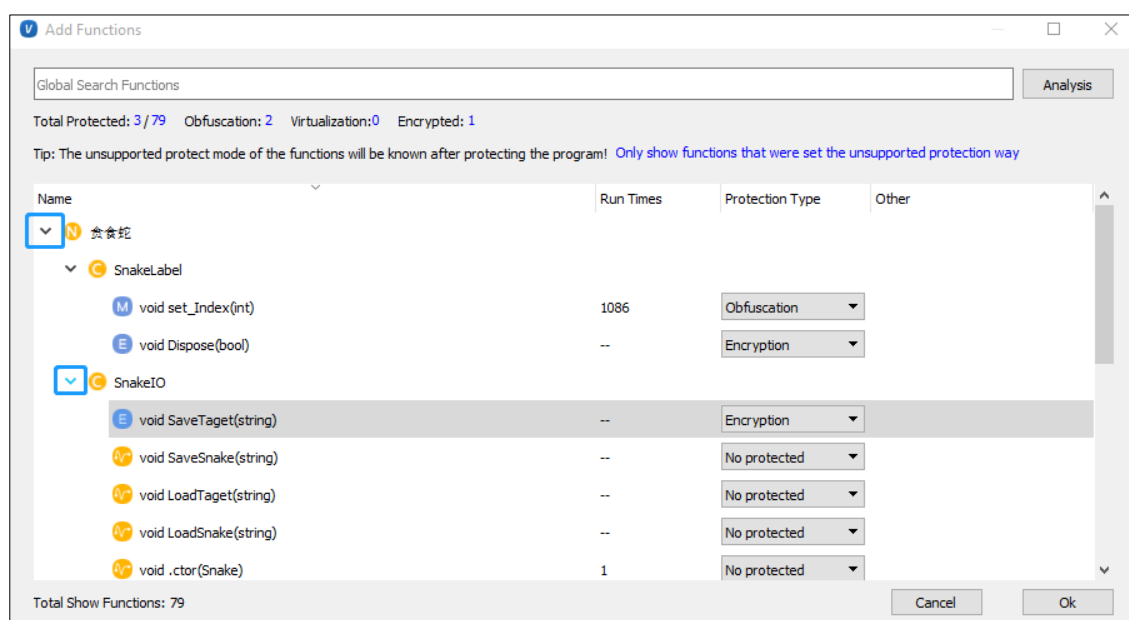


Figure 3-15

Tips: You can click the icon as showing in the picture above to open the function.

Performance Analysis

After you selected the protection mode for the function, you can start to analysis execution performance by clicking "**Analysis**" before finalize your software protection scheme. The Analysis function will show you the software execution performance and the calling times of the protected functions when execution.

After you completed analysis, the functions called times will be displayed in the middle of the panel. For how

long time you run the program you are protecting depends on your actual requirement.

Note: If the program you are analyzing is **DLL** libs, please start the main program. We currently support EXE/ELF program and DLL (Dynamic Link Library) and so library protection.

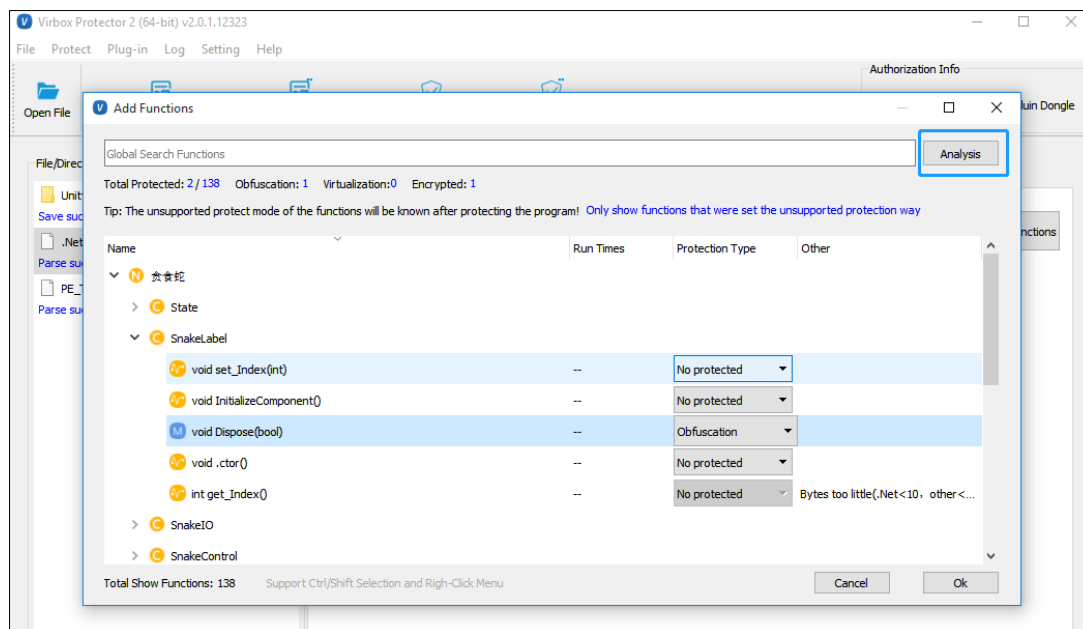


Figure 3-16

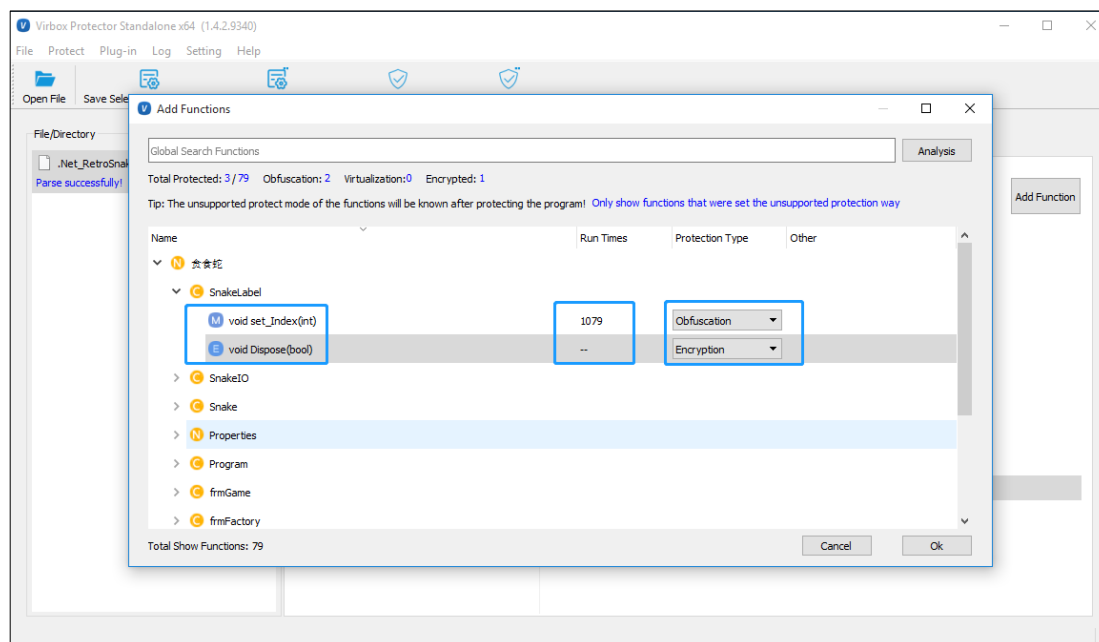


Figure 3-17

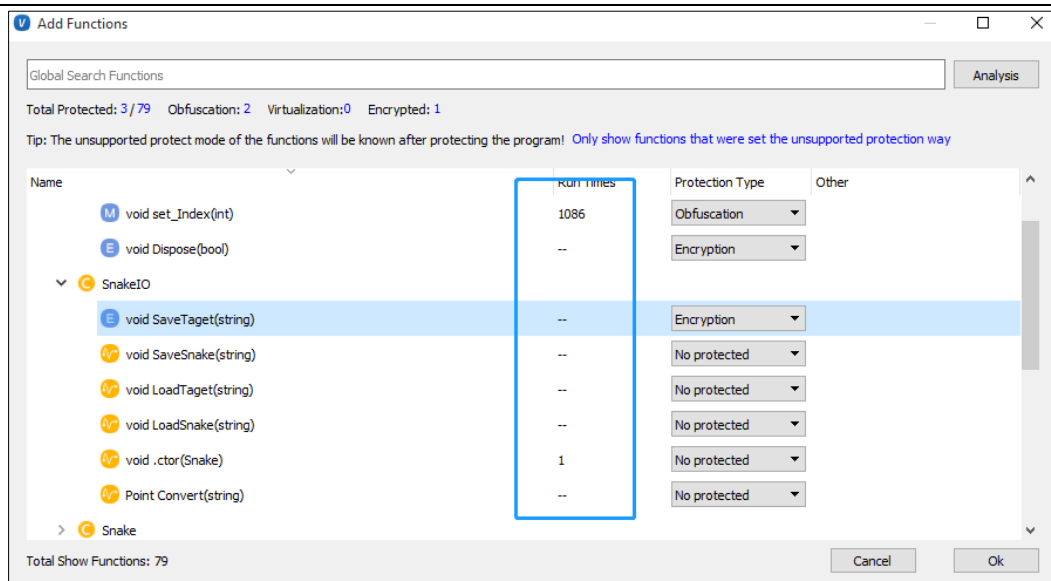


Figure 3-18

Function search:

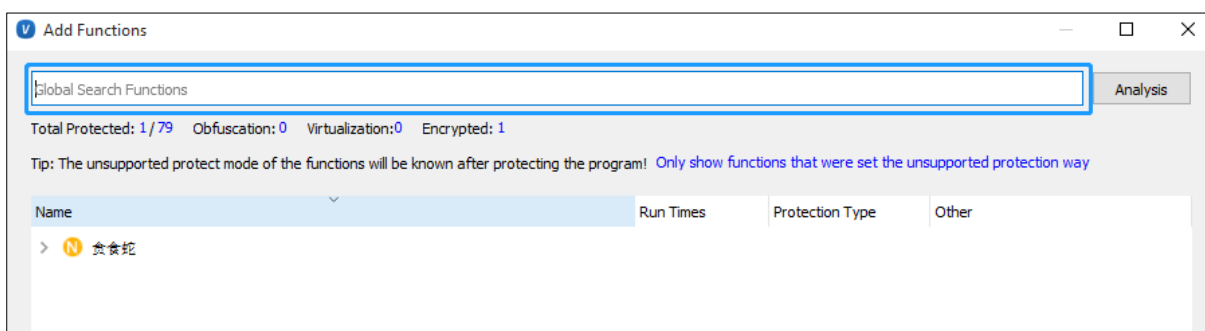


Figure 3-19

After you entered the keyword of the function, Virbox Protector will list the functions which contain the keyword, fuzzy query supported.

3.3.2.3 Protection Options

Protection Option setting will be different for different program, This "**Protection Option**" function has little difference to PE (local program) and .NET application due to difference of PE and .NET technology and Gaming software based Unity3D. Developer can select and setup these "**Protection Option**" in actual project.

For **.NET program**, following options could be set and selected:

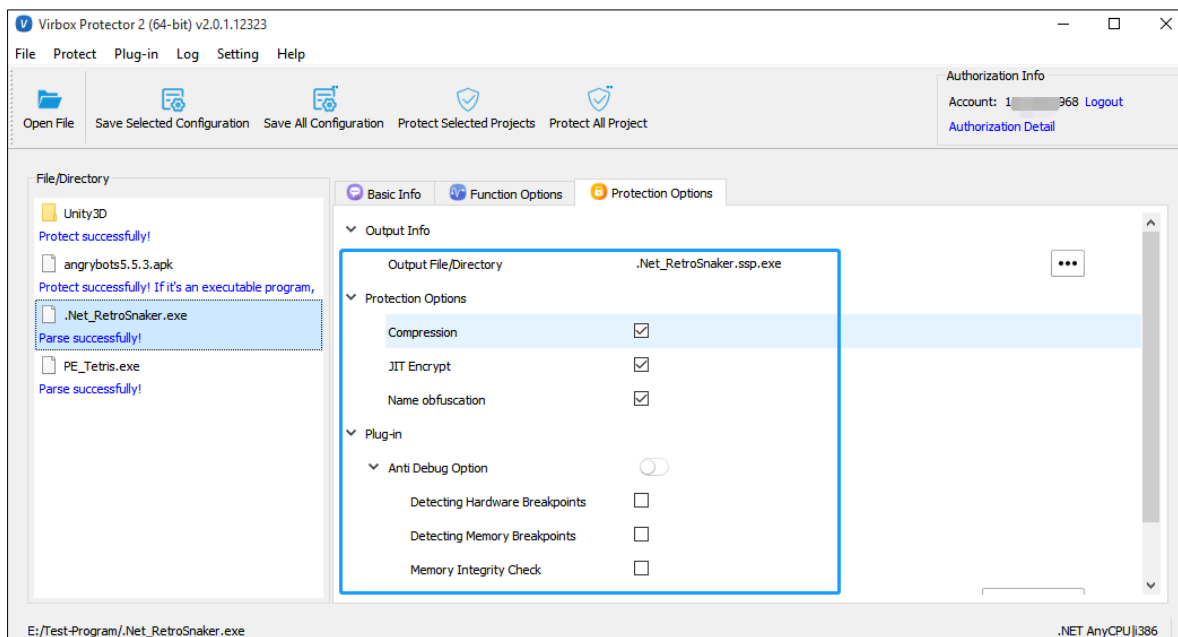


Figure 3-20

For **local program (PE)**, following options could be set and selected:

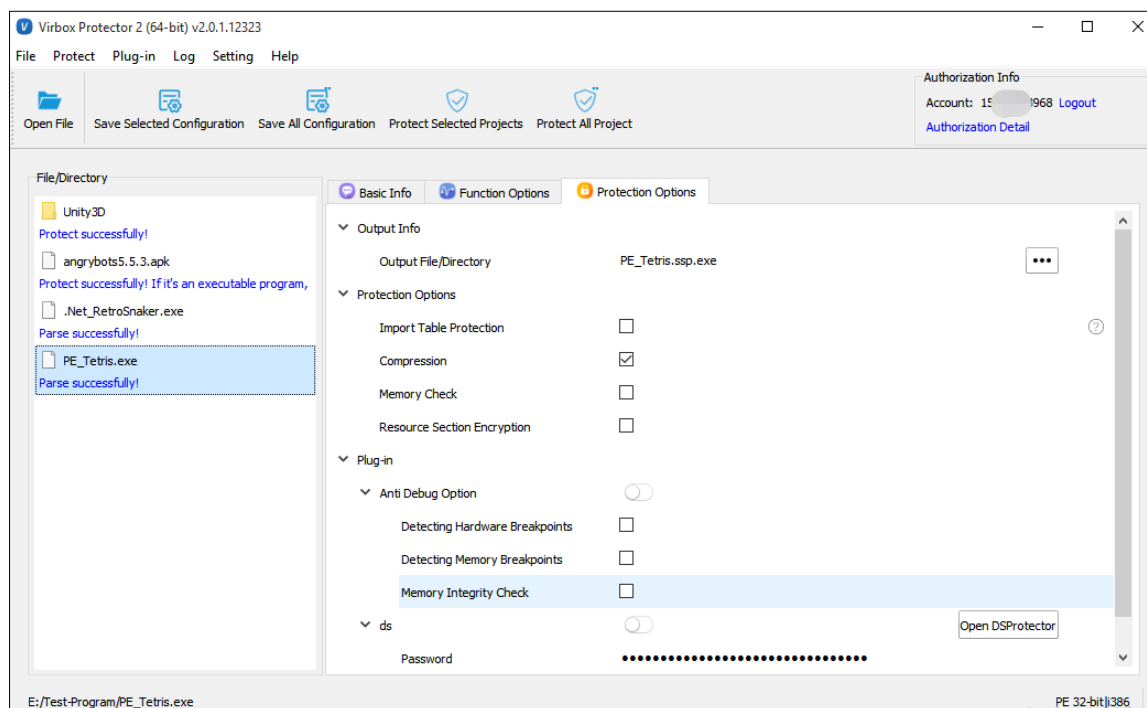


Figure 3-21

For **Unity3D file**, following options will be displayed and selected:

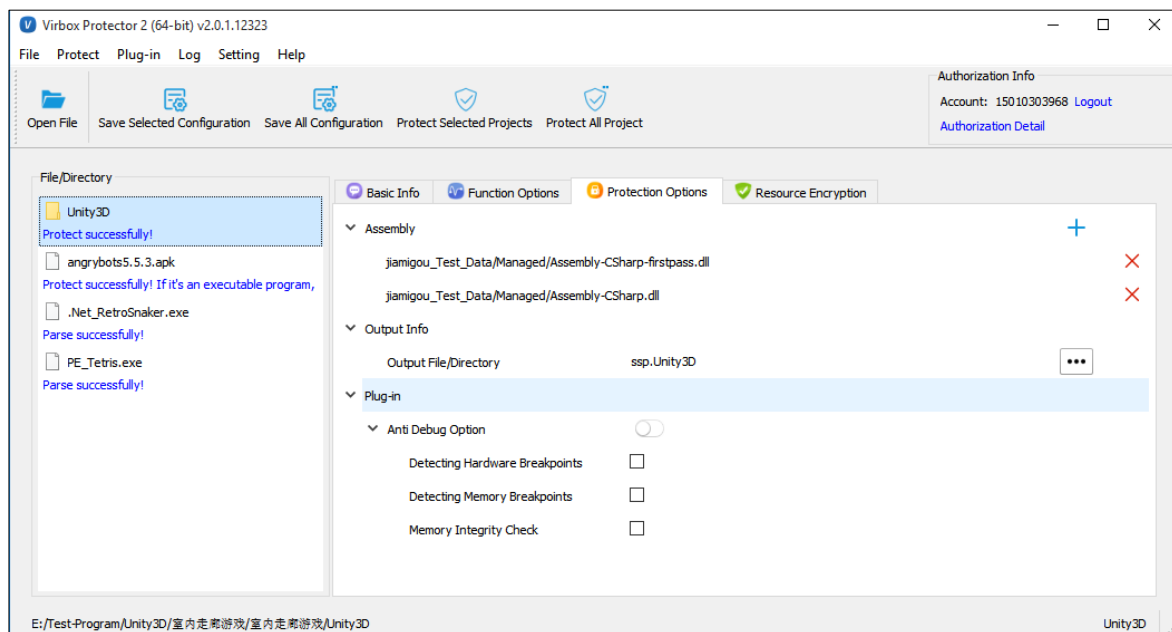


Figure 3-22

Click the "**Protection Option**"

1. **Output File:** Here, it will show the location of the protected/encrypted program. And you can change the output file path of the protected software.
2. **Compression:** Compression means to compress the application after protection and reduce size of application, it is also prevents static anti-compile the software application by hacker who use static anti-compiler tools. When you select Compression: It will keep and control the size of protected software application not too big size. Also will enhance protected software's security level after compression; for the program with big size. This function would be obvious to make the program smaller size.

The original purpose of the compression function is not compress the software size, it will encrypt the code and the data segment and hide the original import table and relocate information, and compressed the original data at same time.

Protection Mechanism:

This function will pack the original data segment with data package and compress the file, replace the original code entry with the packer code. The data segment and code segment will be retrieved when the program is executed, and relocate and to execute the program.

The Purpose:

Prevent the static anti-compile and prevent the program being patched

The Benefit:

Hide the code, data and file structure information of the program, protect the software in overall.

High efficient when the program is run, and only relatively weak performance loss when the program is loaded.

The Weakness:

When the packer code is executed, the code segment and data segment may be retrieved and be dumped.

3. JIT encryption:

.Net JIT encryption will encrypt all of the method IL instruction of the .Net Program, and only in the JIT compile process of the .Net Virtual Machine the instruction will be decrypted, This can be used to prevent static anti-compile and prevent the IL code being Dumped in memory.

JIT encryption will encrypt all of the method in default and enhance the security level of the source code after protection.

JIT encryption .support inheritance, event, reflection, recursive call which is not supported in general encryption solution.

4. **Name Obfuscation:** Select this Option, developer can obfuscate the program file name and transforming software program name into the pseudo code which cannot be identified by use of Static Anti-compiling Tools and then convert these Pseudo code into original program name when execute the protected software.
5. **Anti-Debug option:** you can use the following anti-debug option by clicking this button.
The anti-debug function, including Detecting Hardware Breakpoint, Detecting Memory Breakpoint, Memory Integrity Check. To prevent your program/application from being debugged by the tools, such as: ollydbg or Windbg. **The platform supported:** Windows, Linux, ARM Linux, Android so library and Android Unity3d application.
6. Detecting Hardware breakpoint: when this function is enabled to the protected software, the program will stop execution if memory access breakpoint and memory write breakpoint has been detected.
7. Detecting memory breakpoint: This function will protect your software by exit the program if the program has been detected to be setting memory breakpoint.
8. Memory Integrity Check: when this function enabled, the program will be terminated execution if the memory modification has been detected (e.g.: Being modified by debugger).

For Linux, ARM Linux, Android so library and Android Unity3D program:

The anti-debug plugin function supports to detect the debug tool, and prevent the program being anti-compiled, such as: gdb, IDA, etc tool.

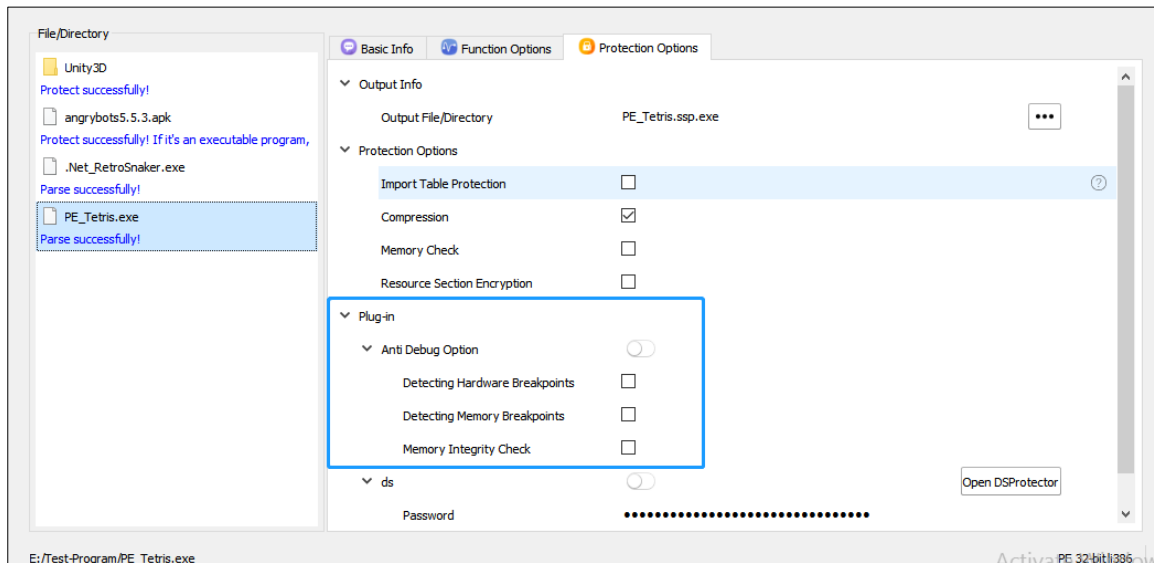


Figure 3-23

Following picture will show you the result, if you have enabled this function

Use GDB to debug the protected program used the anti-debug option:

```
(gdb) r
Starting program: /home/sense/Desktop/0509antitest/asrproxy/asrproxy
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
This application is protected with unregistered version of VirboxProtector. 6 da
ys left

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) c
Continuing.

Program terminated with signal SIGABRT, Aborted.
The program no longer exists.
(gdb)
```

Figure 3-24

Use IDA to debug the protected program used the anti-debug option:

```
Output window
F1FFAB1C: thread has started (tid=28720)
Debugger: thread 28720 has exited (code 0)
F1FFAB1C: thread has started (tid=28721)
D3515000: loaded /data/app/com.KLS.LetteClear-Xgwvm2xcylu6QvChWmk7Hg==/lib/arm/libmain.so
D33C7000: loaded /system/framework/oat/arm/gson.odex
D381E3EA: got SIGSEGV signal (Segmentation violation) (exc.code b, tid 28663)
F204124C: got SIGSEGV signal (Segmentation violation) (exc.code b, tid 28663)
D2F00000: loaded /data/app/com.KLS.LetteClear-Xgwvm2xcylu6QvChWmk7Hg==/lib/arm/libmono.so
F1FFDFAA: got SIGABRT signal (Abort) (exc.code 6, tid 28663)
F1FFDFDC: got SIGABRT signal (Abort) (exc.code 6, tid 28663)
```

Figure 3-25

9. **Ds (Resource protection):** Encrypts the resource section of the protected program, DS Protector is a data resource protection tool that encrypts the data resource files of the program. When you are using this function, you need to switch on the button to "**green**".
10. **Password:** You can also set a password, letters and numbers are supported, but it should not be longer than 64 characters.
11. **DSProtector button:** You can open DSProtector by clicking this button.
12. **Import Table Protection:** Developer may select this option to protect "Import table" which imported to PE Program and encrypt this table, protect the function when the function is called by external program, API list has been hidden and encrypted to enhance the security level to the PE Program, recommend developer select this option. Protect the program by interference the reverse analysis.

The Protection option of "Protect Import Table" support PE program only.

Protection Mechanism:

Remove the original import table, and replace the IAT (Import Address Table) with the repair function, let the packer code take over the function call of the import function.

13. **Resource Section Encryption:** For PE program, Virbox Protector can encrypt the Resource Section in the program, to prevent the resources information from being extracted and tampered illegally.

Protection Mechanism:

When the program is enveloped, Virbox Protector will extract the resource section and encrypt, only the resources that be used externally will be protected (such as program icon, program version information). When the program executed, then these resources will be decrypted.

Note: Only local programs supported to encrypt the resource section.

14. Memory Check:

Memory check is the function implemented by Virbox Protector which is used to check the integrity of the program itself, and can be used to prevent illegal patch, memory patch and software breakpoint. What is more, memory check table and logic check is self protected to make sure the security of the software.

Memory check would be run in the program entry point, Virbox Protector loader will check every memory block to check the integrity. If verified failed, the program will exit.

If SDK label is used, every time you call **VBProtectVerifyImage**, the memory would be checked.

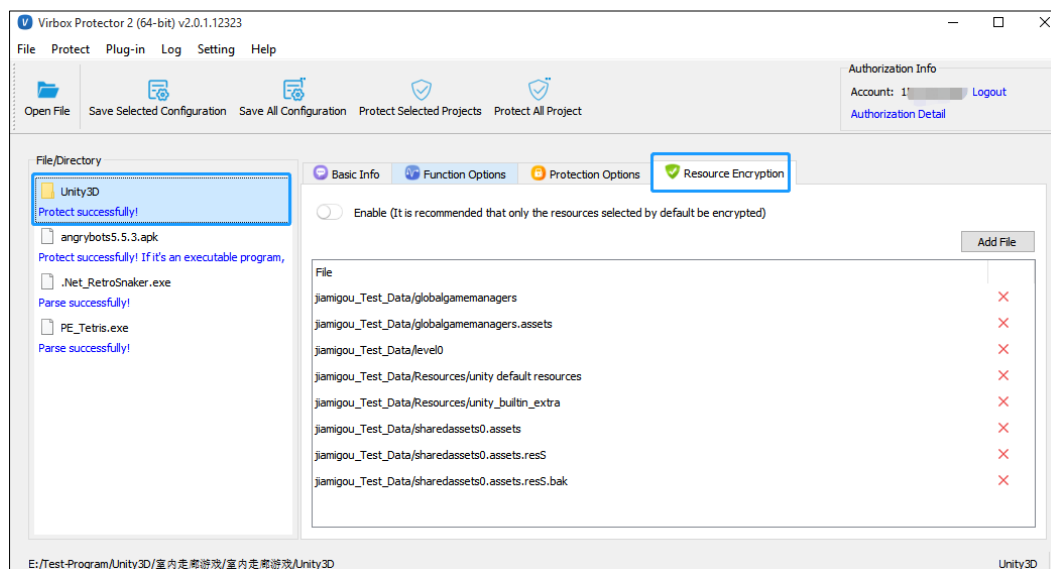
Please noted that the code encryption and Memory check can't be used at same time.

The ds option (Resource encryption) and memory check option can't be used at same time.

3.3.2.4 Resource Encryption

This option is only valid for the program of Unity3D.

You can refer the corresponding chapter for how to use this option.



3.3.2.5 Status bar

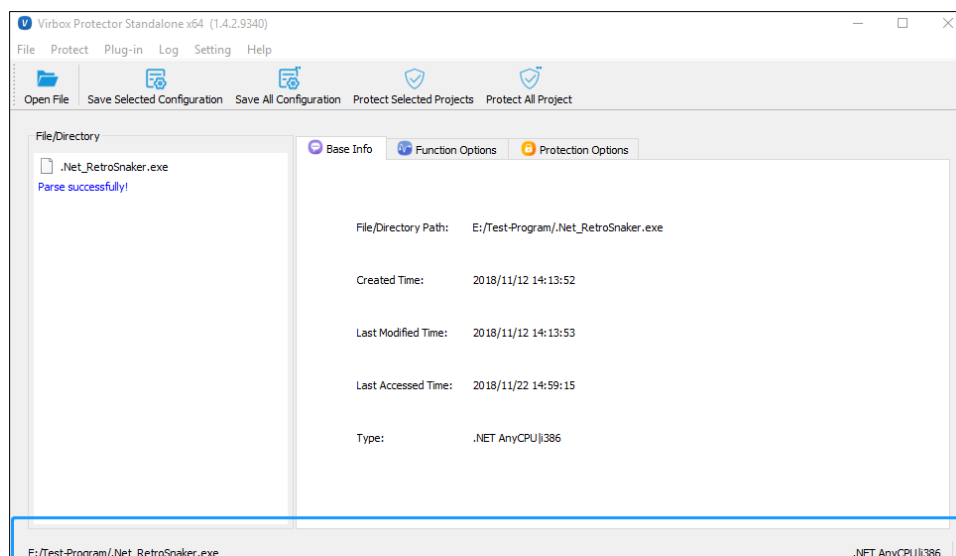


Figure 3-26

In the bottom of the window is the status bar, which will show the corresponding software full location,

software type, and hardware type of the selected software.

To Complete the Protection:

Click the button of "**Protect Selected Project**" to complete the protection process, then prompt with "**Protection Successful**" means the software protection completed. Open the directory where the protected software located, you will find the file: xxx.*ssp*.exe or xxx.*ssp*.dll will be listed in this directory. The executable file that has *ssp* in between filename and extension name is the software application has been protected by Virbox Protector. *Rename* this file name to be the original file name for further evaluation or distribute this protected software in future. Please keep the original software file in safety.

4 The Mechanism of software protection

4.1 Protect the PE program (application) and DLL.

Software Developer use Virbox Protector to protect the executable file and DLL library, with the functions protections Option, Protection Options, "Anti-Debug Option" plug-in feature and other Protection technology, as introduced in Chapter 2 and Chapter 3, Developer may flexible select these functions to protect the software functions, codes, critical algorithms and evaluate the software execution performance.

- Following protection process will be implemented:

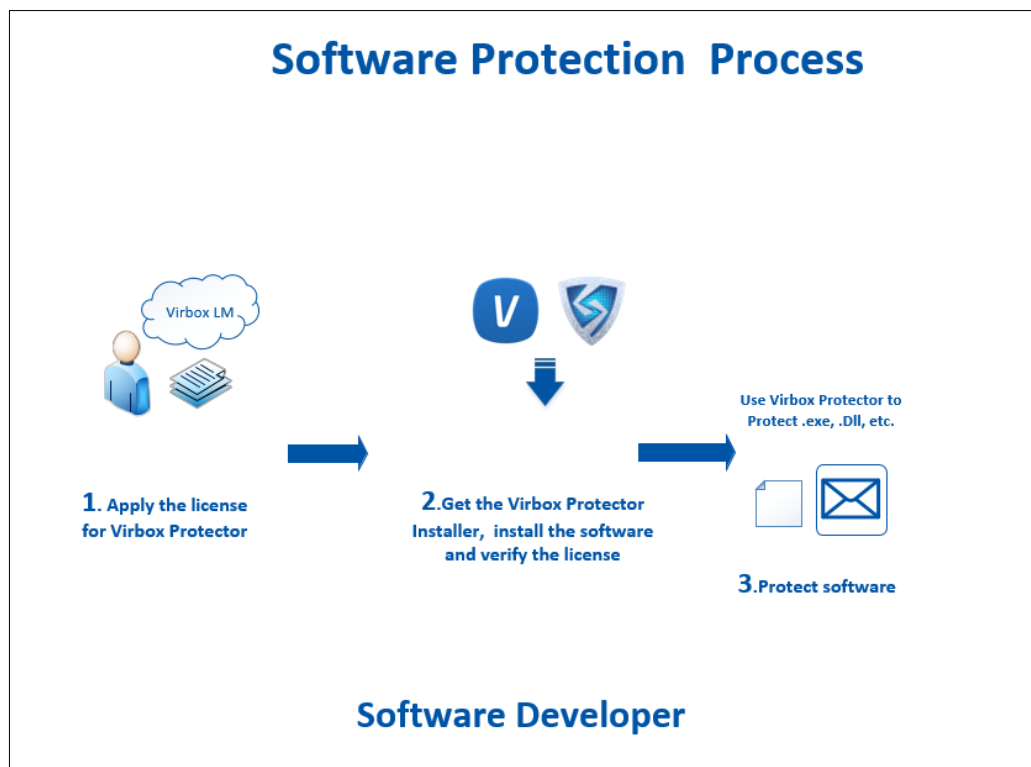


Figure 4-1

For example, you have created a .Net based C# language executable. You can use the above process to protect your software.

4.2 Protect the interpreter and code resource file (Python, PHP, etc.)

Software developer use the Virbox Protector and plug-in Unit (DS Protector) to protect the interpreter and related source code or resource files.

Software need to build an execution environment, for example, install a python environment on your PC and execute the **.py** or **.pcy** file (Or execute an **Mp3**, **Mp4** file with a media player).

- Following protection process will be implemented:
 - Use Virbox Protector to protect interpreter (Python.exe, player, etc.);
 - Use DS Protector to protect the source code or data resources (.py, media files, etc.);

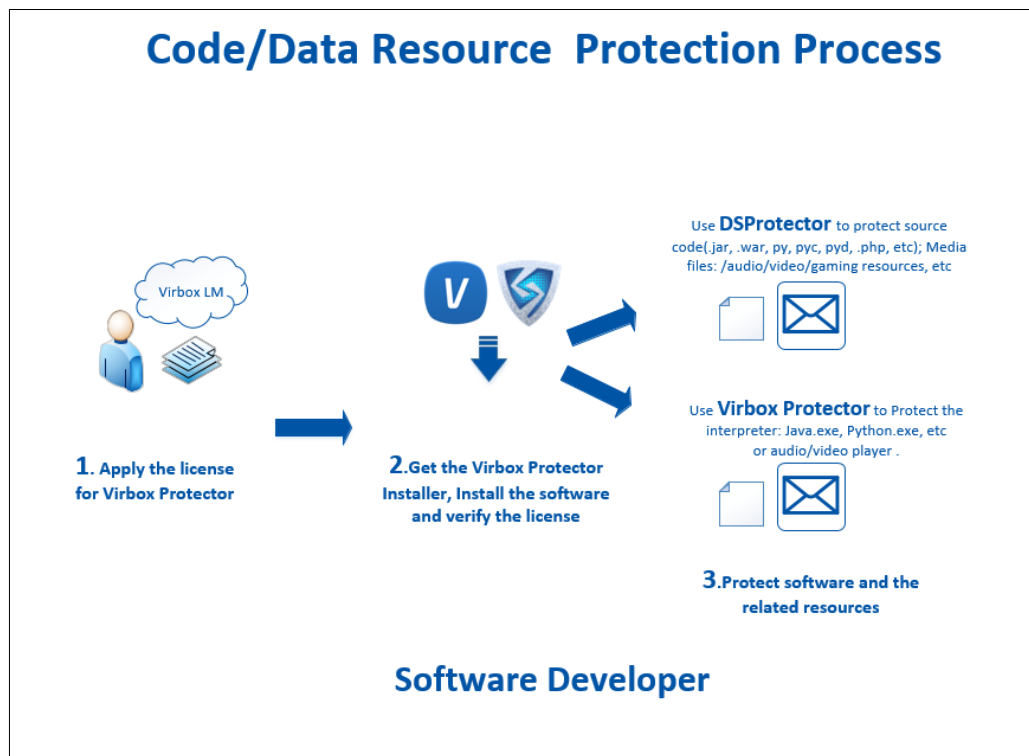


Figure 4-2

4.3 Make the protection scheme for your software

When you open the Virbox Protector, you can directly drag the windows Application to the Virbox Protector to protect, as shown in the figure below:

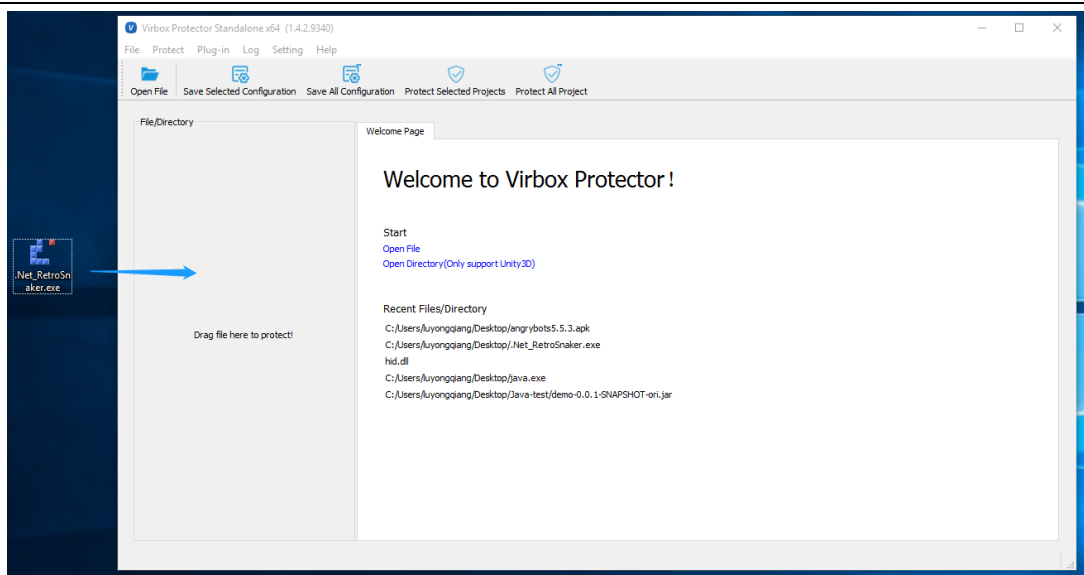


Figure 4-3

You can make your dedicated protection scheme and "configure" the protection options by select following Function Options and Protection Options as shown in below:

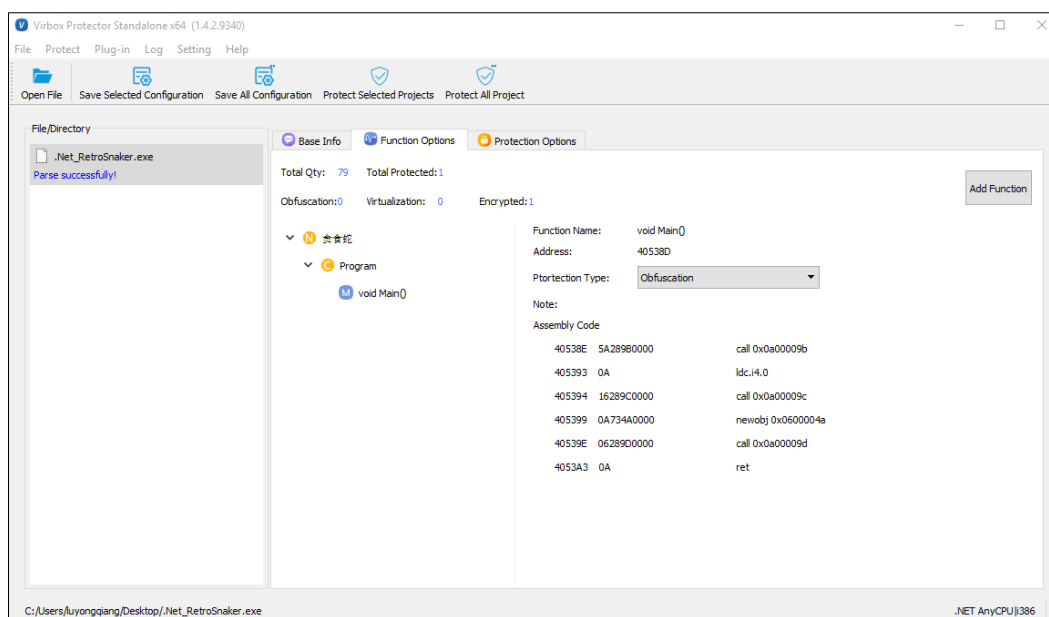


Figure 4-4

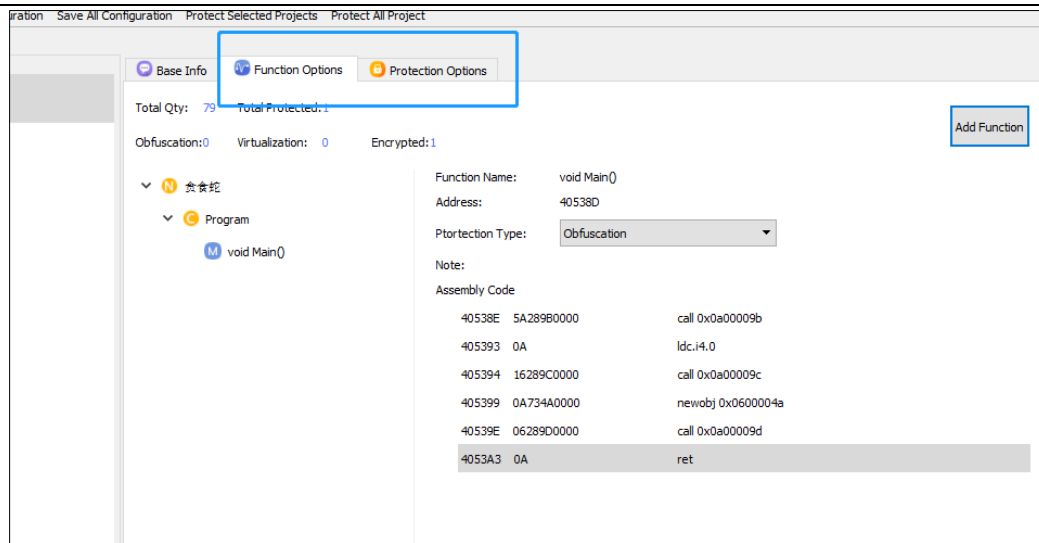


Figure 4-5

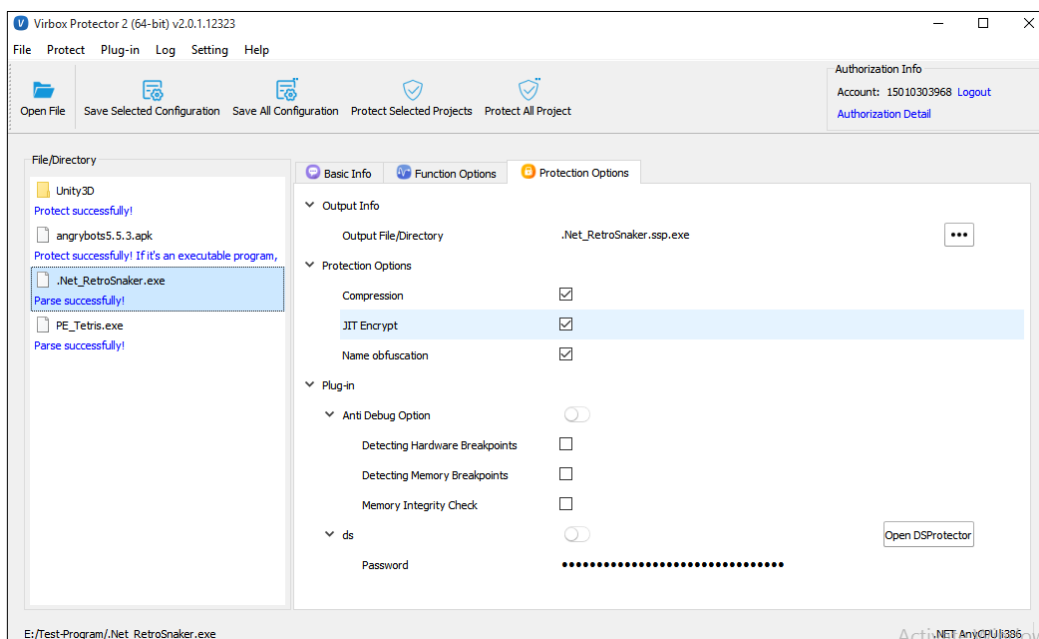


Figure 4-6

Select different technology to encrypt your function/method: **Virtualization, Advanced obfuscation, Code Encryption**. And also you can protect your program overall by selecting these options: **Import table protection, Smart compression, Name obfuscation, Detecting Hardware Breakpoints, Detecting Memory Breakpoints, Memory Integrity Check**; The resource of your program can also be protected by “**Resource Section Encryption**” option.

In this way to protect your software statically and dynamically.

5 Protection Example & Use Case

Before software protection:

With the Basic protection and Function level protection provided by Virbox Protector, it is not only to protect the software overall but also to protect the selected critical functions of the software and enhance software security. With the "**Performance Analysis**" feature, the protected software can get highly security and also no big impact to the software performance.

You can find all the basic protection options from the "**Protection Options**" Tag, and you can protect the functions of your software by selection from the "**Function Options**" Tag. For details information you can refer the below sample.

5.1 Protect the Local Executable

Local executable include: PE (Windows), ELF (Linux), Mach-O (macOS) format executable.

5.1.1 Fundamental protection to software

5.1.1.1 Import Table Protection

Hide the import table of the original program to protect the functions called by external program. With this way, to against the reverse engineering analysis and prevent the unpacking of the program.

The Program supported to be protected by using the "Import Table Protection":

Only PE format program/executable files supported.

Protection Mechanism:

Remove the import table of the original program, replace the Import Address Table (IAT) with stub function, and let the Virbox Protector loader take over the invoking of the import functions.

5.1.1.2 Resources encryption

Resources protection/encryption is for PE format program/encryption function, used to protect the resources and prevent the resources from being extracted illegally or tampered.

Protection Mechanism:

The resources in the PE program will be extracted and encrypted by Virbox Protector while it is protected. Those externally used resources will be decrypted in the Virbox Protector loader program when the program is executed (such as software icon and software version information).

5.1.1.3 Additional data extension

What is Additional extension data?

Additional data is the data (video or database) being combined with the original executable by the compiler or packer tool, these data will be read by the original executable when executed and will not be mapped into the memory directly.

Purpose

As the loader of software protector will modify the original program, if directly merge or join the additional extension data into the protected program, abnormal may happen when the program is executed.

Hook is used in additional data extension to read the additional data normally, and encrypt the additional data to prevent the data being used illegally.

5.1.1.4 Compression

Protection Mechanism

When executed the compression of Virbox Protector, it packs the original code segment and data segment of software and compress the software, it will replace the Original Entry Point (OEP) with the Virbox Protector's code (loader). The data segment and code segment will be retrieved when the program is executed, and relocate to execute the program.

Purpose:

Prevent the static anti-compile and prevent the source program being patched

The Benefit:

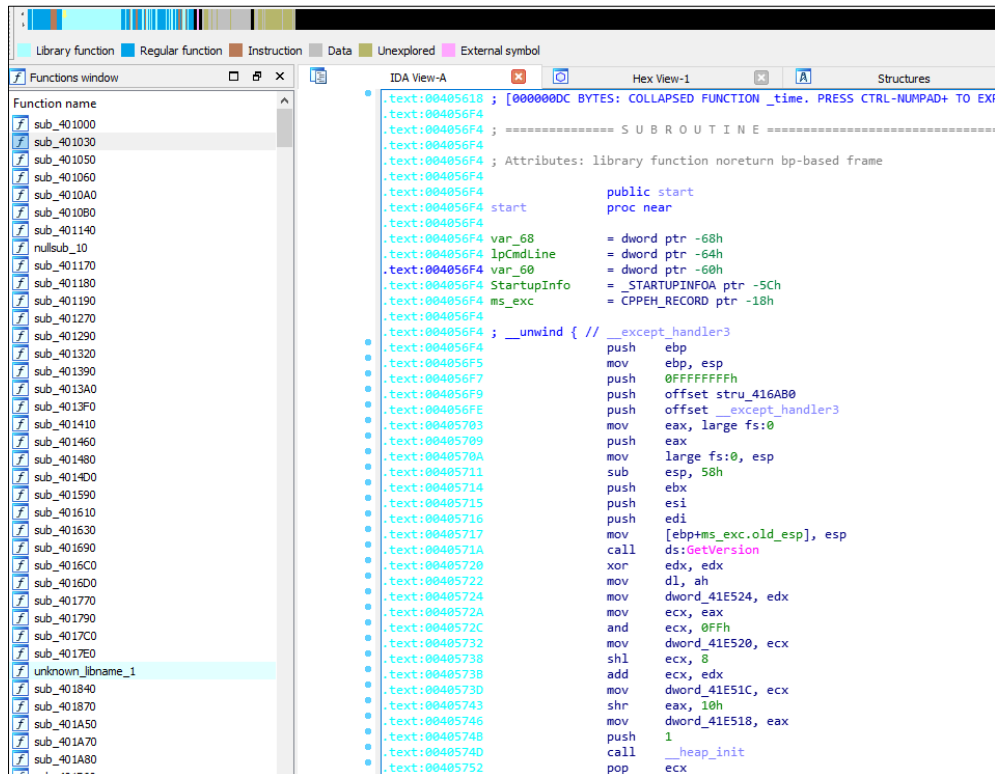
1. Hide the code, data and file structure information of the program, protect the software in overall.
2. Highly efficiency when the program executed, and mini impact to performance when the program is loaded.

The Weakness:

After the Protector code (loader) executed, the code segment and data segment possible be retrieved and be dumped.

Comparison:

Before Protection:



The screenshot shows the IDA Pro interface with the 'Functions window' on the left and the 'IDA View-A' window on the right. The 'Functions window' lists various subroutines, including 'sub_401000', 'sub_401030', 'sub_401050', 'sub_401060', 'sub_4010A0', 'sub_4010B0', 'sub_401140', 'sub_401170', 'sub_401180', 'sub_401190', 'sub_401270', 'sub_401290', 'sub_401320', 'sub_401390', 'sub_4013A0', 'sub_4013F0', 'sub_401410', 'sub_401460', 'sub_401480', 'sub_4014D0', 'sub_401590', 'sub_401610', 'sub_401630', 'sub_401690', 'sub_4016C0', 'sub_4016D0', 'sub_401770', 'sub_401790', 'sub_4017C0', 'sub_4017E0', 'unknown_libname_1', 'sub_401840', 'sub_401870', 'sub_401A50', 'sub_401A70', 'sub_401A80', and 'sub_401B60'. The 'IDA View-A' window displays the assembly code for the 'start' function, which is a public near procedure. The code includes variable declarations for 'var_68', 'lpCmdLine', 'var_60', 'StartupInfo', and 'ms_exc'. It also shows a call to 'ds:GetVersion' and a call to 'heap_init'.

```

.text:00405618 ; [000000DC BYTES: COLLAPSED FUNCTION _time. PRESS CTRL-NUMPAD+ TO EXP
.text:004056F4 ; ===== SUBROUTINE =====
.text:004056F4 ; Attributes: library function noreturn bp-based frame
.text:004056F4 ;
.text:004056F4 public start
.text:004056F4 proc near
.text:004056F4 var_68 = dword ptr -68h
.text:004056F4 lpCmdLine = dword ptr -64h
.text:004056F4 var_60 = dword ptr -60h
.text:004056F4 StartupInfo = _STARTUPINFOA ptr -5Ch
.text:004056F4 ms_exc = CPPEH_RECORD ptr -18h
.text:004056F4 ;
.text:004056F4 ; __unwind { // __except_handler3
.text:004056F4 push ebp
.text:004056F5 mov ebp, esp
.text:004056F7 push 0FFFFFFFh
.text:004056F9 push offset stru_416AB0
.text:004056FE push offset __except_handler3
.text:00405703 mov eax, large fs:0
.text:00405709 push eax
.text:0040570A mov large fs:0, esp
.text:00405711 sub esp, 58h
.text:00405714 push ebx
.text:00405715 push esi
.text:00405716 push edi
.text:00405717 mov [ebp+ms_exc.old_esp], esp
.text:0040571A call ds:GetVersion
.text:00405720 xor edx, edx
.text:00405722 mov dl, ah
.text:00405724 mov dword_41E524, edx
.text:0040572A mov ecx, eax
.text:0040572C and ecx, 0FFh
.text:00405732 mov dword_41E520, ecx
.text:00405738 shl ecx, 8
.text:0040573B add ecx, edx
.text:0040573D mov dword_41E51C, ecx
.text:00405743 shr eax, 10h
.text:00405746 mov dword_41E518, eax
.text:00405748 push 1
.text:0040574D call _heap_init
.text:00405752 pop ecx

```

Figure 5-1

After Protection:

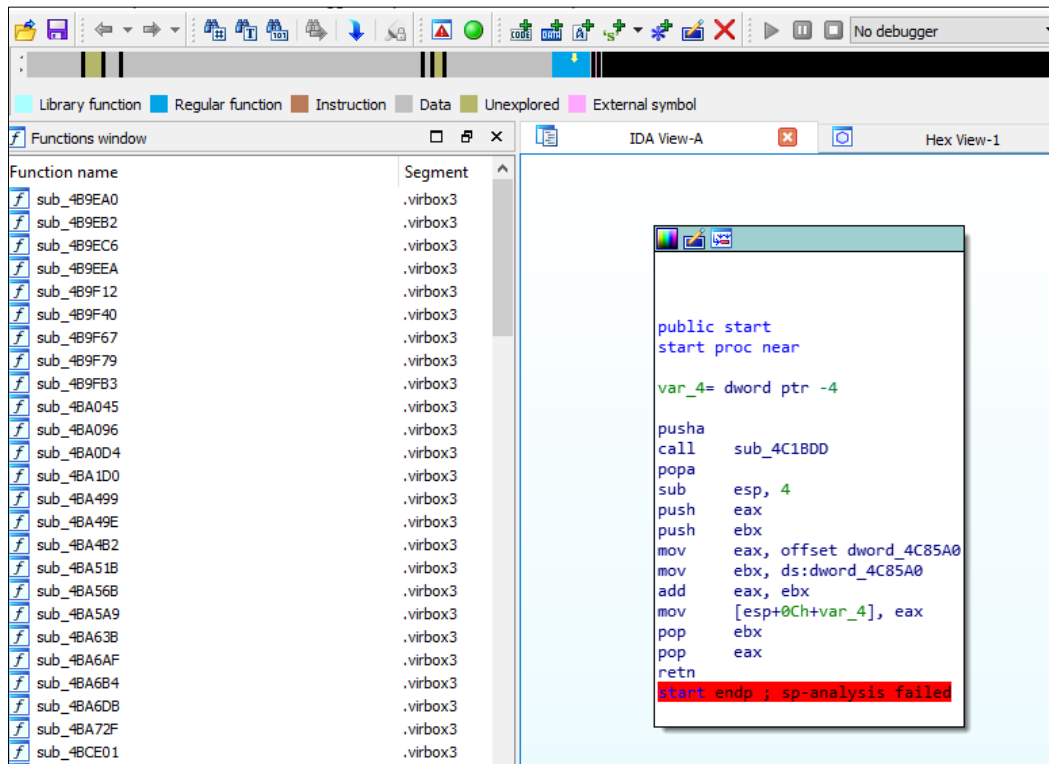


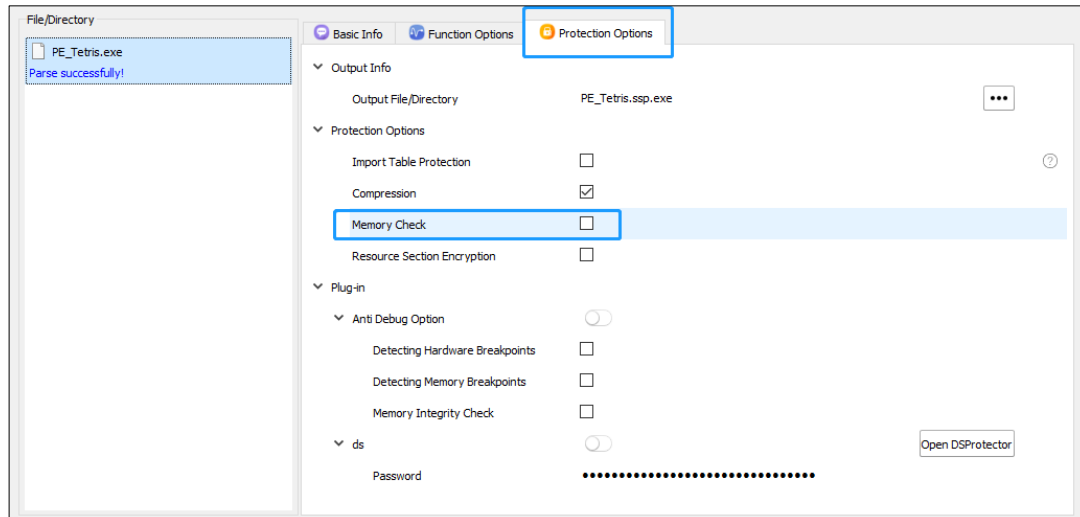
Figure 5-2

5.1.1.5 Memory Check:

1. Memory check is the function implemented by Virbox Protector which is used to check the integrity of the program itself, and can be used to prevent illegal patch, memory patch and software breakpoint. What is more, memory check table and logic check is self protected to make sure the security of the software.
2. Memory check would be run in the program entry point, Virbox Protector loader will check every memory block to check the integrity. If verified failed, the program will exit.
3. If SDK label is used, every time you call **VBProtectVerifyImage**, the memory would be checked.

Instructions for Use:

Drag the PE or ELF program to Virbox Protector, the “**Memory check**” would be shown in the “Protection Option”, you need to select this option to protect the program with memory check.



5.1.2 Protect the critical Functions of software with following technology

5.1.2.1 Code obfuscation

Protection Mechanism:

Select "**Obfuscation**": **Virbox Protector** will translate the code instruction into a stream of pseudo-code that neither the machine nor the human can identify these pseudo code. When the pseudo-code executed, the software will translate and interpret to restore the code into the original code to execute.

Virbox Protector support the obfuscation to **x86/arm /.net** il series instruction.

The Purpose

Obfuscate the source instruction and prevent the program from being static analyzed.

The Benefit:

Prevent from anti-compiled and make it more difficult to analysis the code.

The Weakness:

Performance impact.

Protection Comparison:

Before protection(X86):

```
.text:004144B5 ; __unwind { // loc_414CD8
.text:004144B5      mov     eax, offset loc_414CD8
.text:004144BA      call    __EH_prolog
.text:004144BF      push    ecx
.text:004144C0      mov     [ebp+var_10], ecx
.text:004144C3      mov     dword ptr [ecx], offset off_416508
.text:004144C9 ; try {
.text:004144C9      and     [ebp+var_4], 0
.text:004144CD      add     ecx, 4
.text:004144D0      push    ecx ; void **
.text:004144D1      call    ?AfxDeleteObject@@YGXPAPAX@Z ; AfxDeleteObject(void * *)
.text:004144D6      mov     ecx, [ebp+var_C]
.text:004144D9      mov     large fs:0, ecx
.text:004144E0      leave
.text:004144E1      retn
.text:004144E1 ; } // starts at 4144C9
.text:004144E1 ; } // starts at 4144B5
.text:004144E1 sub_4144B5      endp
.text:004144E1
```

Figure 5-3

After protection(X86):

```
.text:004144B5
.text:004144B5 loc_4144B5:      ; CODE XREF: sub_412E0B+3↑p
.text:004144B5      call    sub_466CF0
.text:004144BA      out     6Ch, eax
.text:004144BC      sub     [esi+50h], ebp
.text:004144BF      call    sub_466D94
.text:004144C4      push    ss
.text:004144C5      adc     dword ptr [edi-37245E23h], 64h
.text:004144C5 ; -----
.text:004144CC      db 0F7h
.text:004144CD ; -----
.text:004144CD loc_4144CD:      ; CODE XREF: .text:0041453D↓j
.text:004144CD      dec     ebx
.text:004144CE      or      ebp, edi
.text:004144D0      adc     bl, bh
.text:004144D2      movsb
.text:004144D3      cmc
.text:004144D4      mov     [ebx-18h], edx
.text:004144D7      test    [ecx], ebp
.text:004144D9      add     eax, 11028200h
.text:004144DE      jle     short loc_41447C
.text:004144E0      mov     cl, 10h
.text:004144E2
```

Figure 5-4

ARM architecture source code:

Before protection

```
.text:00076E9C ; android::register_android_database_CursorWindow(_JNIEnv *)
.text:00076E9C EXPORT _ZN7android38register_android_database_CursorWindowEP7_JNIEnv
.text:00076E9C _ZN7android38register_android_database_CursorWindowEP7_JNIEnv
.text:00076E9C ; CODE XREF: android::register_android_database
; DATA XREF: LOAD:00018730to ...
.text:00076E9C ; __unwind {
.text:00076E9C PUSH {R3-R7,LR}
.text:00076E9C MOV R4, R0
.text:00076EA0 LDR R6, =(aAndroidDatabas - 0x76EA8)
.text:00076EA2 LDR R3, [R0]
.text:00076EA4 ADD R6, PC ; "android/database/CharArrayBuffer"
.text:00076EA6 LDR R2, [R3,#0x18]
.text:00076EA8 MOV R1, R6
.text:00076EAA BLX R2
.text:00076EAC MOV R5, R0
.text:00076EAE CBNZ R0, loc_76EBE
.text:00076EB0 LDR R0, =(aClazzNull - 0x76EBA)
.text:00076EB2 LDR R1, =(aCursorWindow - 0x76EBC)
.text:00076EB4 LDR R2, =(aUnableToFindCl - 0x76EBE)
.text:00076EB6 ADD R0, PC ; "clazz == NULL"
.text:00076EB8 ADD R1, PC ; "CursorWindow"
.text:00076EBA ADD R2, PC ; "Unable to find class %s"
.text:00076EBC B loc_76EE2
.text:00076EBE ;
.text:00076EBE loc_76EBE ; CODE XREF: android::register_android_database
; (aData - 0x76EC8)
.text:00076EC0 MOV R1, R5
.text:00076EC2 LDR R0, [R4]
.text:00076EC4 ADD R6, PC ; "data"
.text:00076EC6 R3, =(aC - 0x76ED4)
.text:00076EC8 LDR.W R7, [R0,#0x178]
.text:00076ECC MOV R2, R6
.text:00076ECE MOV R0, R4
.text:00076ED0 ADD R3, PC ; "[C"
.text:00076ED2 BLX R7
.text:00076ED4 CBNZ R0, loc_76EE8
.text:00076ED6 LDR R0, =(aResNull - 0x76EE0)
.text:00076ED8 LDR R1, =(aCursorWindow - 0x76EE2)
.text:00076EDA LDR R2, =(aUnableToFindSt - 0x76EE4)
.text:00076EDC ADD R0, PC ; "res == NULL"
.text:00076EDE ADD R1, PC ; "CursorWindow"
.text:00076EE0 ADD R2, PC ; "Unable to find static field %s"
.text:00076EE2 loc_76EE2 ; CODE XREF: android::register_android_database
; android::register_android_database_CursorWin
.text:00076EE4 MOV R3, R6
.text:00076EE4 BLX __android_log_assert
00076EBE 00076EBE: android::register_android_database_CursorWindow(_JNIEnv *)loc_76EBE (Synchronized with Hex View-1)
```

Figure 5-5

After Protection:

```
.text:00098E9C ; android::register_android_database_CursorWindow(_JNIEnv *)
.text:00098E9C EXPORT _ZN7android38register_android_database_CursorWindowEP7_JNIEnv
.text:00098E9C _ZN7android38register_android_database_CursorWindowEP7_JNIEnv
.text:00098E9C ; DATA XREF: LOAD:0000C8E0to
.text:00098E9C ; __unwind {
.text:00098E9C B.W _ZN7android38register_android_database_CursorWindowEP7_JNIEnv
.text:00098E9C ; End of function android::register_android_database_CursorWindow(_JNIEnv *)
.text:00098EA0 ;
.text:00098EA0 SUBS R6, #0xF6
.text:00098EA2 STRB R4, [R4,#0x1B]
.text:00098EA4 loc_98EA4 ; CODE XREF: .text:0003AB81j
.text:00098EA4 ADD R6, PC
.text:00098EA6 B.W loc_3A0CC
.text:00098EAA ;
.text:00098EAA BLX R2
.text:00098EAC MOV R5, R0
.text:00098EAE CBNZ R0, loc_98EBE
.text:00098EB0 B.W loc_3A148
.text:00098EB4 ;
.text:00098EB4 B loc_98BE6
.text:00098EB6 ;
.text:00098EB6 ADD R0, PC
.text:00098EB8 ADD R1, PC
.text:00098EBA ADD R2, PC
.text:00098EBC B loc_98EE2
.text:00098EBE ;
.text:00098EBE loc_98EBE ; CODE XREF: .text:00098EA1j
.text:00098EBE B.W loc_3A1C4
.text:00098EBE ;
.text:00098EC2 DCD 0xC7B2
.text:00098EC4 DCD 0xF7A1447E, 0x5A55B9B9, 0x20EE33F4, 0x47B8447B, 0xF7A1B940
.text:00098EC4 DCD 0x271DBA01, 0x44794478
.text:00098EE0 ADD R2, PC
.text:00098EE2 loc_98EE2 ; CODE XREF: .text:00098EBC1j
.text:00098EE2 MOV R3, R6
.text:00098EE4 BLX sub_71384
.text:00098EE8 B.W loc_3A368
.text:00098EE8 ;
.text:00098EEC DCD 0x447F781B, 0xF7A1447E, 0xD581BA99, 0xA4E47EFD, 0x8E01804B
.text:00098EEC DCD 0x47E0447B, 0xF7A1B190, 0x6481BA0B, 0xF7A14479, 0xF7A1B819
.text:00098EEC DCD 0x050EB863, 0x4790233B, 0xF7A1B96B, 0x735CB0A7, 0x44794478
.text:00098EEC DCD 0xE01B447A, 0x8BE2F7A1, 0x44785ED5, 0x447A4479, 0xF7A1E7D3
.text:00098EEC DCD 0xF891BC11, 0x44FED45B, 0x44794A1E, 0xBC4EF7A1, 0x4620447A
.text:00098EEC DCD 0xEC54F7D7, 0x8C8CF7A1, 0x8CDAF7A1, 0x4478ABC7, 0x447A4479
.text:00098EEC DCD 0xEA26F7D8, 0xBF00BDF8, 0x52DAC, 0x4E242, 0x52BA3
00098E9C 00098E9C: android::register_android_database_CursorWindow(_JNIEnv *) (Synchronized with Hex View-1)
```

Figure 5-6

5.1.2.1.1 Anti-Run Trace:

Please noted that the anti-Run Trace is for the ARM architecture based program.

1. Run Trace is the function provided by debugger, every time you run a command in single step, the register status of every command would be recorded, this is a common way to debug trace and anti-obfuscation to the program to be reversed.
2. After the code obfuscation function of Virbox Protector is used for the ARM architecture program, the instruction of the function will set some “hidden pitfall” to test the single step breakpoint, if illegal debug (wrong instruction) command is detected, it will execute some wrong instruction to interrupt the illegal debug and make the program crash and make it impossible to debug the program. In this way to enhance the security of the program you protected with code obfuscation.

Thread	Address	Instruction	Result
0000238D	.text:main	Memory layout changed: 105 segments	Memory layout changed: 105 segments
0000238D			X0=0000000000000001 X1=0000
0000238D	.text:main	STP X29, X30, [SP, #var_20]!	SP=0000007FF1123CE0
0000238D	.text:main+4	MOV X29, SP	X29=0000007FF1123CE0
0000238D	.text:main+8	STR W0, [X29, #0x20+argc]	
0000238D	.text:main+C	STR X1, [X29, #0x20+argv]	
0000238D	.text:main+10	ADRP X0, [REDACTED]	X0=0000005555603000
0000238D	.text:main+14	ADD X0, X0, [REDACTED]	X0=0000005555603110
0000238D	.text:main+18	BL .puts	X30=0000005555958E4
0000238D	.plt:.puts	ADRP X16, #puts_ptr@PAGE	X16=0000005555647000
0000238D	.plt:.puts+4	LDR X17, [X16, #puts_ptr@PAGEOFF]	X17=00000070FE685A48
0000238D	.plt:.puts+8	ADD X16, X16, #puts_ptr@PAGEOFF	X16=00000055556476E8
0000238D	.plt:.puts+C	BR X17; puts	X0=000000000000000A X1=0000
0000238D	.text:main+1C	ADD X0, X29, #0x1C; argc	X0=0000007FF1123CFC
0000238D	.text:main+20	LDR X1, [X29, #0x20+argv]; argv	X1=0000007FF1123D68
0000238D	.text:main+24	BL [REDACTED]	X30=0000005555958F0

5.1.2.2 Code Virtualization

Protection Mechanism:

Virbox Protector will compiles instructions into virtual instructions executed in the specified virtual machine.

The Purpose :

Hide the original instruction, prevent the code logic from being analyzed.

The Benefit:

Highly secured, the original code logic almost can't be identified and analyzed.

The Weakness:

Performance impact.

Please noted that X86, X64 and ARM architecture program are supported to use this function.

5.1.2.3 Code encryption (Native)

Protection Mechanism:

Encrypted the original function of the program by SMC (Self-Modifying Code) technology and only when the program is executed then the function would be decrypted.

The Purpose:

Prevent the program from **being unpacking**, and prevent the program from being dumped.

The Benefit:

No impact to software performance.

The Weakness:

It is possible to decrypt and analyze the functions.

Protection Comparison:

Before protection:

```

.text:004056F4 start      proc near
.text:004056F4             = dword ptr -68h
.text:004056F4 lpCmdLine  = dword ptr -64h
.text:004056F4 var_60     = dword ptr -60h
.text:004056F4 StartupInfo = _STARTUPINFOA ptr -5Ch
.text:004056F4 ms_exc    = CPPEH_RECORD ptr -18h
.text:004056F4 ; __unwind { // __except_handler3
.text:004056F4             push    ebp
.text:004056F5             mov     ebp, esp
.text:004056F7             push    0FFFFFFFh
.text:004056F9             push    offset stru_416AB0
.text:004056FE             push    offset __except_handler3
.text:00405703             mov     eax, large fs:0
.text:00405709             push    eax
.text:0040570A             mov     large fs:0, esp
.text:00405711             sub     esp, 58h
.text:00405714             push    ebx
.text:00405715             push    esi
.text:00405716             push    edi
.text:00405717             mov     [ebp+ms_exc.old_esp], esp
.text:0040571A             call    ds:GetVersion
.text:00405720             xor     edx, edx
.text:00405722             mov     dl, ah
.text:00405724             mov     dword_41E524, edx
.text:0040572A             mov     ecx, eax
.text:0040572C             and     ecx, 0FFh
.text:00405732             mov     dword_41E520, ecx
.text:00405738             shl     ecx, 8
.text:0040573B             add     ecx, edx
.text:0040573D             mov     dword_41E51C, ecx
.text:00405743             shr     eax, 10h
.text:00405746             mov     dword_41E518, eax
.text:0040574B             push    1
.text:0040574D             call    __heap_init
.text:00405752             pop     ecx
.text:00405753             test   eax, eax
.text:00405755             jnz     short loc_40575F
.text:00405757             push    1Ch ; NumberOfBytesWritten
.text:00405759             call    _fast_error_exit

```

Figure 5-7

After Protection:

```

text:004056F4 ; -----
text:004056F4      push    0
text:004056F9      jmp     loc_420000
text:004056F9 ; -----
text:004056FE      db  40h, 42h, 65h, 45h, 39h, 2 dup(0C6h), 14h, 92h, 9, 4Dh
text:004056FE      db  0AFh, 0AEh, 1Ch, 088h, 56h, 8, 4Ch, 38h, 7Dh, 0Dh, 0EAh
text:004056FE      db  0F3h, 8Ch, 19h, 0ACh, 12h, 0C8h, 62h, 4, 36h, 0Ah, 35h
text:004056FE      db  7Ah, 7Fh, 0F8h, 79h, 0E7h, 6, 62h, 4Eh, 51h, 0Ah, 0AAh
text:004056FE      db  0A4h, 0DAh, 1, 0EEh, 8Ah, 0A3h, 2Eh, 3, 0C1h, 69h, 0D0h
text:004056FE      db  0B2h, 61h, 6Dh, 75h, 4Dh, 81h, 92h, 87h, 58h, 0BCh
text:004056FE      db  69h, 33h, 0E7h, 8Bh, 0C9h, 69h, 0C5h, 0BAh, 0CDh, 0C1h
text:004056FE      db  0A7h, 45h, 66h, 8Fh, 0FFh, 71h, 0CFh, 9Dh, 40h, 0B0h
text:004056FE      db  90h, 37h, 16h, 0EEh, 7Dh, 42h, 70h, 2 dup(35h), 1Eh
text:004056FE      db  64h, 2Eh, 83h, 26h, 0D8h, 75h, 0DDh, 3Fh, 0A1h, 0E2h
text:004056FE      db  8Dh, 0A4h, 0F2h, 3, 0A4h, 17h, 5Ch, 49h, 0F8h, 27h
text:004056FE      db  76h, 90h, 0C8h, 7, 0B8h, 7, 0AEh, 89h, 61h, 98h, 98h
text:004056FE      db  86h, 7Fh, 70h, 0F8h, 9Eh, 5Eh, 0FAh, 0D0h, 57h, 0C7h
text:004056FE      db  0FCh, 38h, 0AEh, 64h, 0Ch, 85h, 67h, 3Ah, 0B2h, 9Bh
text:004056FE      db  7, 0F9h, 0EAh, 0ACh, 0C9h, 0BAh, 8Bh, 67h, 65h, 0D9h
text:004056FE      db  0D2h, 96h, 0CDh, 3Dh, 0D9h, 0B7h, 0FEh, 83h, 0F9h, 3Eh
text:004056FE      db  0B4h, 0E9h, 82h, 9Bh, 82h, 66h, 0CEh, 49h, 2Dh, 7Fh
text:004056FE      db  11h, 8Eh, 0F3h, 0DEh, 0FEh, 72h, 4Bh, 37h, 32h, 38h
text:004056FE      db  0ADh, 0B2h, 0EEh, 3Dh, 17h, 62h, 53h, 63h, 3Dh, 28h
text:004056FE      db  89h, 0FCh, 0Dh, 34h, 60h, 0ACh, 45h, 68h, 28h, 92h
text:004056FE      db  51h, 36h, 0D6h, 86h, 5Dh, 7Ch, 6Ch, 46h, 31h, 43h, 35h
text:004056FE      db  5Ch, 0EFh, 0F6h, 0C3h, 76h, 2Eh, 3Ch, 75h, 32h, 63h
text:004056FE      db  5Fh, 0A6h, 7, 99h, 57h, 15h, 0E0h, 0FDh, 0F5h, 4Dh
text:004056FE      db  7Bh, 0E3h, 65h, 7Fh, 1Dh, 0B7h, 8Bh, 65h, 0E8h, 0FFh
text:004056FE      db  75h, 98h, 0E8h, 3Fh, 10h, 2 dup(0)
text:004057FC

```

Figure 5-8

5.1.3 Automatically protection to local executable files by using "Command line"

Virbox Protector provide 2 ways to developer to protect their local executable application:

Using GUI to "select and click" way, which is most easier way to developer to protect their application;

For Some developer who has rich experience in protection, they may more prefer to use command line to protect the critical functions in their application.

5.1.3.1 Generating & Using Map file

For the compiled application, The function will be shown and listed with the "address" when Virbox Protector parse the functions of PE program which is not easy to developer to identify and select these functions with "address" , and The function will be shown and listed with function name if map file available, so using the "map" file will more convenient to developer to select the functions to be protected when use the Virbox Protector to protect their application, here we brief how to generate the "map" file with different language.

5.1.3.1.1 Generate Map file for BCB Program

BCB: Borland C++ Builder, here briefing how to use C++ Builder to generate the map file.

Project settings as shown below:

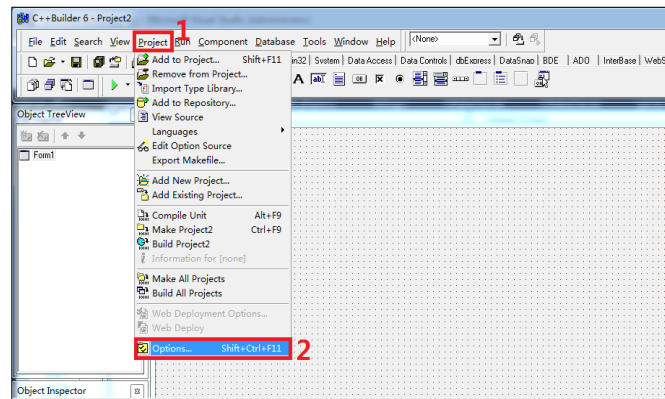


Figure 5-9

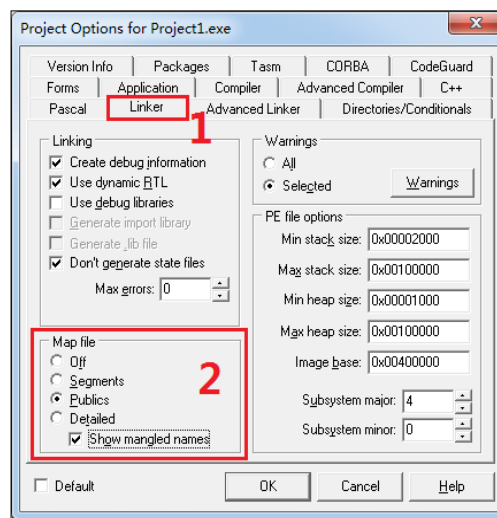


Figure 5-10

5.1.3.1.2 Generate map file for VC program

Project settings as shown below:

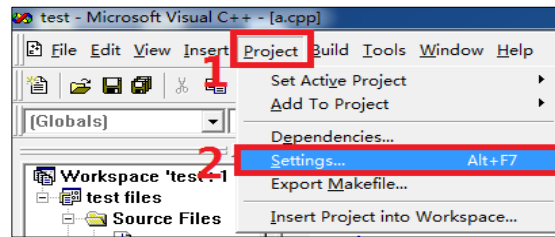


Figure 5-11

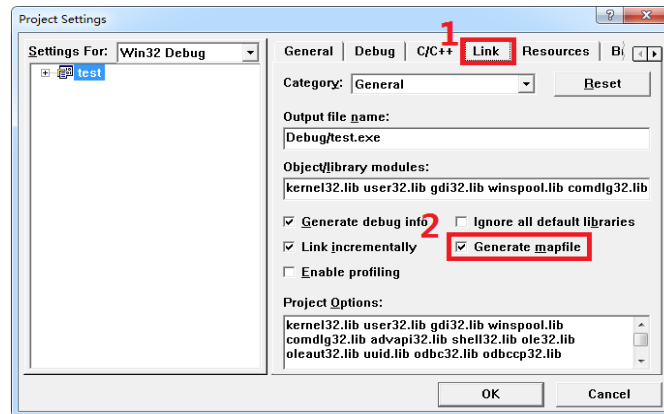


Figure 5-12

5.1.3.1.3 Generate map file for VS program

Project settings as shown below:

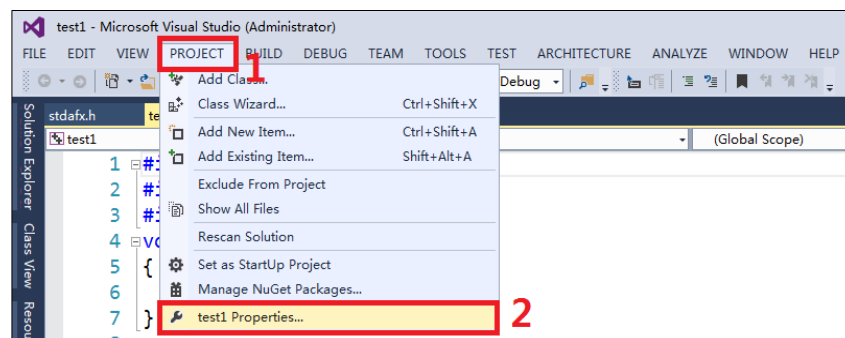


Figure 5-13

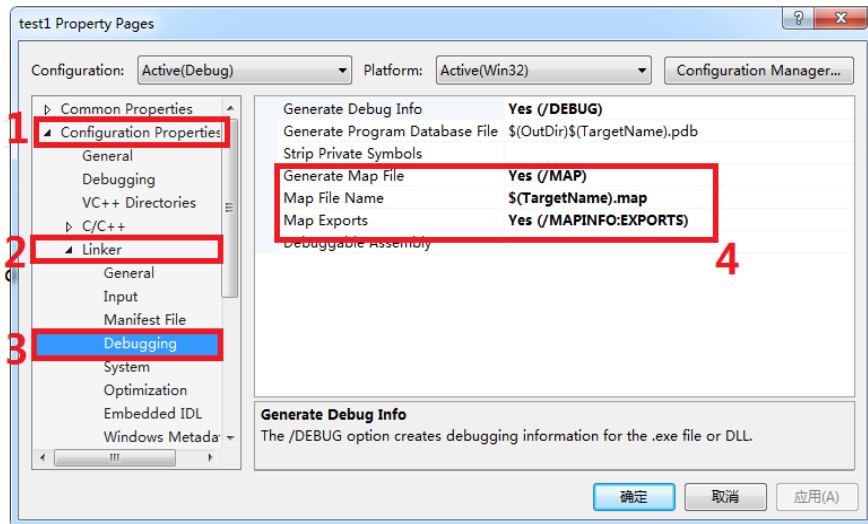


Figure 5-14

5.1.3.1.4 Generate map file for Delphi program

Project settings as shown below:

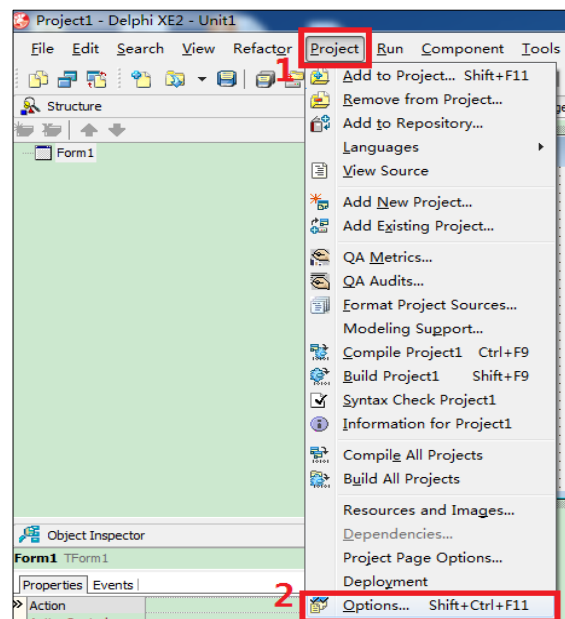


Figure 5-15

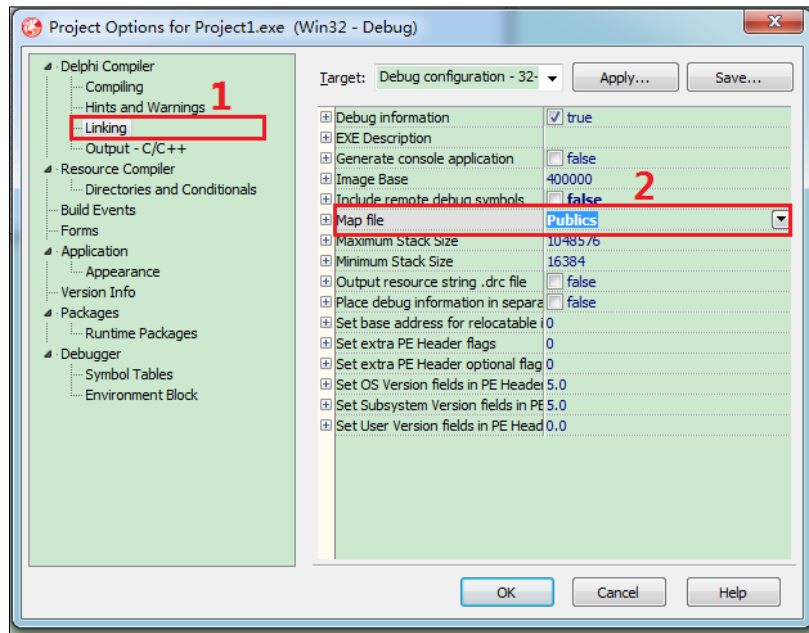


Figure 5-16

5.1.3.1.5 Generate a map file for vb6.0 program

Add a "LINK" value to the system environment variable. The value is "/MAP". Restart the computer. This compiles and generates the exe program. The map file will not be automatically deleted, but will be retained.

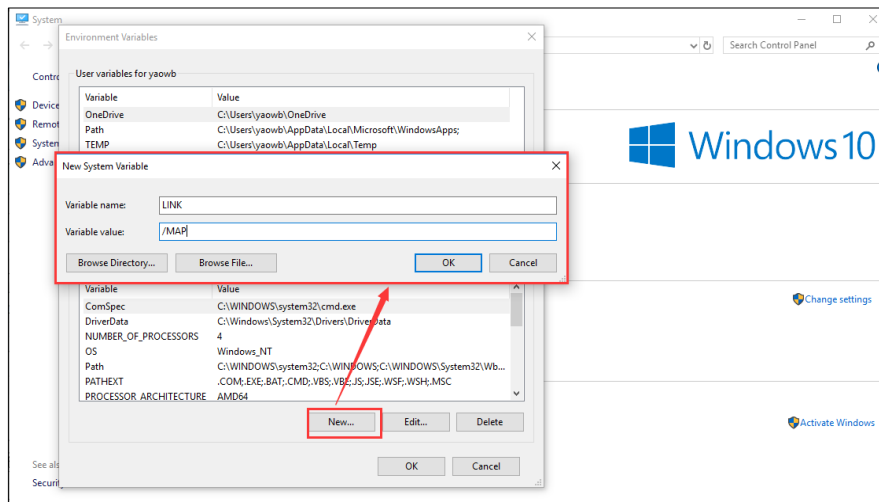


Figure 5-17

5.1.3.2 Using the SDK label API to mark the critical functions

This function and related API helps developer to mark and locate the important function that need to be protected by using SDK label API. For the program is written by C++ language, if you want to find the important

function to protect it later. And only the address of the function will be shown which makes difficult to locate the function you want to protect it. So this label API will help you find the function you want to protect, and no need to find the function in thousands lines of code.

The Header file, static library and dynamic library to SDK label API can be found in the Virbox Protector SDK Kit. The software developer can load the label API statically into the function protected. Then the Virbox Protector can locate the critical code and protect those functions.

5.1.3.2.1 Using SDK Label and Protect your functions

Till now Virbox Protector supports to add these kind of label to the functions:

VBProtectBegin: normal protection

VBVirtualizeBegin: Virtualization protection

VBMutateBegin: Obfuscation protection

VBSnippetBegin: Code snippet protection (Code fragmentation)

VBProtectDecrypt: License encryption and decryption

5.1.3.2.2 Notes:

1. SDK label can only be loaded statically and not supported to load dynamic link lib(LoadLibrary);
2. The String parameter imported by VBProtectBegin, VBVirtualizeBegin, VBSnippetBegin, VBMutateBegin can't be shared with other functions.
3. Make sure the imported parameter to be ASCII code, then the right function name would be shown, otherwise it will show messy code and unreadable.
4. Every begin will follow and match an end, use the "begin" and "end" in pair, and only one pair is allowed in one function.
5. If the protection mode marked in the label inconsistent with the protection mode saved in the project file. , the system will use the protection mode saved in your project file.
6. The code in between the "Begin" and "End" is better to more than 3 lines. To make sure the protected code will be shown in the GUI of Virbox Protector. (it will not be shown in the Virbox Protector for the instruction less 15 bytes.
7. SDK provides 32bit and 64bit dll, you do need to use these libs accordingly.
8. Not support Java, Unity3D.
9. Begin/end do not support nesting use.

10. VBProtectDecrypt, the length of the encrypted string or buffer should be multi times of 16, i.e.: `char g_test_string[16] = {"test_decrypt"};`
11. VBProtectDecrypt, the input buffer and output buffer can't be the same buffer.
12. VBProtectDecrypt, the buffer input need to be put outside of the function, which means is a global variable. For detail how to use, please refer demo.
13. .Net program, is not supported by VBProtectDecrypt currently.

5.1.3.2.3 How to encrypt and decrypt the string by SDK

1. The encrypted string must be a constant value.
2. VBDecryptData also can be used to encrypt and decrypt the data, but the length and the data should be constant value.
3. The following type of string is supported:
 - String encryption:
`VBDecryptStringA("test_string");`
 - Local static variable:
`static const char g_string[] = "test_string";`
 - global variable:
`char g_test_string[] = "test_string";`
`const char g_test_string[] = "test_string";`
`static const char g_test_string[] = "test_string";`

If the program to encrypt is too complicated and the data to be encrypted can't be parsed, it will report error when you protect the software, we recommend you make the program less complicate. Usually it mostly happened to the Linux program which compiled with `-fpic` or `-fpie` with `O2`.

The compiler may merge the same constant string to be one string, if only one of these string is encrypted, an error would be reported:

For example:

```
const char* a = "test_string";  
const char* b = VBDecryptStringA("test_string");  
printf("a = %s, b = %s\n", a, b);
```

For this case, messy code would be shown when you print string "a".

5.1.3.3 Generate .ssp configuration file

.ssp File is the configuration file which need to be used for protected software, here we introduce how to generate the .ssp file.

Generate ssp file manually:

Drag in the program into Virbox Protector, after you completed the configuration of software protection option, the ssp configuration file will be generated in the same path with the program protected after you clicked “Save the selected configuration” or “Save All Configuration”.

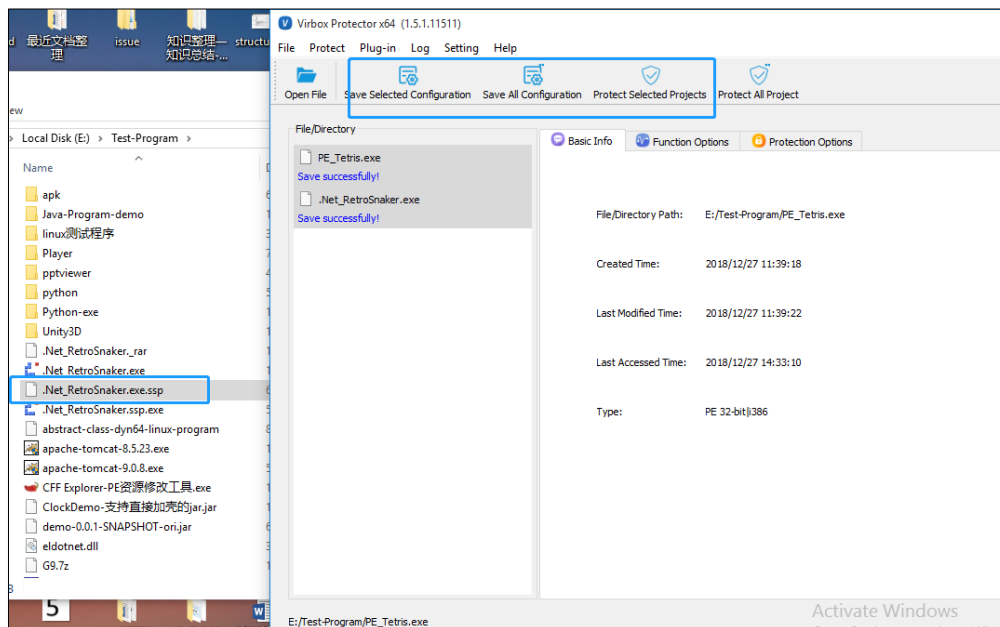


Figure 5-18

5.1.3.4 Protect software with command line

Command parameter :

Command	Description (Local Dongle)	Description(Cloud Lock)	Note
filename	The files that need to be protected/encrypted		/
-u3d	Protect/Encrypt the Unity3D program		
-o output	The output directory of the protected/encrypted program		

5.1.3.4.1 Using Command line to protect the application in Linux Environment

5.1.3.4.1.1 Normal application protection/encryption

Command line in Linux system:

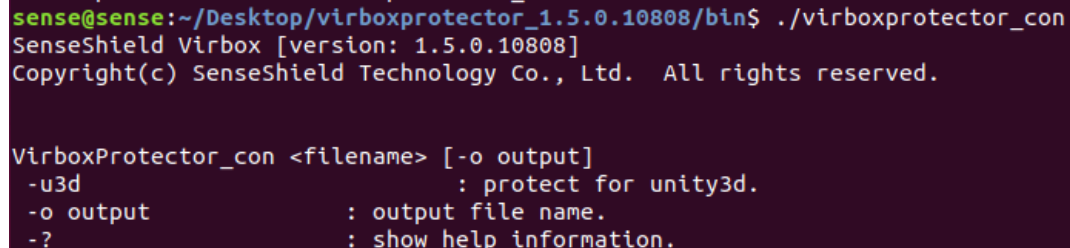
For the protection/encryption of normal application in Linux environment, here we take the Linux platform program as an example:

Use the Virbox Protector GUI tool to generate configuration files (optional)

- If a configuration file is generated, you can select the function number and protection mode in the GUI tool
- If no configuration file is generated, only the default entry function will be protected

Open a terminal in linux platform, enter the path where "virboxprotector_con" is located, and enter "virboxprotector_con" to run Virbox Protector

Help information can be viewed:



```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

VirboxProtector_con <filename> [-o output]
-u3d                      : protect for unity3d.
-o output                  : output file name.
-?                          : show help information.
```

Figure 5-19

For the programs executed in different platforms, the Virbox Protector has different edition & license. You need to contact Virbox team to get the corresponding license.

Command: Path of VirboxProtector_con path of the program to be protected -o path of output file

- If the license is not verified, when you use Virbox Protector to protect program, it will prompt "Can not find the license", as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/virbox_test/cb_bytes_test' -o '/home/sense/Desktop/virbox_test/cb_bytes_test.ssp.vp'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

loading cb_bytes_test ...
Error (13000020): Can not find the license.
```

Figure 5-20

- After the license have been obtained, you can protect the program with Virbox Protector successfully, as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/virbox_test/cb_bytes_test' -o '/home/sense/Desktop/virbox_test/cb_bytes_test.ssp.vp'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

loading cb_bytes_test ...
link...
save as cb_bytes_test.ssp.vp ...
Succeed.
```

Figure 5-21

5.1.3.4.1.2 Using Command Line to protect the Unity3D program

Unity3D, as a special file type, the protection methods is different from the normal program. For the Unity3D program for Windows, Linux and macOS platforms, the entire directory of Unity3D needs to be protected; for Unity3D for the Android platform, the apk of Unity3D program needs to be protected. Here we take a Linux Unity3D as an example:

- Use the Virbox Protector GUI tool to generate configuration files (optional)
- Open a terminal window, enter the path where "virboxprotector_con" is located, and enter "virboxprotector_con" to run Virbox Protector. Help information can be viewed.
- For the programs in different platforms, the Virbox Protector need to verify the license in different platform. You need to contact Virbox team to obtain the corresponding license.

Command: Path of VirboxProtector_con path of the program to be protected -u3d -o Path of output file

If no license has been verified, when you run Virbox Protector, it will prompt "**Can not find the license**", as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particles2018.1.9f1'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d Particles2018.1.9f1 ...
Error (13000020): Can not find the license.
```

Figure 5-22

After the license is verified, the program can be successfully protected by Virbox Protector, as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particles2018.1.9f1'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d Particles2018.1.9f1 ...
Succeed.
```

Figure 5-23

5.1.3.4.2 Using Command line to protect executables in Windows environment

5.1.3.4.2.1 Take Windows application as an example

Use the Virbox Protector GUI tool to generate configuration files (optional)

- If a configuration file is generated, you can select the quantity of protected function and protection mode in the GUI tool
- If no configuration file is generated, only the entry function on default will be protected

Open terminal in window from the start menu, input:

Virbox Protector_con.exe

To start execution.

This command can get the help info.

Command line help info:

VirboxProtector_con <filename> [-o output]

-o output : output file name.

-? : show help information.

```
C:\Program Files\senseshield\Virbox Protector Standalone\bin>virboxprotector_con.exe
SenseShield Virbox [version: 1.4.2.9353]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

VirboxProtector_con <filename> [-o output]
-o output          : output file name.
-?                 : show help information.

C:\Program Files\senseshield\Virbox Protector Standalone\bin>virboxprotector_con.exe -?
SenseShield Virbox [version: 1.4.2.9353]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

VirboxProtector_con <filename> [-o output]
-o output          : output file name.
-?                 : show help information.
```

Figure 5-24

When you use the command without license, it will have this error report: "Can not find the license".

Difference license of Virbox Protector will be required when you are protecting the program on different platform. Please contact Virbox Team for the corresponding license.

Command: path of *Virbox Protector_con.exe* the path of the program that to be protected -o path of output file.

```
D:\test\Virbox Protector Standalone\bin>virboxprotector_con.exe PE_Tetris.exe PE-protected.exe
SenseShield Virbox [version: 1.4.2.9353]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

loading PE_Tetris.exe ...
License login failed, error code: 0x13000030(No logged user)!
Error (318767152): Unknown error

D:\test\Virbox Protector Standalone\bin>
```

Figure 5-25

After the license is verified, you can complete protection.

```
VirboxProtector_con <filename> [-o output]
-o output          : output file name.
-?                : show help information.

D:\test\Virbox Protector Standalone\bin>virboxprotector_con.exe PE_Tetris.exe PE-protected.exe
SenseShield Virbox [version: 1.4.2.9353]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

loading PE_Tetris.exe ...
link...
save as PE_Tetris.ssp.exe ...
Succeed.

D:\test\Virbox Protector Standalone\bin>
```

Figure 5-26

5.1.3.4.2.2 Unity3D program protection:

- Use the Virbox Protector GUI tool to generate configuration files (optional)
- Open a terminal window, enter the path where "virboxprotector_con.exe" is located, and enter "virboxprotector_con.exe" to run Virbox Protector. Help information can be viewed.
- For the programs of different platforms, Virbox Protector has different license. You need to contact Virbox team to obtain the license.

Command: path of *Virbox Protector_con.exe* the path of the program that to be protected *-u3d -o* path of output file.

If no license have been found, when you protect the software it will show:

```
C:\Users\test\Desktop\virboxprotector_standalone_1.4.2.10236_windows_x64\bin>virboxprotector_con.exe C:\User
s\test\Desktop\sample\angrybots5.5.3.apk -u3d -o C:\Users\test\Desktop\sample\ssp.angrybots5.5.3.apk
SenseShield Virbox [version: 1.4.2.10236]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d angrybots5.5.3.apk ...
Error (13000020): Can not find the license.
```

Figure 5-27

If the license is successfully found and protect the program successfully:

```
C:\Users\test\Desktop\virboxprotector_standalone_1.4.2.10236_windows_x64\bin>virboxprotector_con.exe C:\User
s\test\Desktop\sample\angrybots5.5.3.apk -u3d -o C:\Users\test\Desktop\sample\ssp.angrybots5.5.3.apk
SenseShield Virbox [version: 1.4.2.10236]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d angrybots5.5.3.apk ...
Succeed.
```

Figure 5-28

5.2 Protect the .Net application

Drag the execute file you want to protect into Virbox Protector:

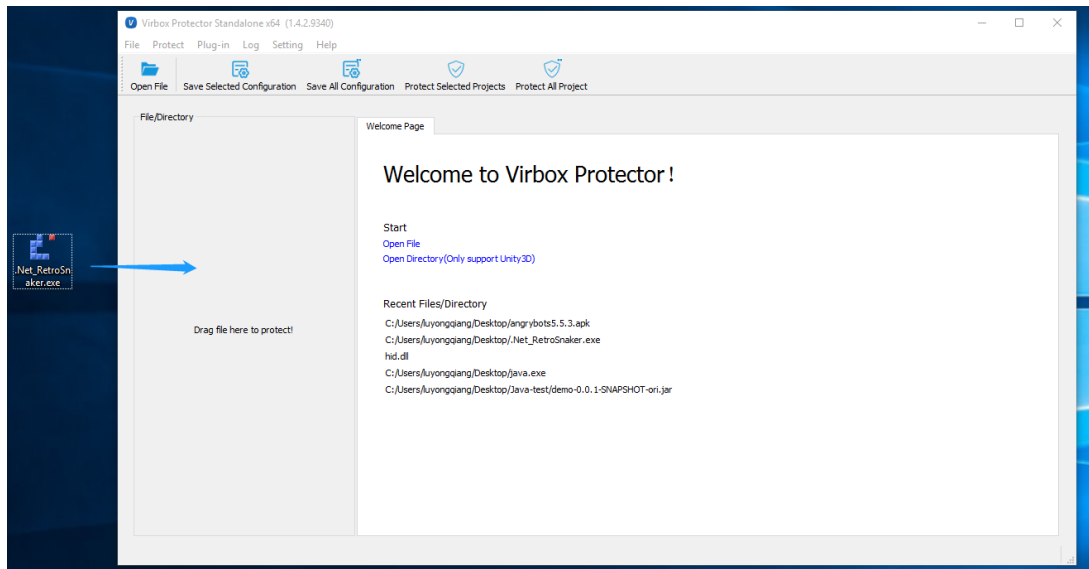


Figure 5-29

5.2.1 Protect the .NET application in fundamental

Protection Option Tag to .NET application as shown below:

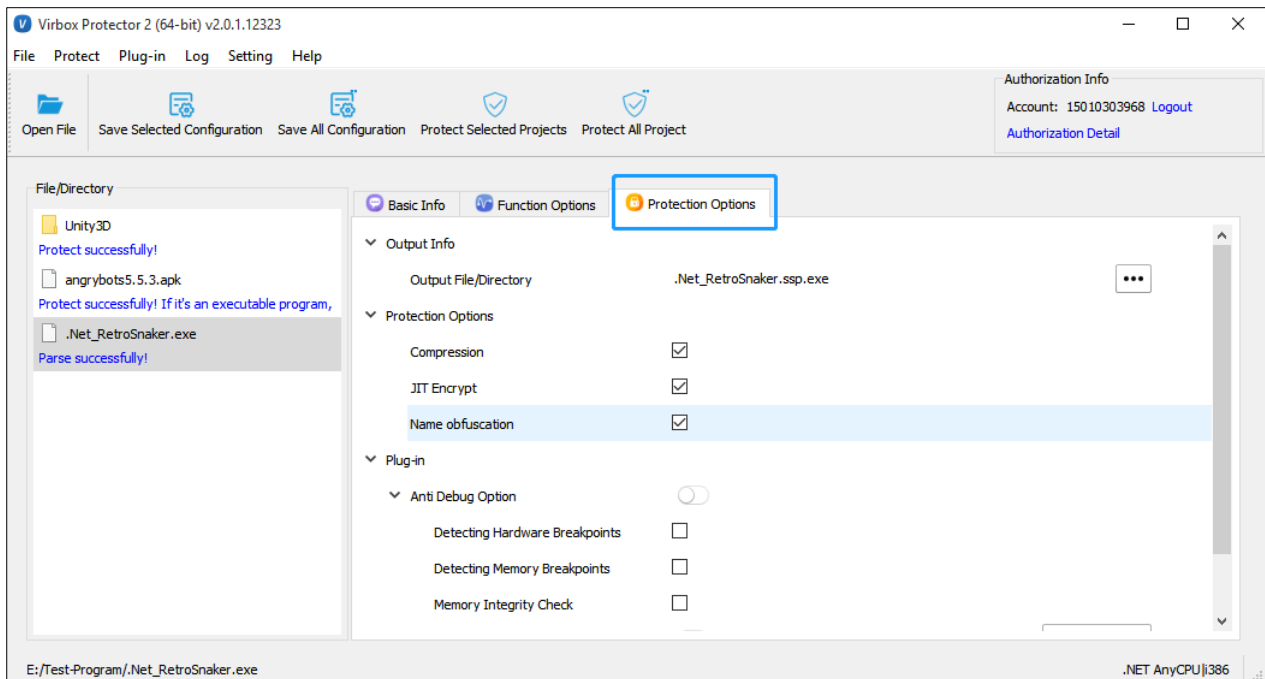


Figure 5-30

5.2.1.1 Name Obfuscation

Rename the .Net program method name and class name with random string, the name that exported for external call would not change.

Protection Comparison:

Before protection:

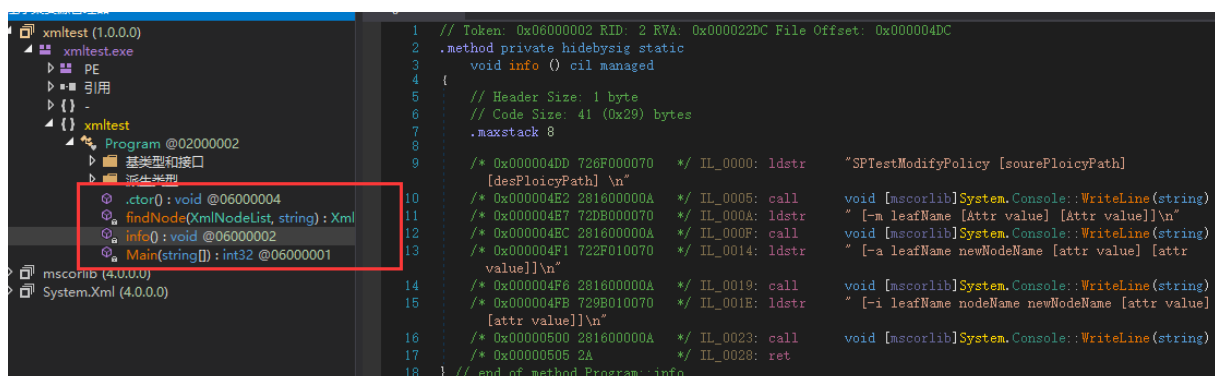


Figure 5-31

After Protection:

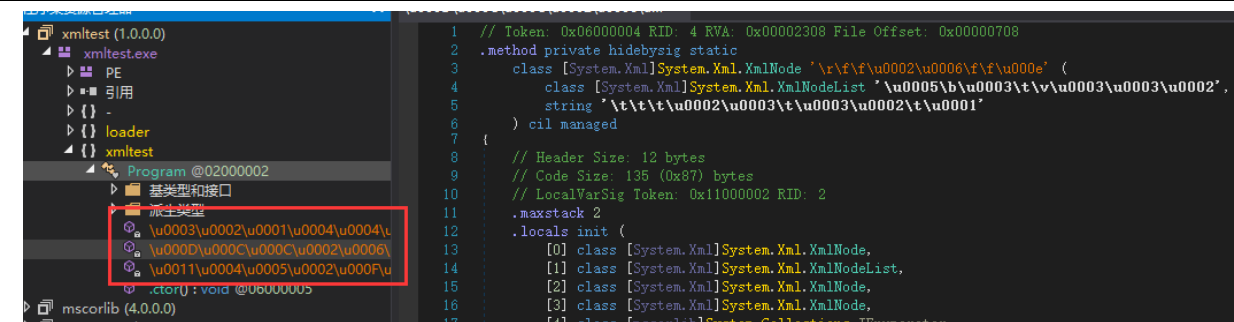


Figure 5-32

5.2.1.2 Compression

The main purpose of the compression is not only to compress the software, it will encrypt the code and the data segment and hide the original import table and relocate the information of the protected application, and compressed the original data size at same time.

The compression will protect the .Net program overall, in this way to protect the method being anti-compiled by DnSpy, ILSpy and .Net Reflector. This provide better compatible by using IL enveloper code.

Protection Mechanism

This function will pack the original data segment with data package and compress, replace the Original Entry Point (OEP) with the packer code. The data segment and code segment will be retrieved when the program is executed, and relocate it to execute the program.

The Purpose:

Prevent the static anti-compile and prevent the program being patched

The Benefit:

1. Hide the code, data and file structure information of the program, protect the software in general.
2. High efficiency when the program executed, and small impact to performance when the program is loaded.

The Weakness:

1. When the pack code is executed, the code segment and data segment may be retrieved and be dumped.

Protection Comparison:

Before Protection:

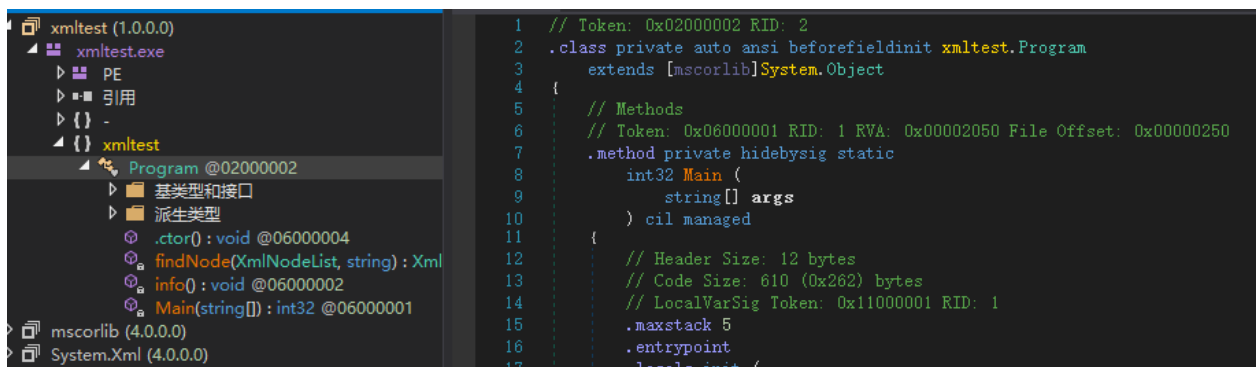


Figure 5-33

After protection:

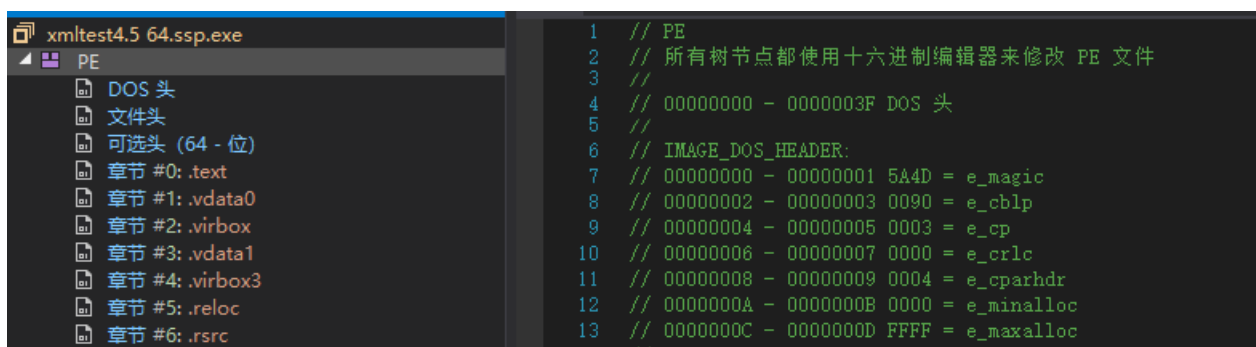


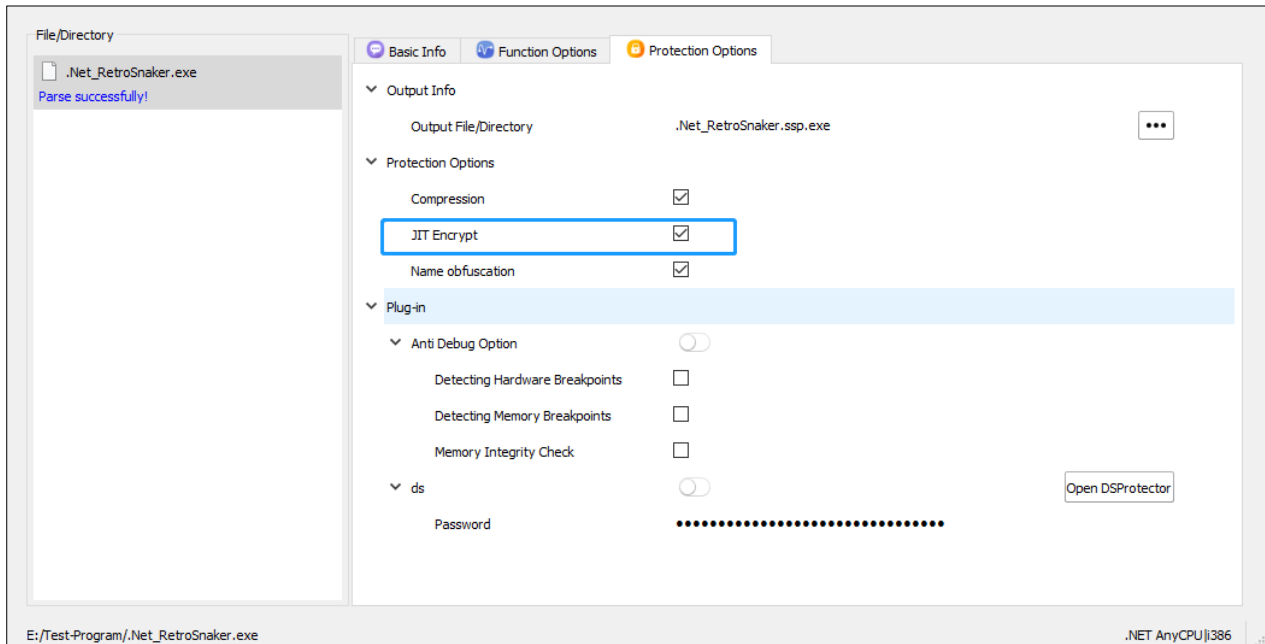
Figure 5-34

5.2.1.3 JIT encryption

- .Net JIT encryption will encrypt all of the method IL instruction of the .Net Program, and only in the JIT compile process of the .Net Virtual Machine the instruction will be decrypted, This can be used to prevent static anti-compile and prevent the IL code being Dumped in memory.
- JIT encryption will encrypt all of the method in default and enhance the security level of the source code after protection.
- JIT encryption .support inheritance, event, reflection, recursive call which is not supported in general encryption solution.

Usage Guide:

Drag the .Net program into Virbox Protector, the “Protection Option” will show “JIT encryption” option. You need to select this option if you want to use this function to protect your program.



5.2.1.4 Remove Strong Name

1. StrongName provides the .Net program with a mechanism which adding the label of version and label of the original author information.
2. StrongName can be used to help the software user to verify if the program comes from the original author and not be modified (prevent tampering).
3. So the software developer need to remove the strong name before protection/encryption and add the Strong name after protection/encryption again.

5.2.2 Protect the critical Functions

Function Option tag:

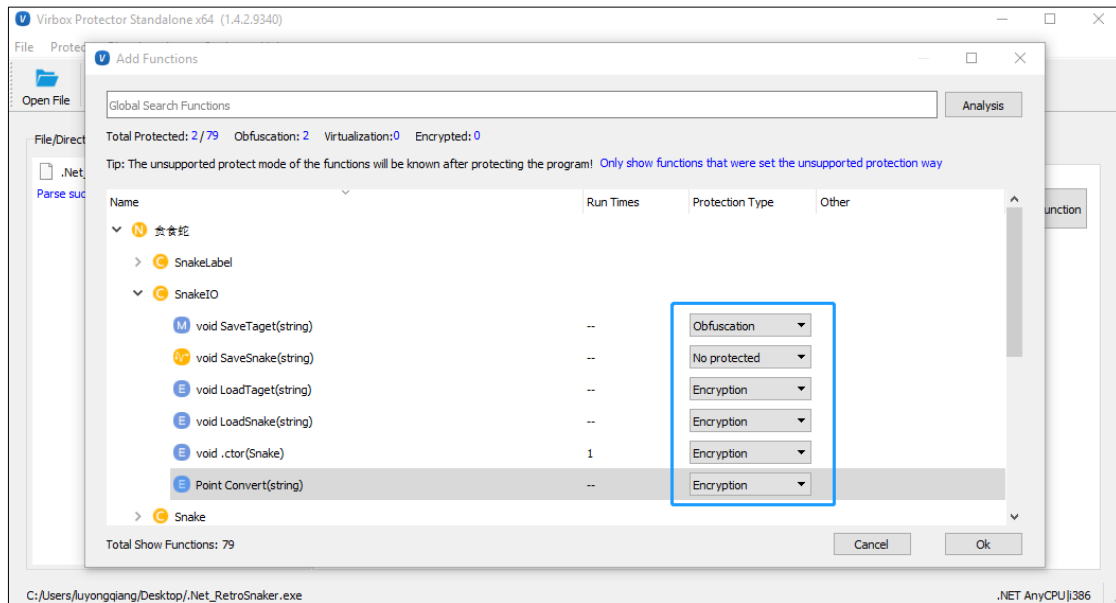


Figure 5-35

Function Level Protection:

5.2.2.1 Code encryption (.Net)

Protection Mechanism

Code encryption is dynamic code protection technology, the original method byte code will be encrypted and decrypted only when the method is executed.

The Purpose:

Prevent the program from being unpacked and being dumped

The Benefit:

Almost no impact to software performance.

The Weakness:

The method of the program may be analyzed when decrypted to execute.

Protection Comparison:

Before protection:

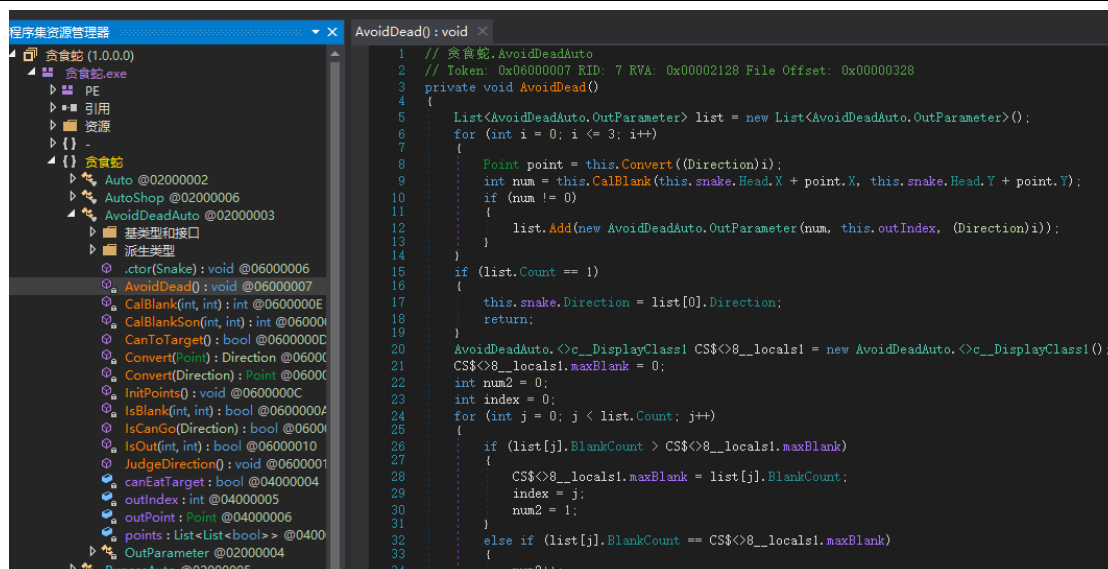


Figure 5-36

After protection:

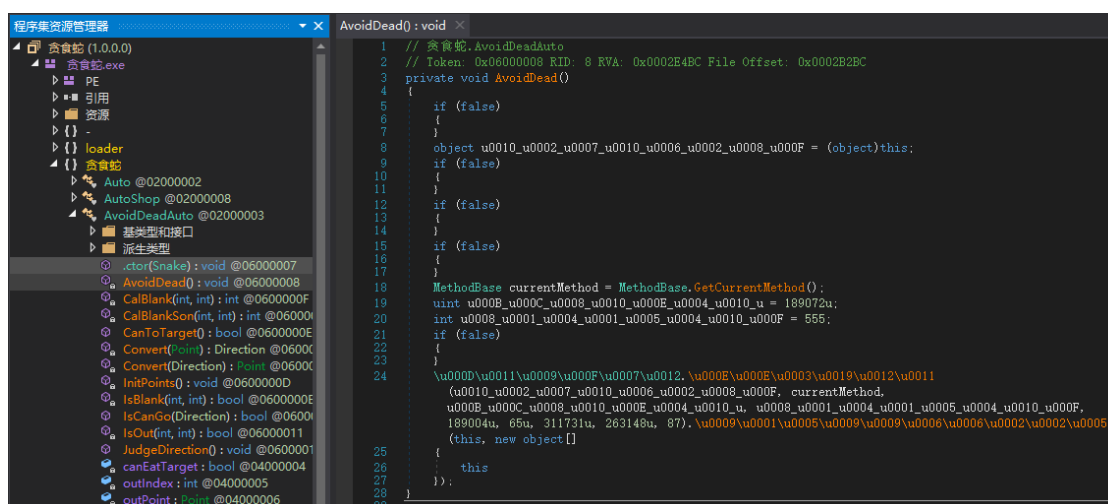


Figure 5-37

Following scenario is not supported to code encryption:

For C# program, maybe the message of "Part of the function has been set the unsupported protection technology, and please modify the protection option you used. Error code: 0xA000A000"

Following scenario is not supported to "code encryption":

1. The non-static method of Value type: System.ValueType (and their Inherited class)
2. Generic methods are not supported

3. C++ .net not supported
4. Recursive calling is not supported
5. Variable parameters are not supported
6. Default parameters are not supported

5.2.2.2 Code Obfuscation

Virbox Protector will translate the code instruction into a stream of pseudo-code that neither the machine nor the human can recognize. When the pseudo-code executed, the software will translate and interpret to restore the code into the original code. To let it executable when it is executed.

Virbox Protector supports the obfuscation for x86/arm .net il serial instruct.

Protection Mechanism:

To obfuscate the original instruction and prevent from being static analyzed.

The Benefit:

Prevent from anti-compiled and make the hacker more difficult to analysis the code.

The Weakness:

Negative impact to software performance.

Limited protection to software.

Protection Comparison:

Before protection:

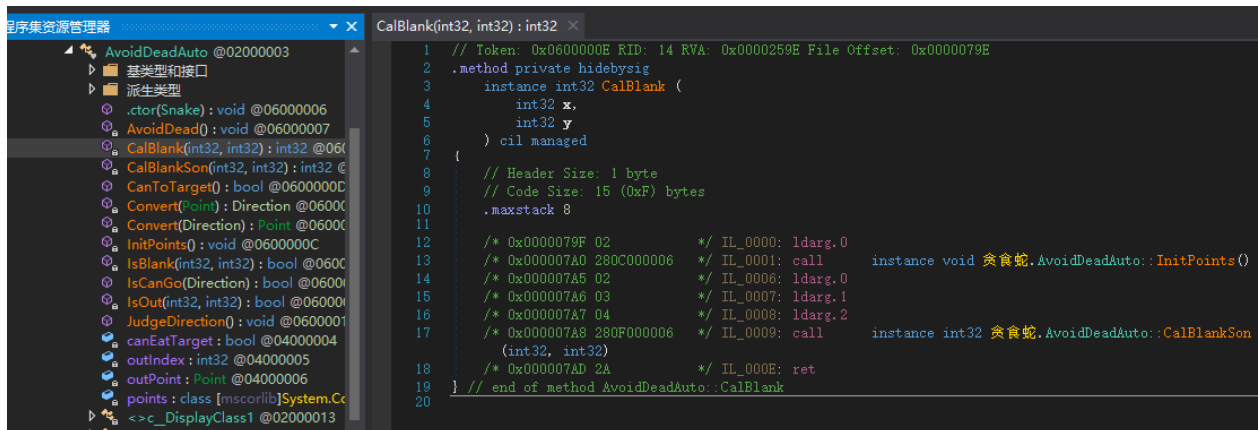


Figure 5-38

After Protection:

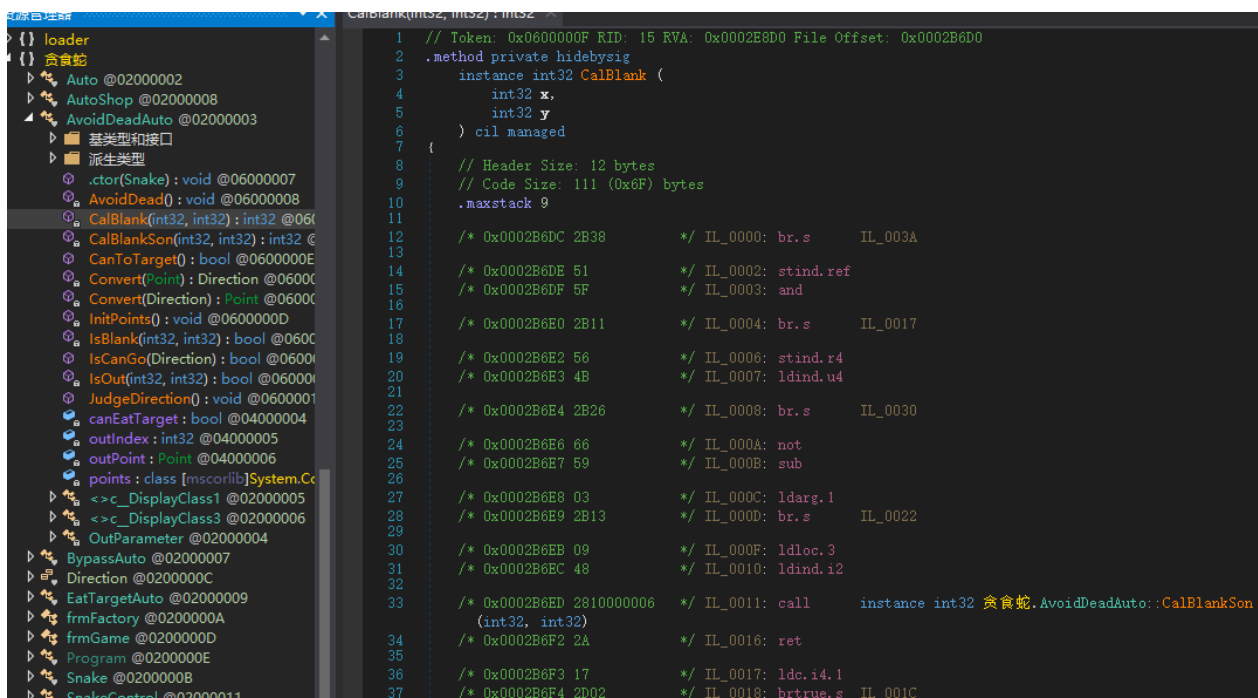


Figure 5-39

You can select the corresponding protection option according to the introduction of the function.

Then click **“Protect all project”** button to complete protection:

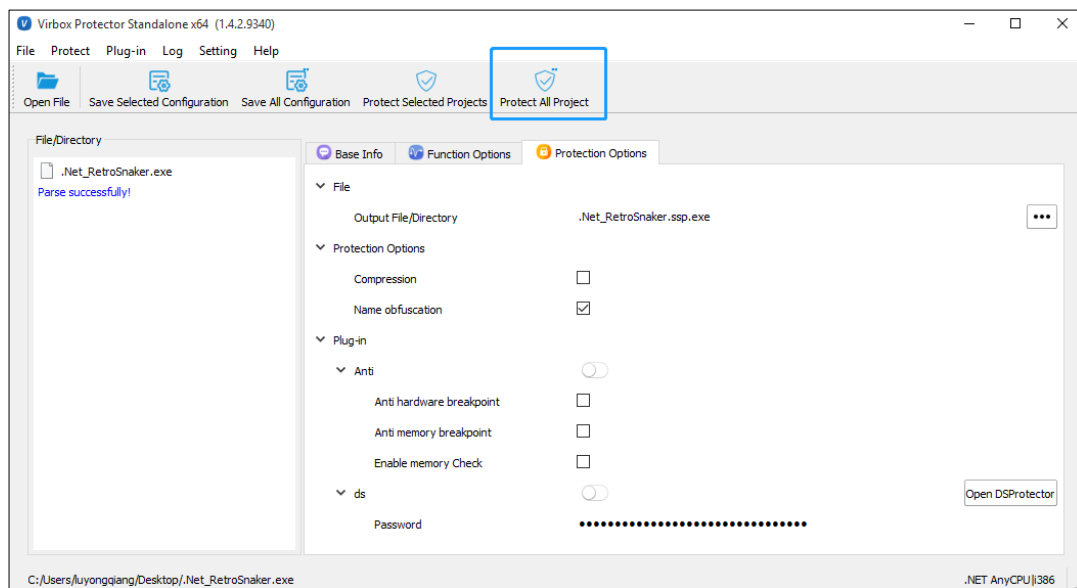


Figure 5-40

5.3 Java Program protection:

Before Java program Protection:

5.3.1 Protection background and introduction

Java program support cross platform operation which rely on the java execute in the Virtual machine environment as intermediate code, the challenge to protect java is: the de-compilation to java code is much easier compare with to de-compilation to other languages. And the decompiled code is almost compatible with the source code after optimization.

There are many Java Obfuscator available in the market to protect Java Application. The mechanism of the Java obfuscator is to obfuscate the compiled code, and makes the decompiled code difficulty to read and understand. And increasing the difficulty to reverse engineering. For the people who familiar to use the de-compilation tool. It is almost transparent. So the security level for the Java application protected by Java Obfuscator is quite limited.

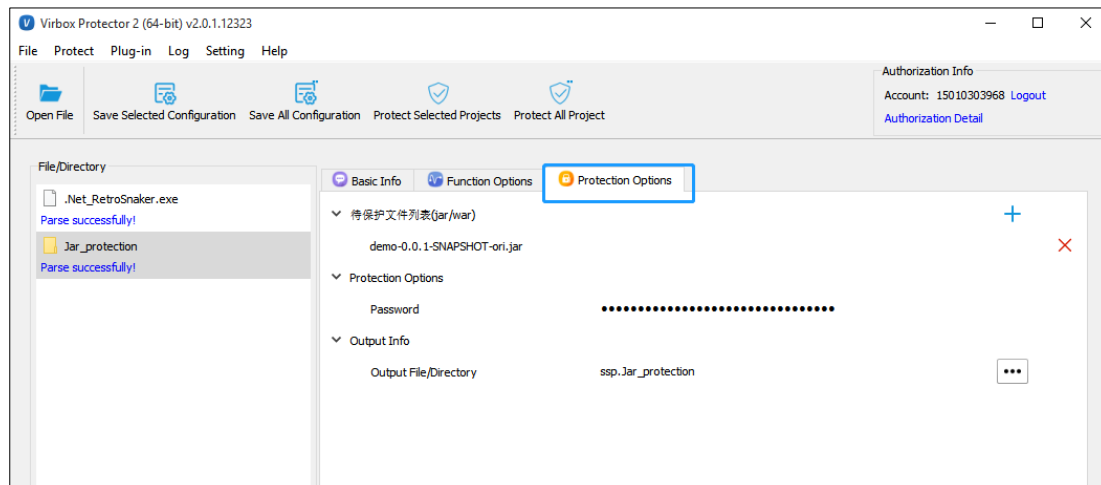
After compile of the Java source code, it is easy to anti-compile the .class file which contains the class name, method name and variable name. The hacker almost can obtain the source code which is completely same with the original source code by anti-compile engineering.

Virbox support directly protect the Jar archive and War archive. Which will encrypt the byte code of every method to prevent the source code from being anti-compiled and it is quite easy to operate with Virbox

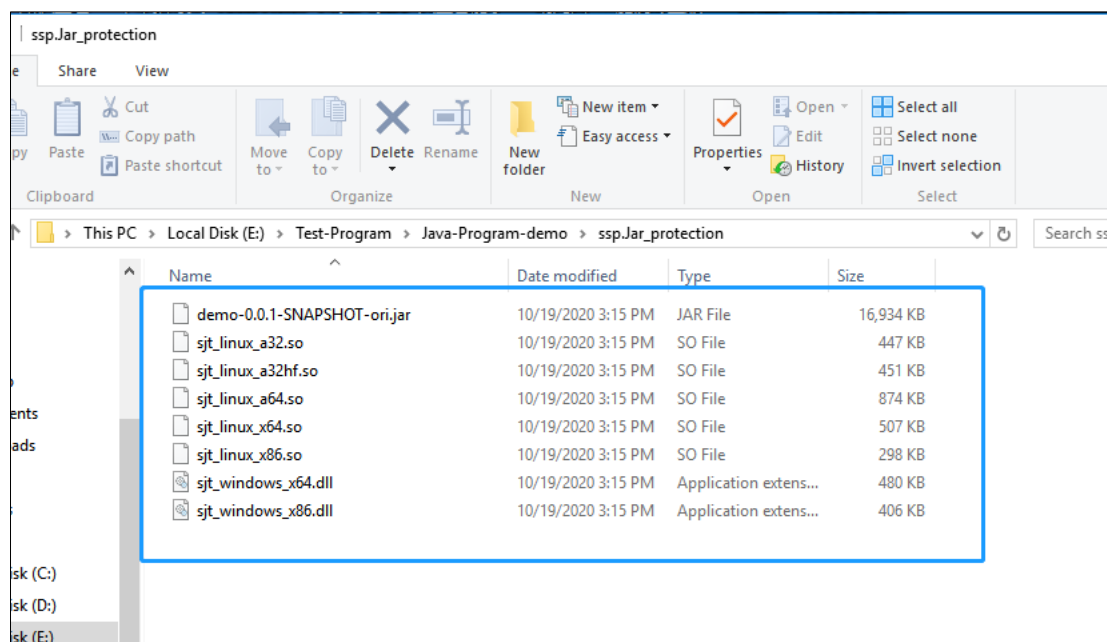
Protector. Currently the source code from Windows, Linux, ARM Linux platform are supported to be protected.

5.3.2 Protect to Jar archive

1. Drag the Jar archive directly into the Virbox Protector:



2. After protection, an encrypted Jar archive and sjt plugin would be generated, like the picture showing:



5.3.2.1 Deployment

Windows:

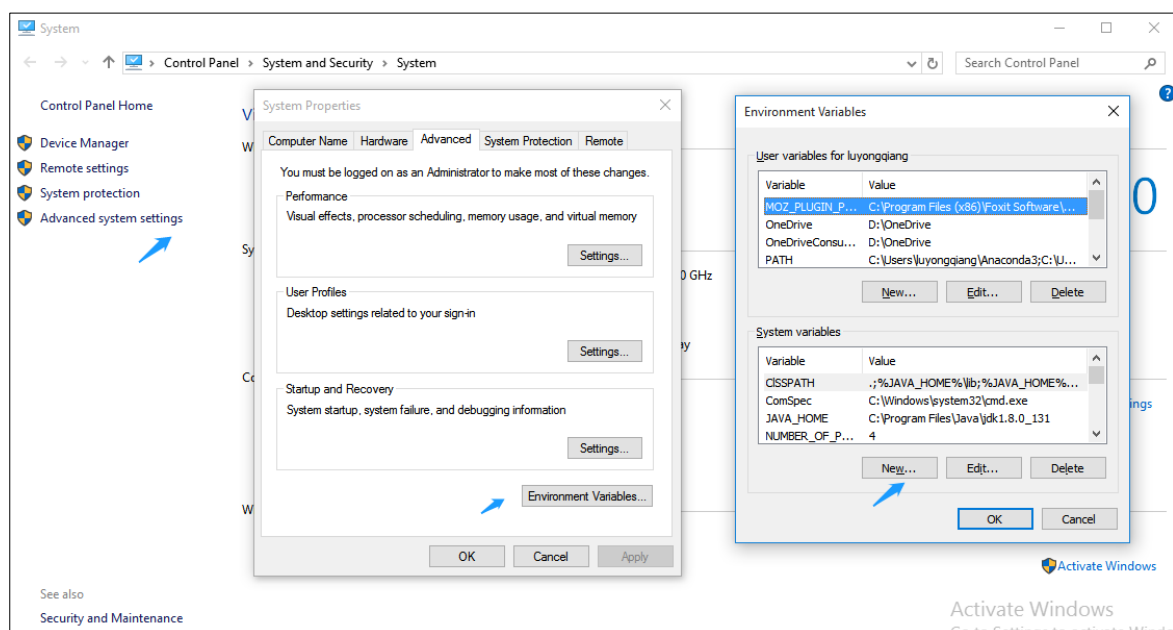
Add the environment variable to the computer:

1. Put the sjt library into a fixed directory, put the "sjt_windows_x64.dll" and "sjt_windows_x86.dll" to the system environment variable.

"This PC" -> "Properties" -> "Advanced system settings" -> "Environment variables" -> "New..."

-> "Environment variable" -> "system variable", new variable: "JAVA_TOOL_OPTIONS", variable value:

"-agentpath:C:\Users\test\Desktop\sjt\sjt_windows_x64.dll"



2. After you configured the environment, you can directly run the Jar archive, or directly call the Jar archive.

Directly run the protected Jar archive

1. If Java version is 64 bit version, you need to assign the "sjt_windows_x64.dll", If the Java version is 32 bit, you need to use "sjt_windows_x86.dll".

If sjt library and the Jar archive are in the same directly, you can directly run the following command in the current Jar archive directly.

Command: **java -agentpath:sjt_windows_x64.dll -jar ***.jar**

2. If sjt library and jar archive is not in the same directly, you need to assign the full directory.

Command: **java -agentpath:C:\Users\test\Desktop\sjt\sjt_windows_x64.dll -jar ***.jar**

Linux System:

- Add ***JAVA_TOOL_OPTIONS=-agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x86.so*** to ***/etc/profile*** environment variable.
- Input: ***Source /etc/profile*** to make the configure file take effect.
- After you have completed the environment configuration, directly run jar archive or call Jar archive.

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_261
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH

# sense sjt
export JAVA_TOOL_OPTIONS="-agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x86.so"
```

Directly run the protected program:

If you are using 64 bit Java, you need to use "***sjt_windows_x64.dll***", for 32 bit java please use "***sjt_windows_x86.dll***";

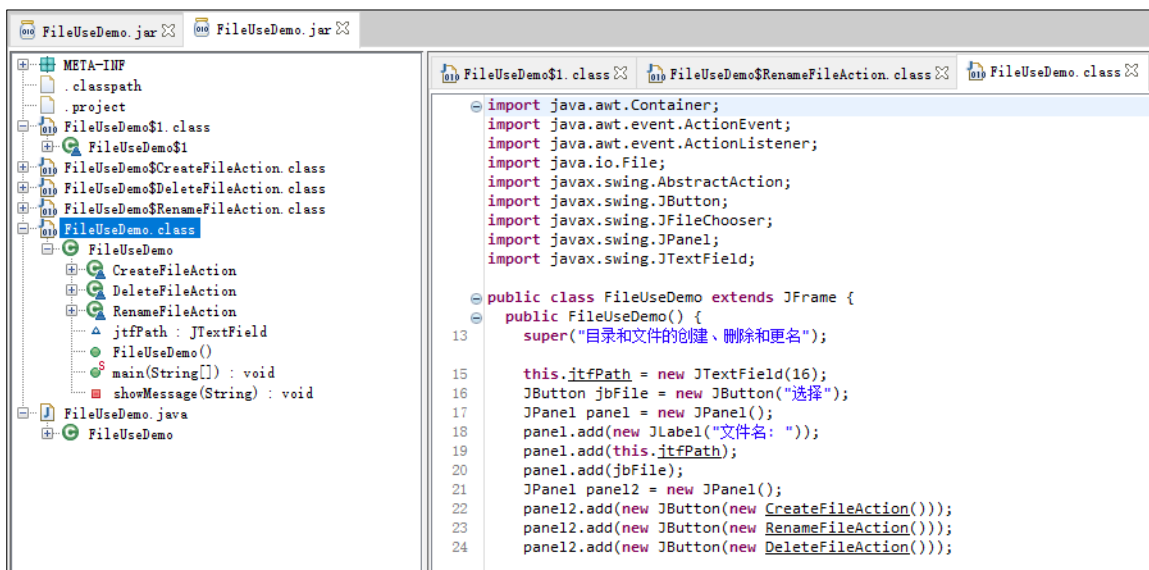
- If the sjt library are in the same directory with Jar archive, you can run the following command in the current directory:
- Command: ***java -agentpath:sjt_windows_x64.dll -jar ***.jar***
- If the sjt library is not in the same directory with the jar archive, you need to assign the complete directory:
- Command: ***java -agentpath:C:\Users\test\Desktop\sjt\sjt_windows_x64.dll -jar ***.jar***

Please noted that after configuration of the environment variable, the Jar archive will use the environment variable even you have assigned the sjt library location.

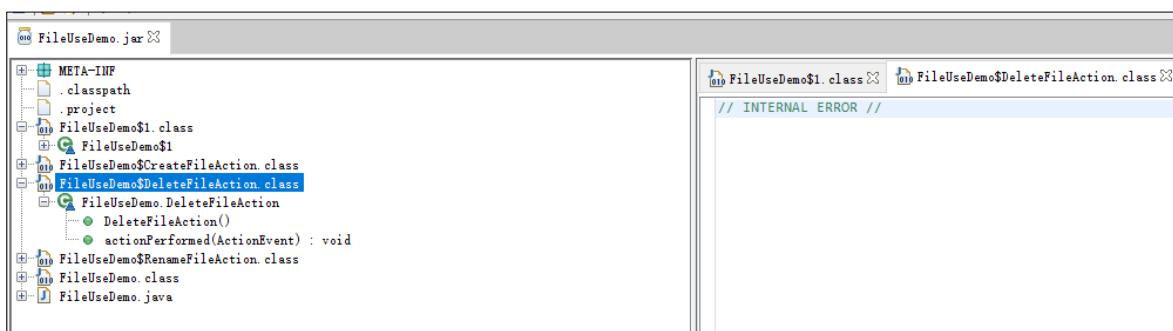
Jar archive Protection performance compare:

(Check the encryption performance by ***jd-gui***)

Before protection:



After protection:



5.3.3 War archive protection:

Virbox Protector will protect the class file in the War archive.

1. Drag War archive into Virbox Protector
2. After successfully encrypted, *sjt* plugin and the protected War archive would be generated.

名称	修改日期	类型	大小
hello.war	2020/9/30 10:36	WAR 文件	4 KB
myhome1.war	2020/9/30 10:36	WAR 文件	26,403 KB
sample.war	2020/9/30 10:36	WAR 文件	5 KB
sjt_linux_a32.so	2020/9/30 10:36	SO 文件	188 KB
sjt_linux_a32hf.so	2020/9/30 10:36	SO 文件	188 KB
sjt_linux_a64.so	2020/9/30 10:36	SO 文件	666 KB
sjt_linux_x64.so	2020/9/30 10:36	SO 文件	218 KB
sjt_linux_x86.so	2020/9/30 10:36	SO 文件	32 KB
sjt_windows_x64.dll	2020/9/30 10:36	应用程序扩展	129 KB
sjt_windows_x86.dll	2020/9/30 10:36	应用程序扩展	110 KB

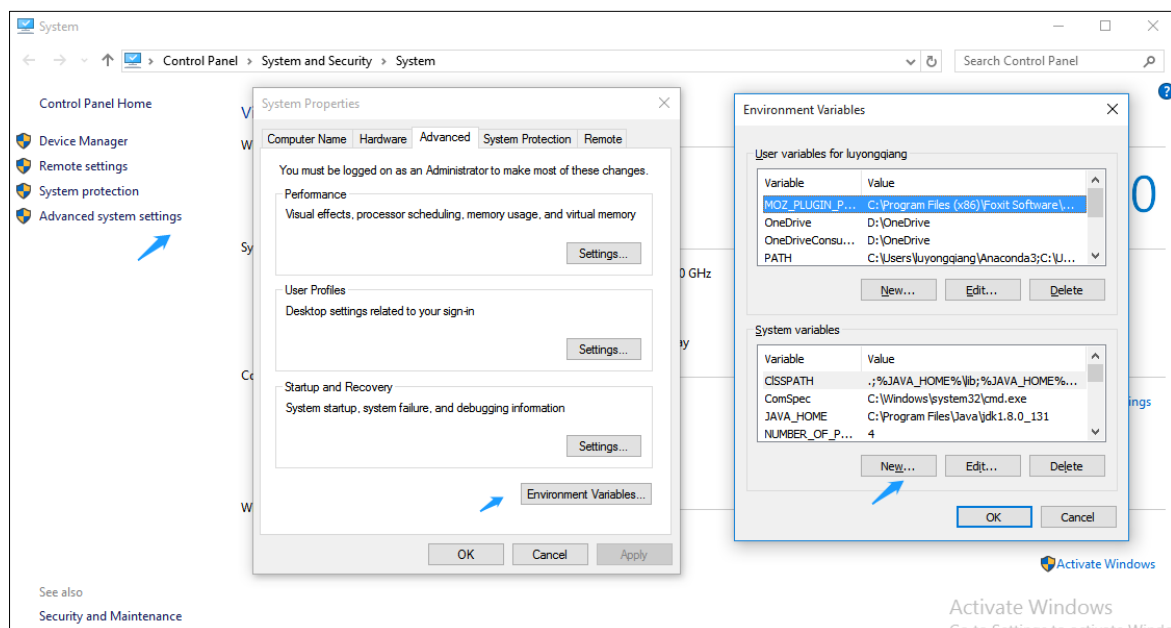
Deployment:

Windows system:

Following 2 ways to configure the system, you can select one of them:

1. Add it to the system variable:

- You can put the sjt library into a fixed directory, and add the **"sjt_windows_x64.dll"** and **"sjt_windows_x86.dll"** to the system environment variable.
- "This PC"->"Properties"->"Advanced system settings"->"Environment variables" -> "New..."**
- "Environment variable"->"system variable", new variable: "JAVA_TOOL_OPTIONS", variable value:**
- "-agentpath:C:\Users\test\Desktop\sjt\sjt_windows_x64.dll"**



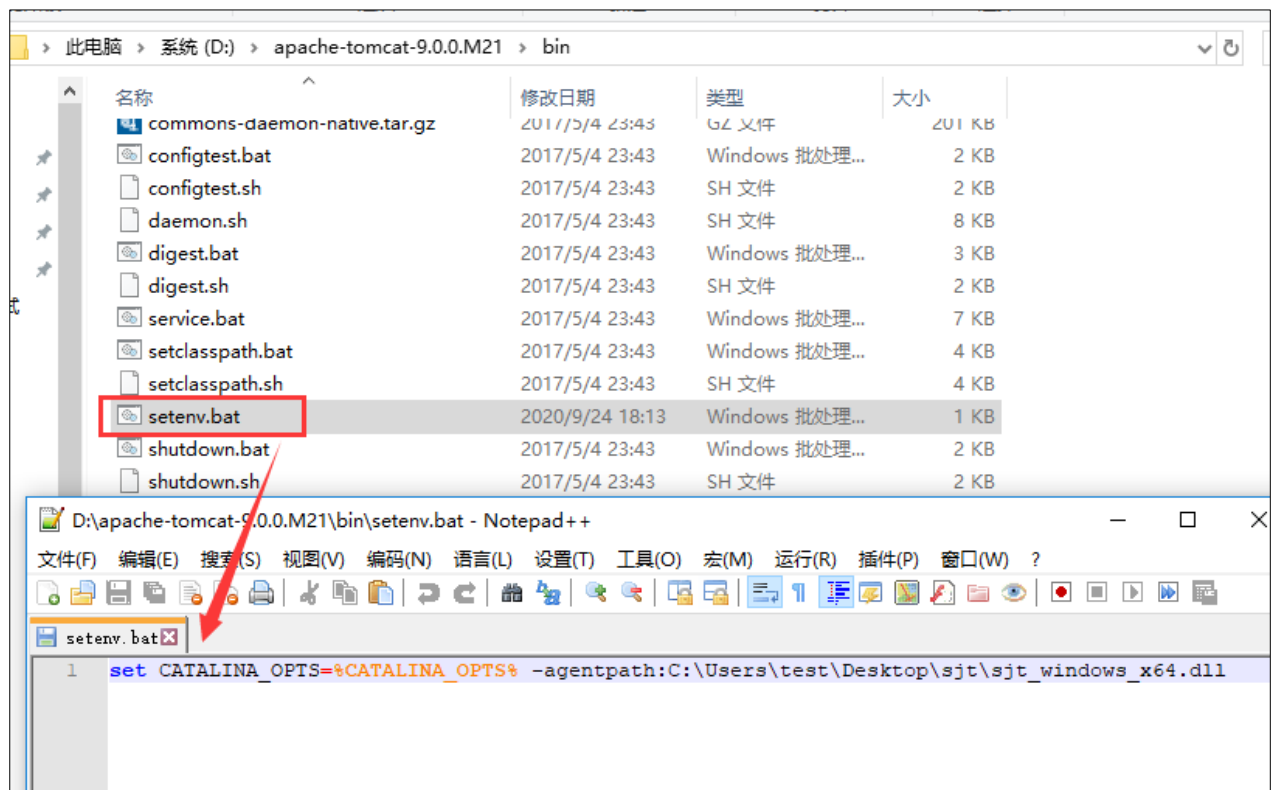
IF you use tomcat to run the protected program, you need to put encrypted archive into this location:
.\apachetomcat\webapps directory, and start tomcat service.

2. Set the setenv.bat in the tomcat directory:

Set the setenv.bat in the tomcat\bin directory:

Set the setenv.bat in the tomcat\bin directory, such as:

- Create “setenv.bat” in the tomcat\bin directory, set the environment variable such as (full path):
Set CATALINA_OPTS=%CATALINA_OPTS% -agentpath:sjt_windows_x64.dll



- Put the encrypted War archive into the location: **.\apache-tomcat\webapps** and start the tomcat service.

Linux System:

Add the library into the /etc/profile environment variable:

- **JAVA_TOOL_OPTIONS=-agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x86.so** into the /etc/profile

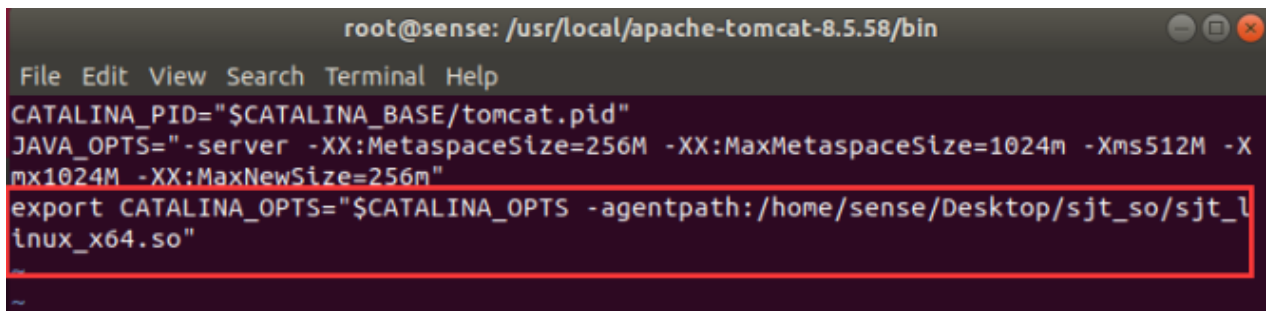
```
export JAVA_HOME=/usr/local/java/jdk1.8.0_261
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH

# sense sjt
export JAVA_TOOL_OPTIONS="-agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x86.so"
```

- Source **/etc/profile** environment variable to make it take effect.
- You can directly start the tomcat service after you have configured the environment and put the War archive in the directory: **./apache-tomcat/webapps**

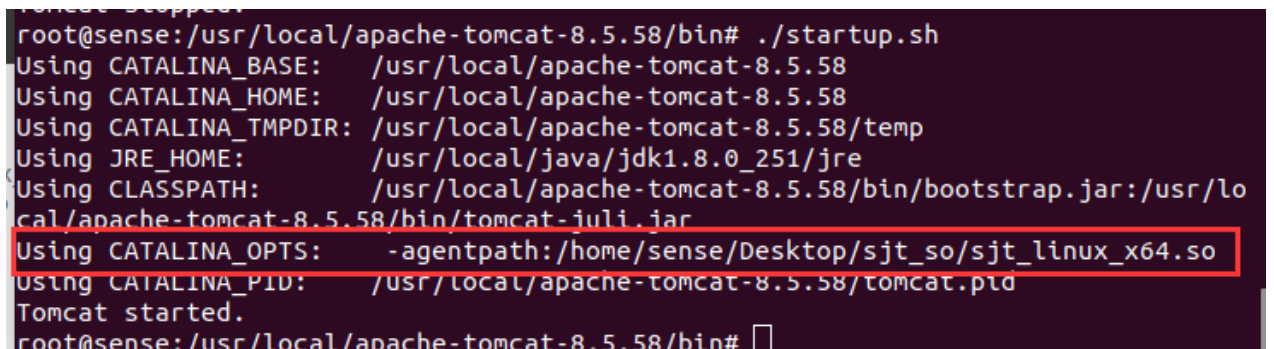
Setenv.sh in the tomcat directory:

- Create a new setenv.sh in the tomcat\bin directory, the full path environment variable can be set as follows: **CATALINA_OPTS=%CATALINA_OPTS% -agentpath:sjt_windows_x64.dll** .



```
root@sense: /usr/local/apache-tomcat-8.5.58/bin
File Edit View Search Terminal Help
CATALINA_PID="$CATALINA_BASE/tomcat.pid"
JAVA_OPTS="-server -XX:MetaspaceSize=256M -XX:MaxMetaspaceSize=1024m -Xms512M -Xmx1024M -XX:MaxNewSize=256m"
export CATALINA_OPTS="$CATALINA_OPTS -agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x64.so"
```

- Start tomcat service, you can set the **CATALINA_OPTS** parameter.



```
root@sense:/usr/local/apache-tomcat-8.5.58/bin# ./startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-8.5.58
Using CATALINA_HOME:   /usr/local/apache-tomcat-8.5.58
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-8.5.58/temp
Using JRE_HOME:        /usr/local/java/jdk1.8.0_251/jre
Using CLASSPATH:       /usr/local/apache-tomcat-8.5.58/bin/bootstrap.jar:/usr/local/apache-tomcat-8.5.58/bin/tomcat-juli.jar
Using CATALINA_OPTS:   -agentpath:/home/sense/Desktop/sjt_so/sjt_linux_x64.so
Using CATALINA_PID:    /usr/local/apache-tomcat-8.5.58/tomcat.pid
Tomcat started.
root@sense:/usr/local/apache-tomcat-8.5.58/bin#
```

- Put the encrypted war package in the directory: **./apache-tomcat/webapps**
If the War package can be parsed correctly, the webpage can run correctly.

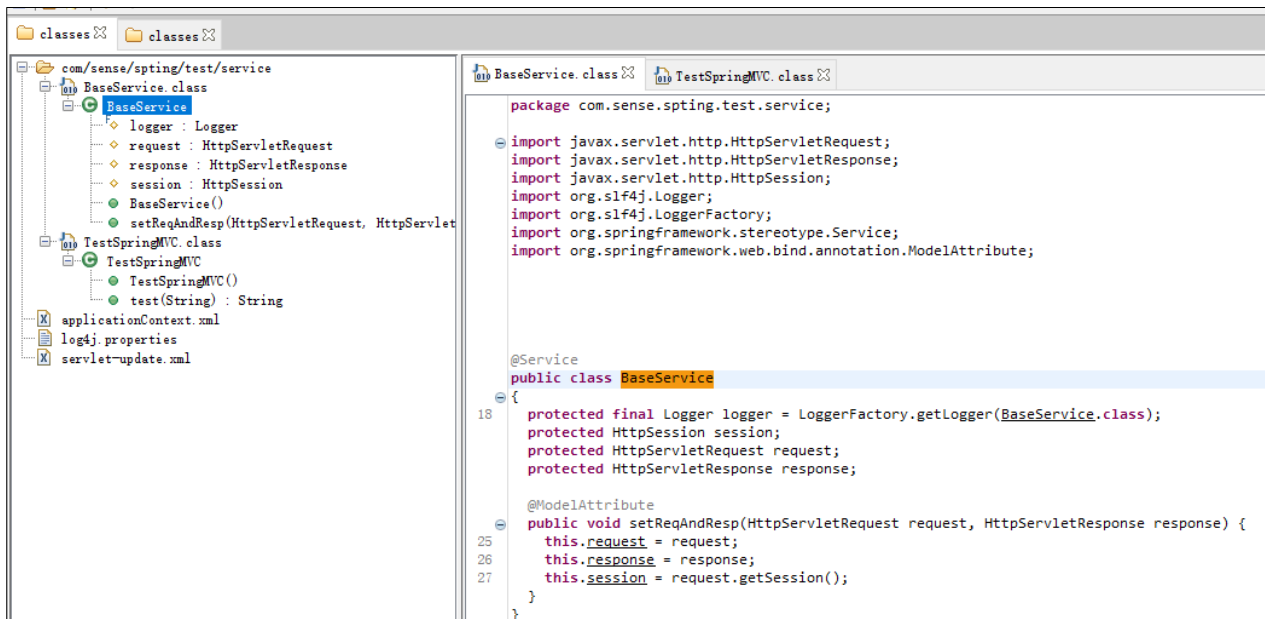
Please noted that: If you have configured the environment variable, the default Java running environment will use the environment variable you have set even you have assigned the **sjt** library location.

Protection performance compare:

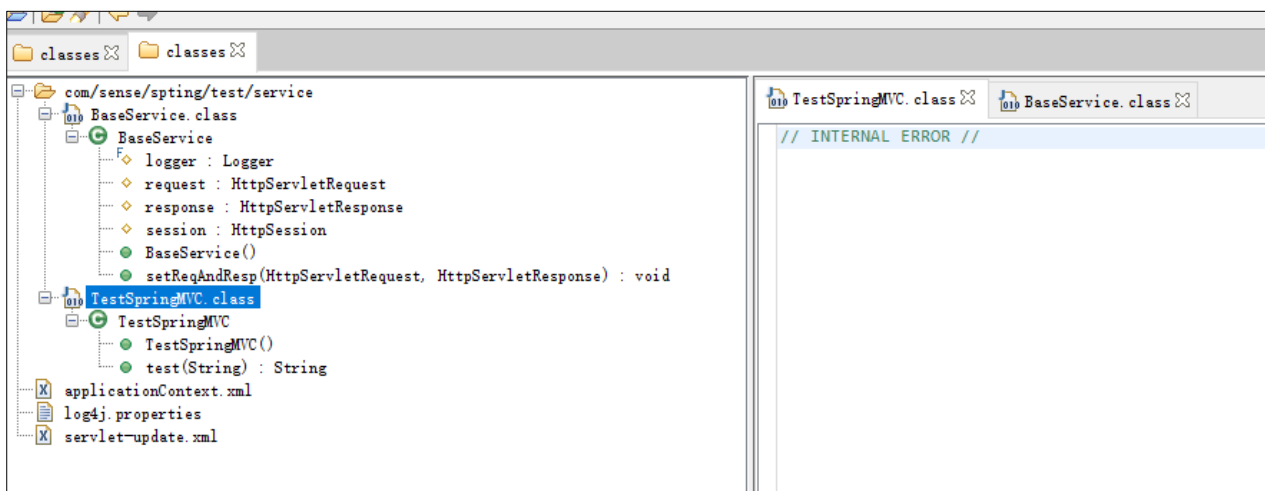
Please noted that, currently only the class file in the war archive would be encrypted.

The following is the encrypt performance comparing by using jd-gui.exe to check the anti-compile protection to the source code:

Before protection:



After protection:



5.4 Unity 3D Program Protection

Why we need to protect the Gaming program based on Unity3D engine?

5.4.1 Introduction

The Unity 3D program mainly uses the C# and open source mono to execute the code logic and algorithm. All of the code is not compiled into the exe file and located at {APP}\build\game_Data\Managed\Assembly-CSharp.dll (note that the program with Unity-2017 is slightly different).

And the mono execution is compatible with the Microsoft .NET Framework but the execution principle is completely different. The traditional protection to .NET Framework will be invalid to protect the mono execution. Since Assembly-CSharp.dll is neither a dynamic library in PE format nor a dynamic library in .NET, it cannot be loaded from the .NET Framework. Instead, the mono.dll read the C# script inside of Assembly-CSharp.dll from mono.dll. Interpreter it and execute the program.

If you protect the Unity3D program with traditional software protection tool, it would not protect the main code source. But Virbox Protector will not only protect the source code, but also protect your resources (.resS). To protect your copyright and IP.

5.4.2 Protection Mechanism

Virbox Protector protects the whole source directory of the **Unity3D** program, for the resource file, Virbox Protector will use “**Resource Encryption**” to protect it. In this way to protect your software main source code from being decompiled. And prevent your resources (**.resS**) from being extracted illegally. To protect the copyright and IP of the software developer.

Protection Mechanism:

1. Parse the Assembly-CSharp.dll script file and convert the function into IL code.
2. Encrypt the IL code, where the key is randomly generated every time and kept in the script file.
3. Link and regenerates the Assembly-CSharp.dll script file. All codes have been encrypted.
4. Process the .NET runtime library mono of Unity3D, locate the function that parses the .NET method and add hook.
5. Insert the hook code to decrypt the Assembly-CSharp.dll method, recompile to generate a new mono and replace the original dynamic Library

The Purpose:

1. Encrypt the Unity3D script C# code, prevent reverse engineering and anti-compile;
2. Add the program set file: the c# program set in the managed directory which is developed by the software developer.

5.4.3 Windows, Linux, macOS platform protection

5.4.3.1 Protect with Virbox Protector GUI

Drag the whole U3D directory to the Virbox Protector

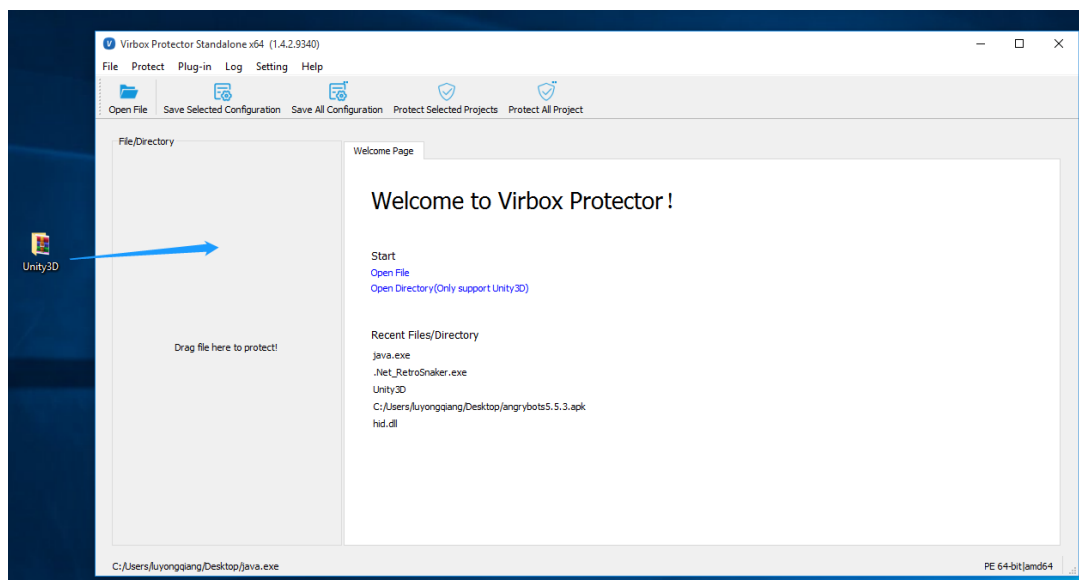


Figure 5-41

The Assembly-CSharp-firstpass.dll and Assembly-CSharp.dll would be loaded in the “Protection Options” tag, you can add the customized C# program set in the /Managed directory.

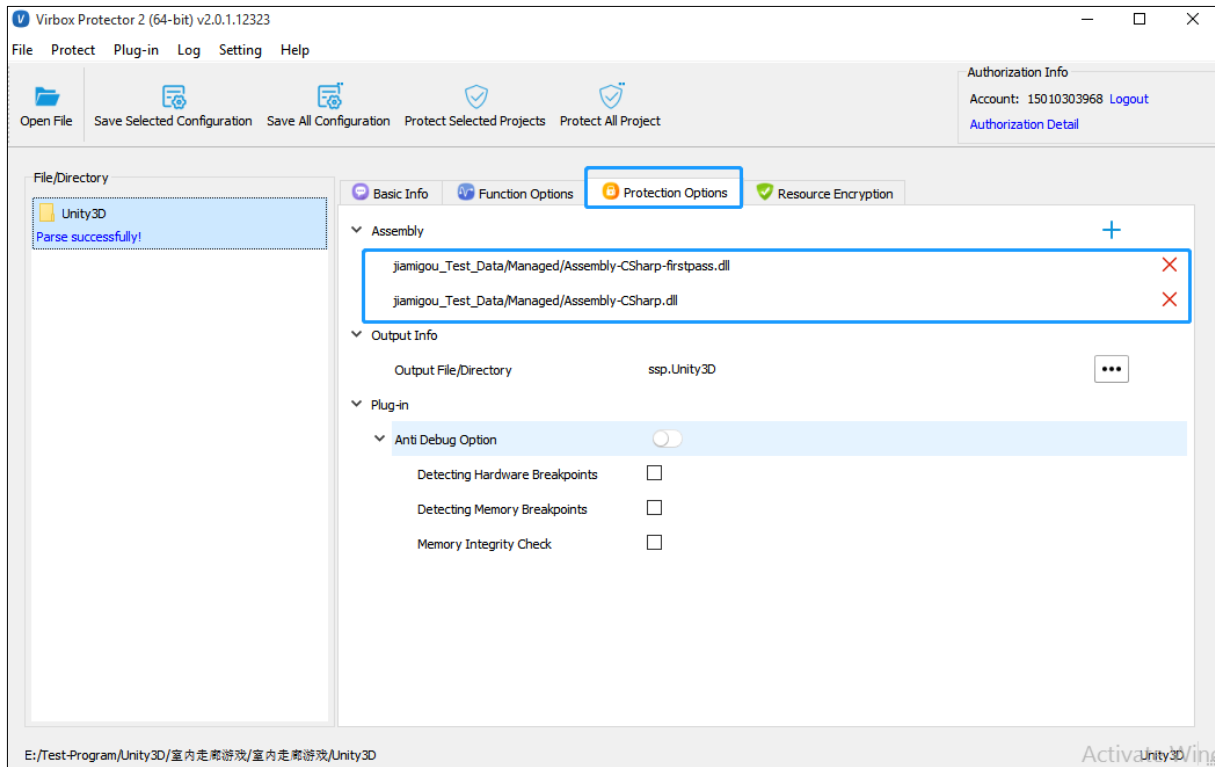


Figure 5-42

You can also enable the “Resource Encryption” option such as the picture such as the following shows to encrypt the resources in your Unity3D program:

Please noted that: It is recommended that only the resources selected by default will be encrypted.

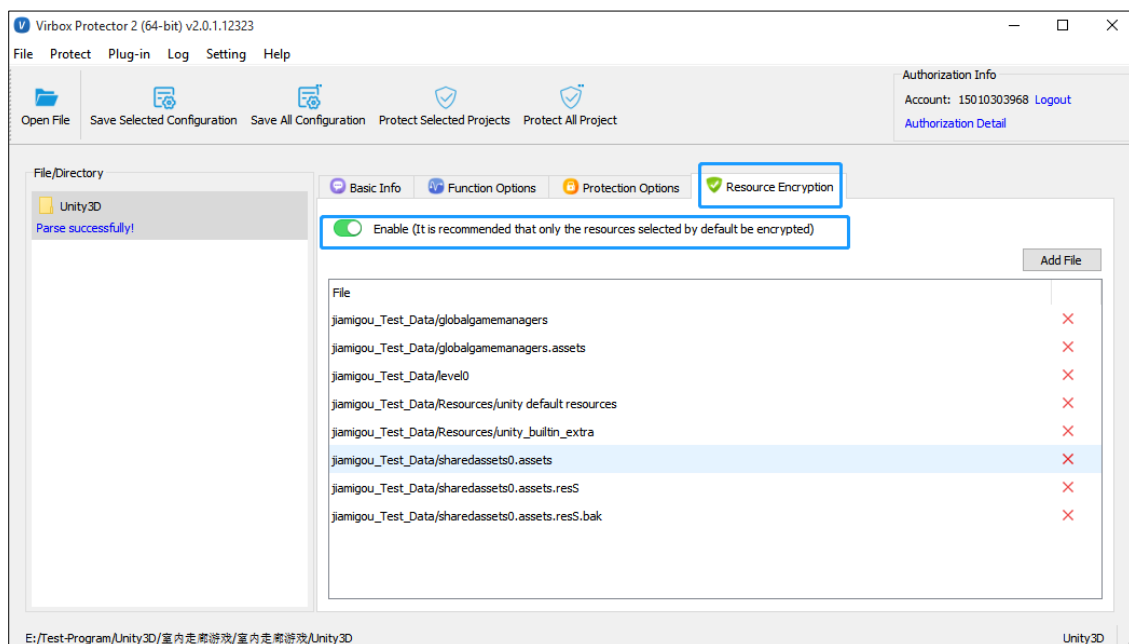


Figure 5-43

For the configuration of “**functions to be protected**”, we do not need to set, Virbox Protector will protect all of the functions in “**Assembly-CSharp.dll**” and “**Assembly-CSharp-firstpass.dll**”.

After protection, Virbox Protector will generate a new folder, same directory with the original folder named ssp.xxx (xxx is the original directory name), the picture shown below.

Two more files would be generated:



Figure 5-44

And it will remind you protection successful.

“**Unity3D-Test.ssp**” is the configuration file you may need to use for resources protection.

“**ssp.Unity3D-Test**” folder is the Unity3D program after protection, you can distribute this file to the software user in future.

5.4.3.2 Using Command Line to protect the Unity3D program

Unity3D, as a special file type, the protection methods is different from the normal program. For the Unity3D program for Windows, Linux and macOS platforms, the entire directory of Unity3D needs to be protected; Here we take a Linux Unity3D as an example:

- Use the Virbox Protector GUI tool to generate configuration files (optional)
- Open a terminal window, enter the path where “**virboxprotector_con**” is located, and enter “virboxprotector_con” to run Virbox Protector. Help information can be viewed.
- For the programs in different platforms, the Virbox Protector need to verify the license in different platform. You need to contact Virbox team to obtain the corresponding license.

Command: *Path of VirboxProtector_con path of the program to be protected -u3d -o Path of output file*

If no license has been verified, when you run Virbox Protector, it will prompt “Can not find the license”, as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particles2018.1.9f1'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d Particles2018.1.9f1 ...
Error (13000020): Can not find the license.
```

Figure 5-45

After the license is verified, the program can be successfully protected by Virbox Protector, as shown in the figure:

```
sense@sense:~/Desktop/virboxprotector_1.5.0.10808/bin$ ./virboxprotector_con '/home/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particles2018.1.9f1'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d Particles2018.1.9f1 ...
Succeed.
```

Figure 5-46

5.4.4 Unity3D android application

5.4.4.1 Protect android application with GUI

5.4.4.1.1 Mono format

For unity3D android application different protection process need to be used for different compile option, mono and IL2CPP.

For Mono:

1. Unzip the Android Unity3D application(apk), the library directory will show the so library



Figure 5-47

- If the libmono.so library is contained in the /lib directory, means the compile option used is mono, now you need to protect *all of the apk directory* by use of Virbox Protector:

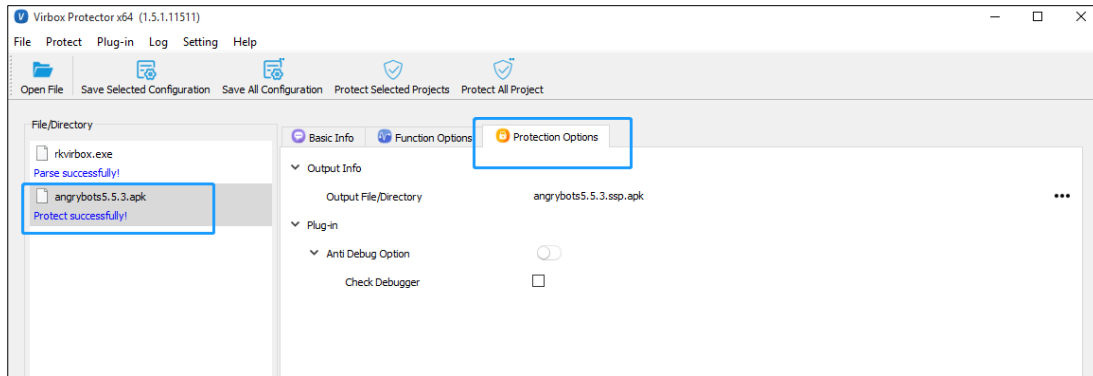
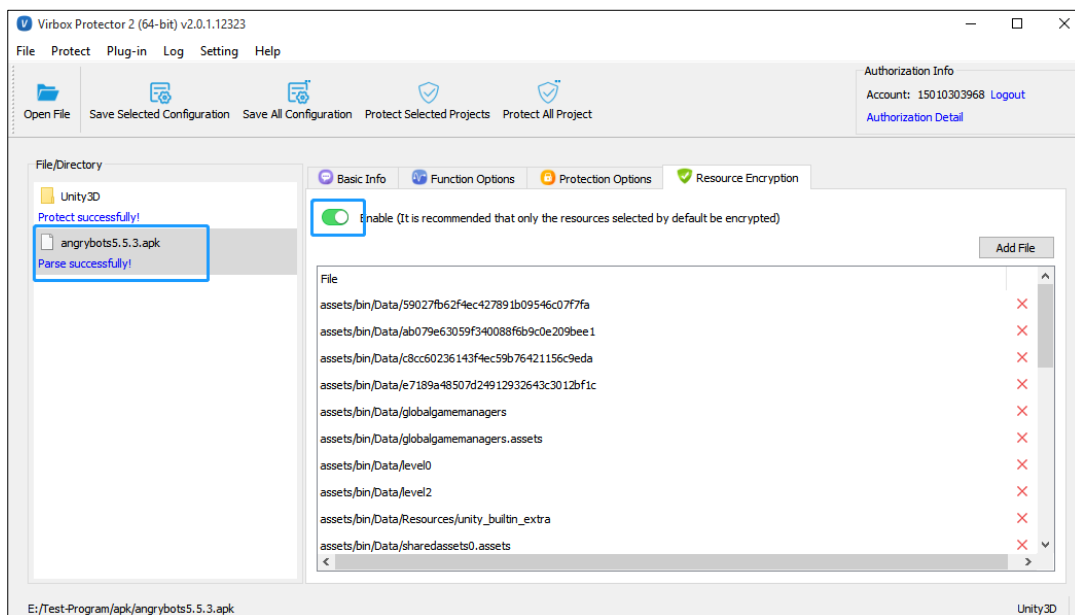


Figure 5-48

- Open the “Enable” button in the “Resource Encryption” tag, then you can encrypt the resource file of the Unity3D program.



- After protection, a new ssp.apk would be generated, you can re-sign this ssp.apk and pack, to let it can be installed.

Drag the apk of Android Unity3D in to the Virbox Protector, after protection a new file named xxx.ssp.apk will be created, Virbox Protector will protect the “libmono.so” and “Assembly-CSharp.dll”.

Before protection:

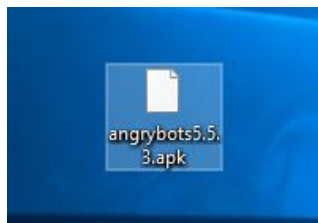


Figure 5-49

The un-protected file is named “angrybots5.5.3.apk”

After protection:

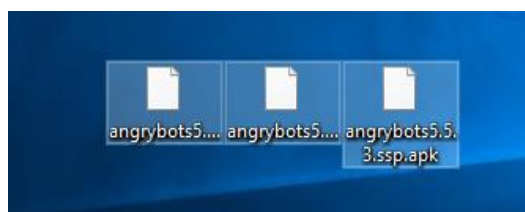


Figure 5-50

After protection,

“angrybots5.5.3.ssp.apk” is the file after protection. This file can be released to software user in future.

“angrybots5.5.3.apk.ssp” is the configuration file. If you want to protect the data resources you need to use this file or this file can be deleted.

“angrybots5.5.3.apk” is the unprotected file. You **can’t** release this file.

After the application is successfully protected, you need to sign the protected application (.ssp.apk file) and release it.

5.4.4.1.2 IL2CPP

1. Unzip the Android Unity3D apk and check the library directory:

桌面 > apktool > android_il2cpp > lib > armeabi-v7a				
搜索"armeabi-v7a"				
名称	修改日期	类型	大小	
libdun.so	2020/4/14 15:06	SO 文件	1,734 KB	
libil2cpp.so	2020/4/14 15:06	SO 文件	14,219 KB	
libmain.so	2020/4/14 15:06	SO 文件	351 KB	
libunity.so	2020/4/14 15:06	SO 文件	9,172 KB	
libxlua.so	2020/4/14 15:06	SO 文件	1,387 KB	

Figure 5-51

2. If the so library libil2cpp.so is contained in the /lib directory, means the application used IL2CPP compile option. You need to protect the *so library* in the lib directory,
3. After protection, re-pack the application and sign the signature to the application then release it.

5.4.4.2 Use command line to protect Unity3D apk.

Virbox Protector use a different way from normal apk program to protect the Unity3D program as the particularity of Unity3D program of Android platform.

1. Use the Virbox Protector GUI generate “.ssp” configuration file.
2. Open the Virbox Command line window, get into the “Virboxprotector_con.exe” directory, and input “virboxprotector_com.exe”, you can see the help info;

Command: path of “VirboxProtector_con.exe”+ “the path of the program to be protected” + -u3d -o “the path of the output path of the program”

If the license to unity3D program can be found:

```
C:\Users\test\Desktop\virboxprotector_standalone_1.4.2.10236_windows_x64\bin>virboxprotector_con.exe C:\Users\test\Desktop\sample\angrybots5.5.3.apk -u3d -o C:\Users\test\Desktop\sample\ssp.angrybots5.5.3.apk
SenseShield Virbox [version: 1.4.2.10236]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

protect unity3d angrybots5.5.3.apk ...
Error (13000020): Can not find the license.
```

If the program can be protected:

```
C:\Users\test\Desktop\virboxprotector_standalone_1.4.2.10236_windows_x64\bin>virboxprotector_con.exe C:\Users\test\Desktop\sample\angrybots5.5.3.apk -u3d -o C:\Users\test\Desktop\sample\ssp.angrybots5.5.3.apk
SenseShield Virbox [version: 1.4.2.10236]
Copyright(c) SenseShield Technology Co., Ltd. All rights reserved.

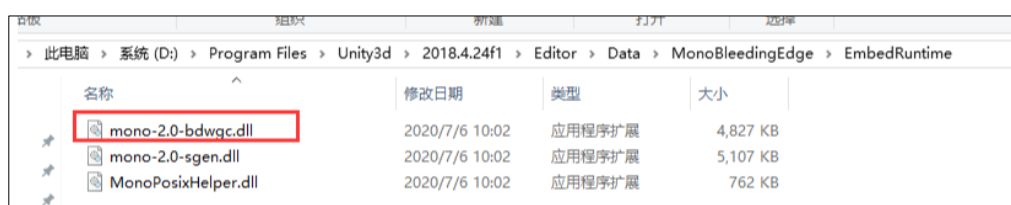
protect unity3d angrybots5.5.3.apk ...
Succeed.
```

5.4.5 Unity3D program call Net dll plugin

1. Firstly find the location of **the mono.dll** or **mono-2.0-bdwgc.dll** of the Unity3D compiler:
 - The **mono.dll** is always in the location: **.\Editor\Data\Mono\EmbedRuntime** directory

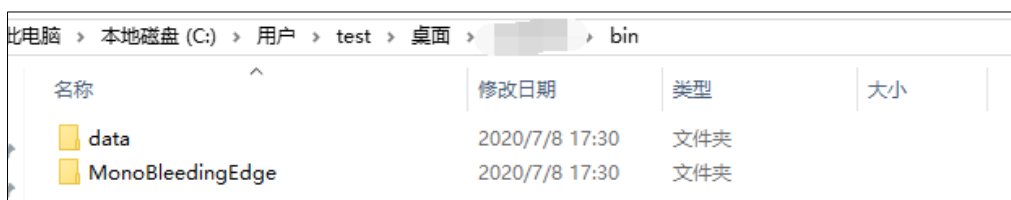


- **mono-2.0-bdwgc.dll** is always in the location:
.\Editor\Data\MonoBleedingEdge\EmbedRuntime

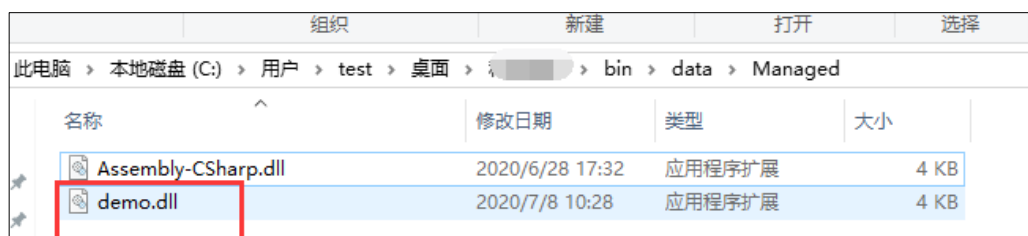


- You can create a fake Unity3D directory, the directory which need to be same with the Unity3D program directory. This is to help Virbox Protector can recognize the program successfully. Here we take the file mono-2.0-bdwgc as an example:

- Put **mono-2.0-bdwgc.dll** in the directory **bin\MonoBleedingEdge\EmbedRuntime**

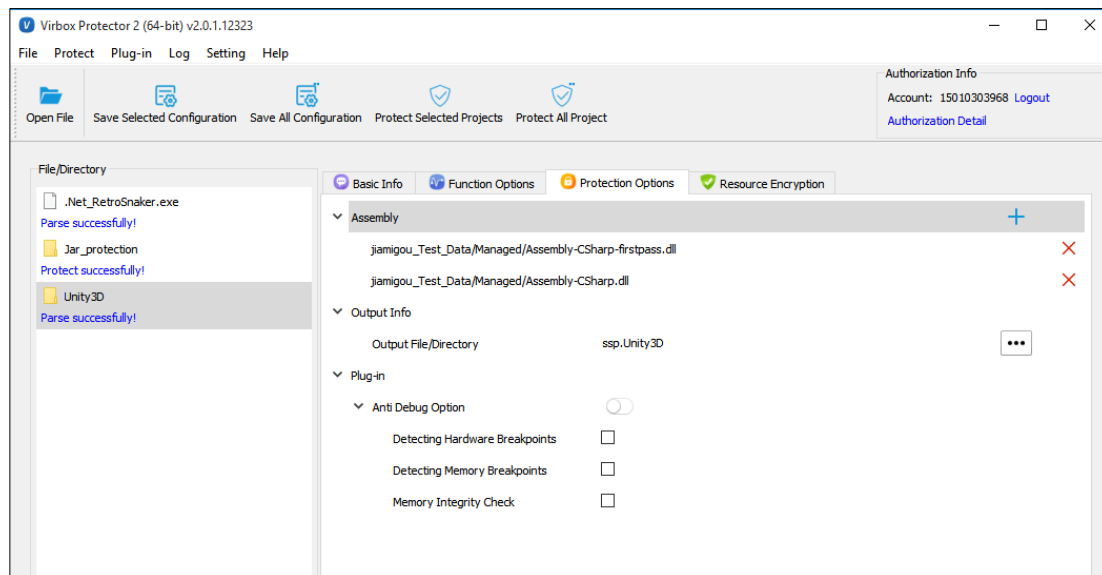


- Put the dynamic link library to the location: **bin\data\Managed such as(demo.dll)**

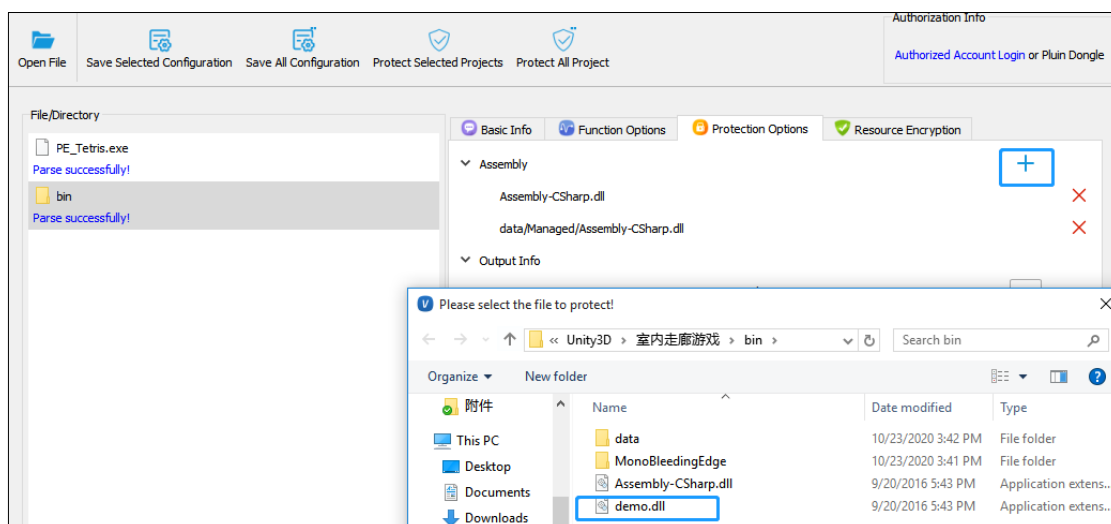


- Drag the bin directory to the Virbox Protector:

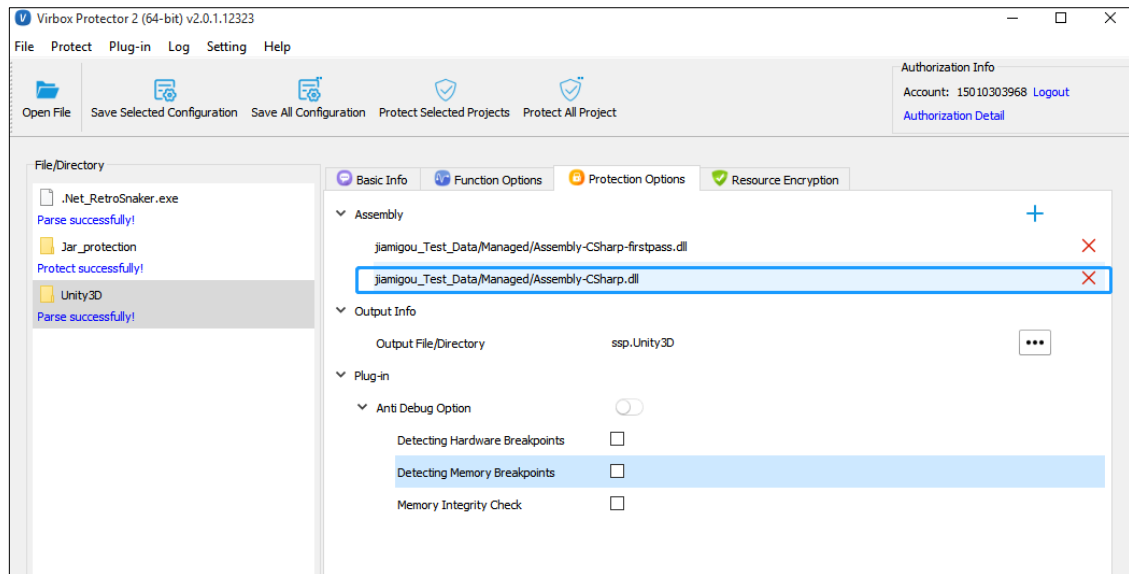
- Set License:



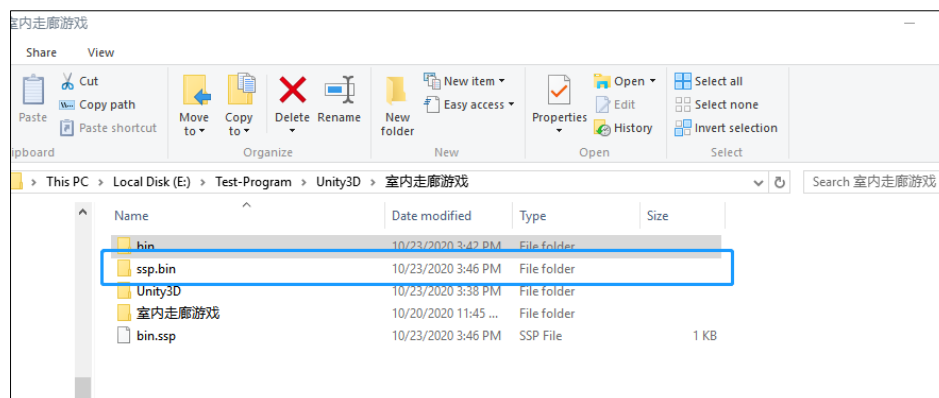
- Add the program set in the “protection option”:



- After the program set is added successfully, delete the non-necessary program and reserve your dll.



- Click ***“Protect the selected protect”*** to protect the program, the ssp.bin would be generated.
This is the program after protection.



- Get into the ssp.bin directory, copy the

“mono-2.0-bdwgc.dll”

in the ***ssp.bin\MonoBleedingEdge\EmbedRuntime***

directly to ***“Editor\Data\MonoBleedingEdge\EmbedRuntime”*** then put the ***“demo.dll”*** in the
directly ***“ssp.bin\data\Managed”*** into project.

In this way to protect the program you want to protect.

5.4.6 Protection Comparison

Assembly-CSharp*.dll script file has been protected by using Virbox Protector.

Before protection/encryption:

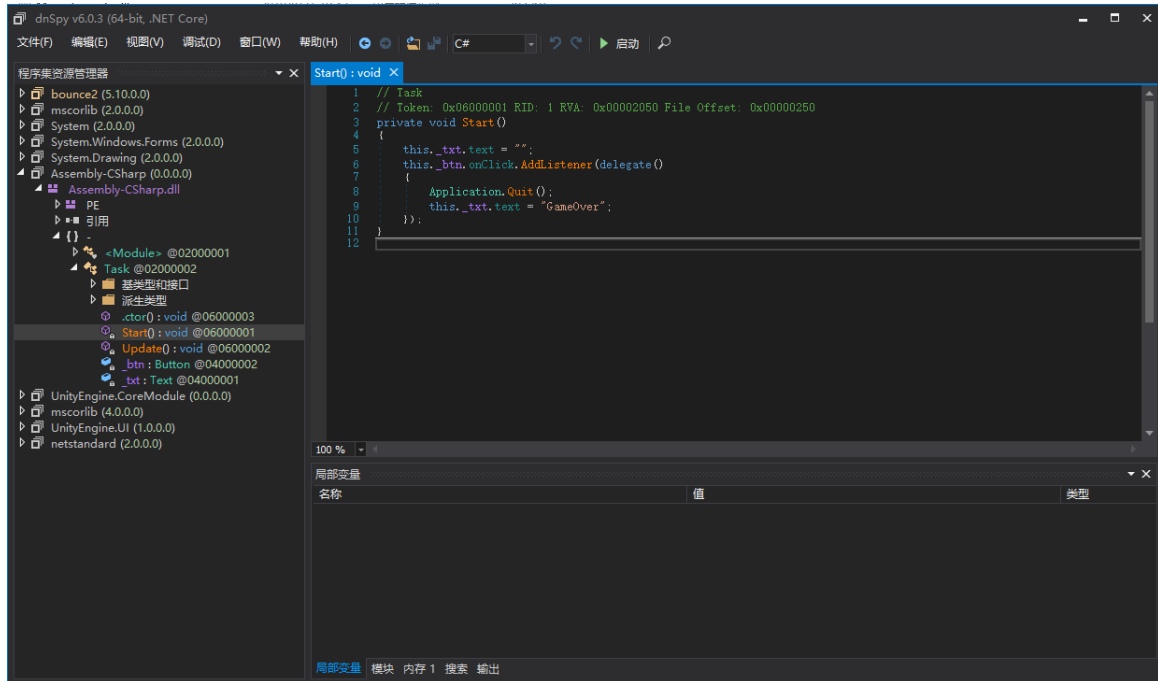


Figure 5-52

After protection/encryption:

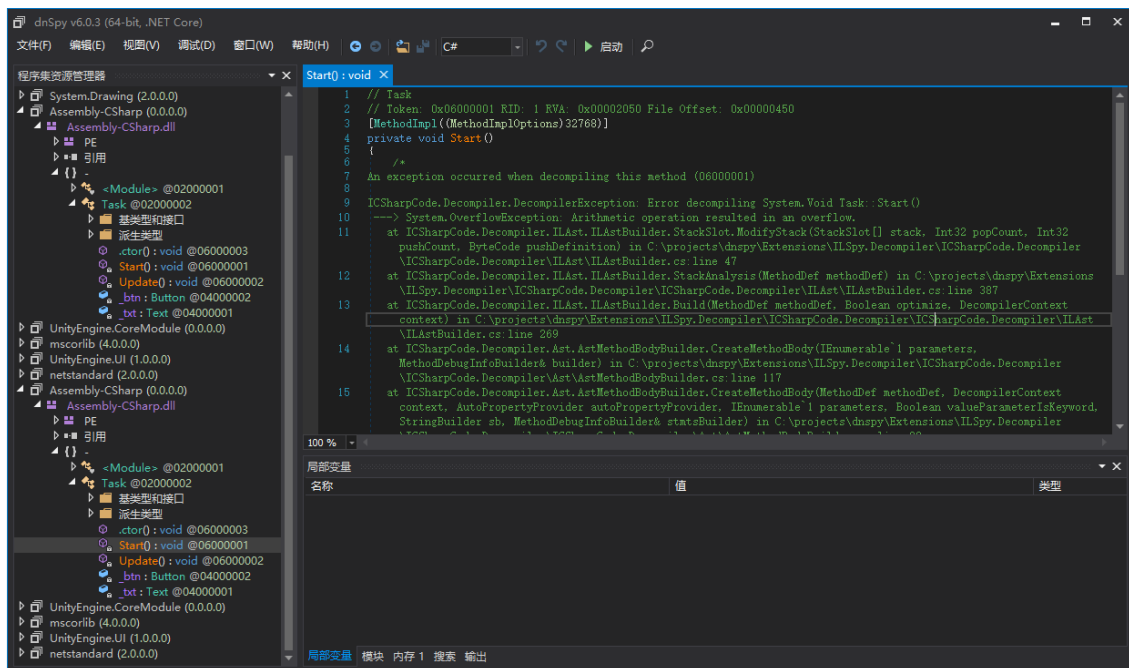


Figure 5-53

Protection comparison (Using the de-compiler tool: AssetStudio to decompile the resource file with and w/o protected by DS Protector.)

Before protection:

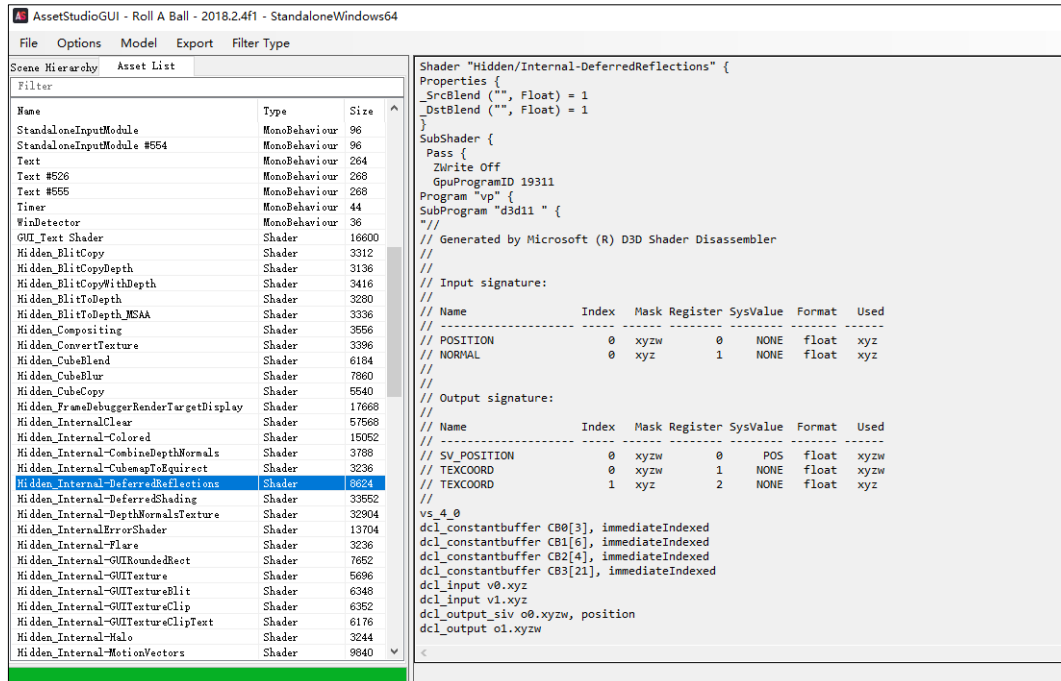


Figure 5-54

After Protection:

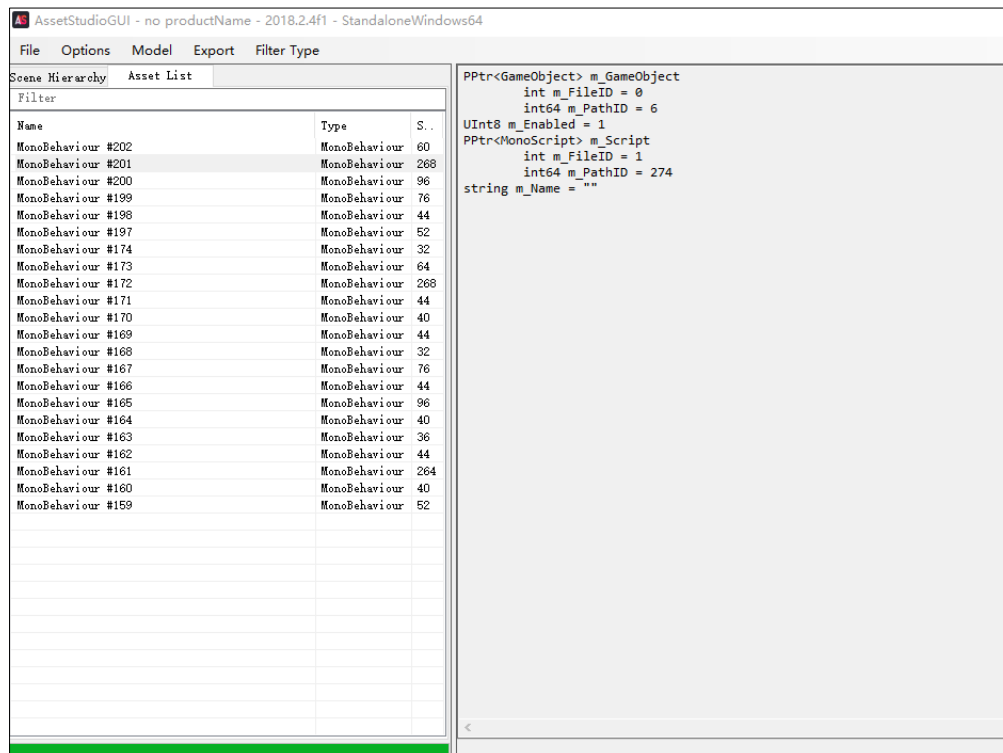


Figure 5-55

5.5 Protect Android application

5.5.1 Normal apk application

For normal apk application, the application need to be unzipped and then to protect the so library in lib directory by use of Virbox Protector, such as:



名称	修改日期	类型	大小
libad.so	2020/5/13 15:10	SO 文件	668 KB
libBugly.so	2020/5/13 15:10	SO 文件	437 KB
libgetuiext3.so	2020/5/13 15:10	SO 文件	1,060 KB
libInnoSecure.so	2020/5/13 15:10	SO 文件	442 KB
libInnoSo.so	2020/5/13 15:10	SO 文件	1,849 KB
libNativeExample.so	2020/5/13 15:10	SO 文件	528 KB
libpl_droidsonroids_gif.so	2020/5/13 15:10	SO 文件	322 KB
libsgmain.so	2020/5/8 16:04	SO 文件	382 KB

Figure 5-56

After protection, you need to pack all of files with signature to the apk application, then release the application.

5.6 Protect the python based application

Parser protection

- Python.exe (interpreter) file based on python protection, the detail steps are same with Windows Application, please refer the steps above. Chapter 5.1, use the default setting to encrypt the **Python.exe**

Function Options Setting:

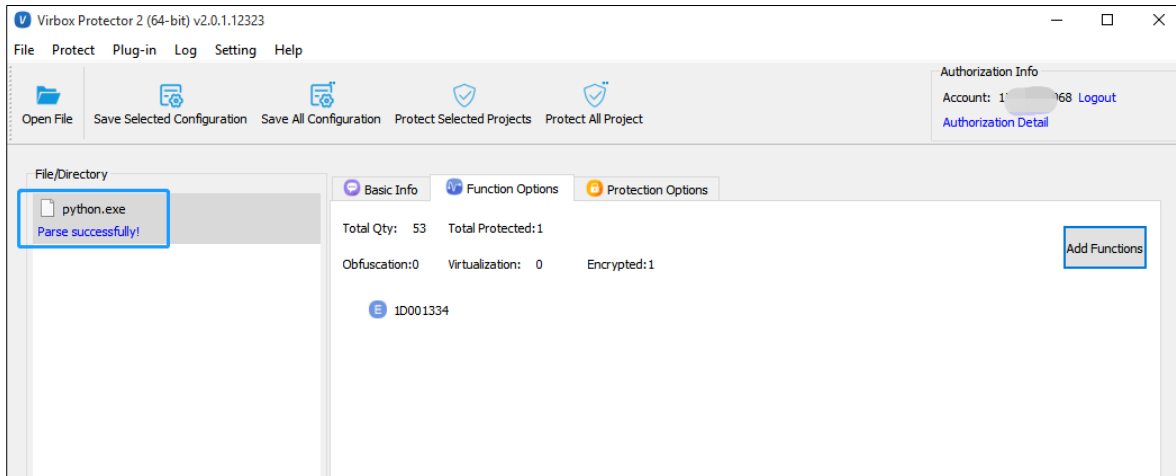


Figure 5-57

Protection Options Setting:

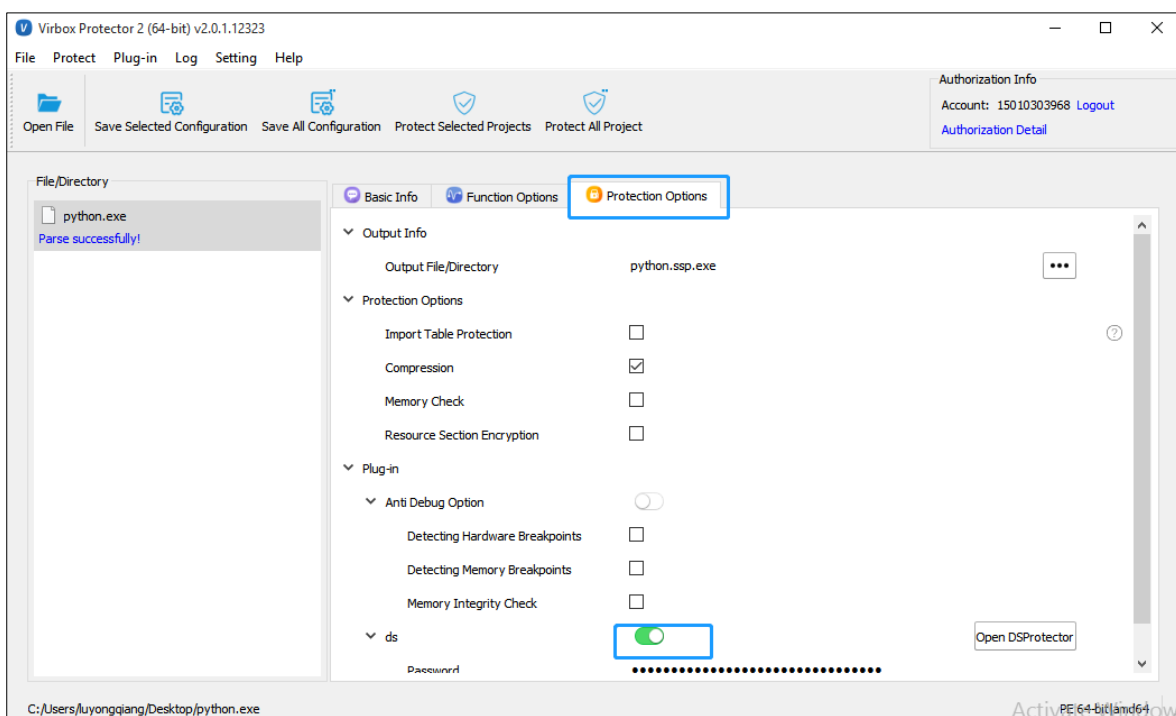


Figure 5-58

Resource Protection

- For resources protection, protect the python.exe with Virbox Protector first, and then protect the **.pyc**, **.pyd** and **.py** file with DSProtector.

In this example, after protect the relevant file with above process, you will get 3 file

“python.exe.ssp” is the configuration file, and when you are protecting the **.py** and **.pyc** file, you will need this

file.

“**ssp.python.exe**” is the **python.exe** file after protection, you need to use this file to parse the protected **.py** and **.pyc** file. (The **.py** and **.pyc** file need to run with the **ssp.python.exe** file). When you run the protected **.py** and **.pyc** file.

Please modify the **python.ssp.exe** to be **python.exe**, in order not to influence the existed python environment.

“**python.exe**” is the file before protection.

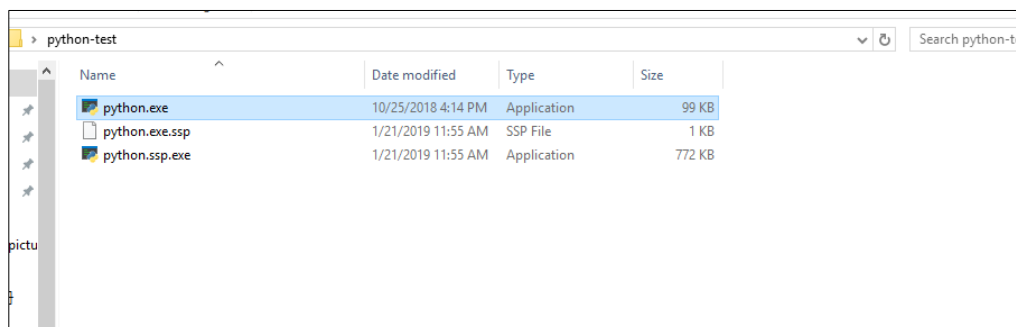


Figure 5-59

Open the DSProtector for **.pyc** and **.py** file protection,

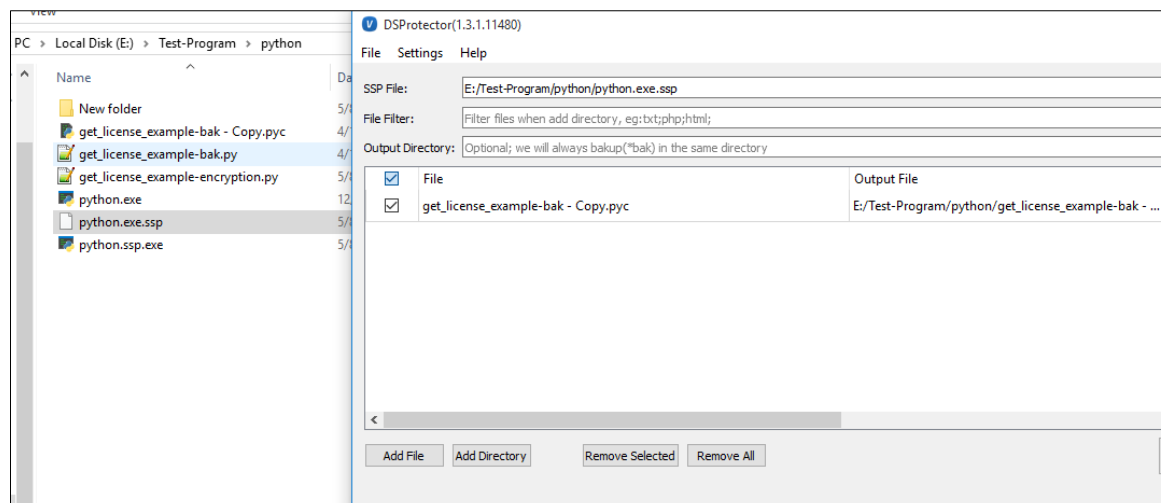


Figure 5-60

Choose the **ssp configuration file** created last step when you are protecting “**python.exe**”

Drag in the **.py** and **.pyc** file,

Click “**Protect**”, it is showing “**protect success**”

The protected file name will use the original name, and the unprotected file will be with bak file name extension.

Till now the protection of **python resources** file has been completed, and you can release the protected file to

the customer.

6 Note

Type		Unsupported scenario
Other		You can't use Virbox Protector to protect the software which protected by third party protector or wrap tools, any protected software can't be protected by Virbox Protector or third party Wrap tools to protect again.
		The Jar archive is not supported to be protected by Virbox Protector, The Jar archive can be protected by DS Protector
		The program with Self verify function can't be protected by Virbox Protector
File Type	.NET	The program with StrongName signed can't be protected by Virbox Protector
		Third party runtime library is not supported to be protected by Virbox Protector, only Micro soft standard runtime library can be protected
		SDK label doesn't support for .Net program
		If .Net program developed by C# language contains the method that called by external program or contain public method, name obfuscation can't be used to encrypt these kind of method. Because the name obfuscation may change the method name and cause some of functions may not be used.
		Compression will modify the program type from AnyCPU to be PE32. So if you compress .Net AnyCPU program. It would be no longer to be called by other program.
		Compression function do not support .NET dynamic link library
	PE	Resources protection do not support the executable converted from Powerpoint
		Resources protection do not supported for the program of VB6.0 language
		Import Table: The symbol imported must be function and can't be import variable, or the program would crash when started.
		If the protected program run with load memory type, it can't be started if compression is used.
	ELF	Additional data is not supported for Linux program currently.
		ELF program which is compiled by -static compile option is not supported to be protected by Virbox Protector.
		Map file analysis is not support ELF format file.
		If all of the symbol in the program is exported, crash may happen when you run the program. Only required function suggested to be exported.
Protection	Code	The function parsed by Virbox protector without function name may not

Option	Encryption	supported by Virbox Protector, because of external function entry may exist.
		For the function instruction which too less may not be protected by Virbox Protector
	Obfuscation/ Virtualization	For ELF and Mach-O program which compiled with C or C++, the program may not be use obfuscation and Virtualization to protect. Because of stack frame issue. For such case, you may use the " code encryption " option to protect the application.
		For the function's instruction is too less, it may not be protected.
		Virtualization and code snippet is not supported for ARM architecture program.

6.1 Known Issues

- The protected .Net program only support Microsoft standard running lib, do not support third party running lib
- When using the command line to protect your software, the configuration file of the objective file must be exist.
- When `GetField("name", bindingAttr)` used in the .Net program to be protected, and if you select the "Name obfuscation" in "Function Option", the software may fail in execution, and you need to remove the obfuscation from the "Function option".
- You may fail to protect the software with code snippet, because of too less of instruction of the snippet code, maybe jump, and it can't be code ported.
- The name of the software after protection will be changed, please modify it to be the original name. The ARX plugin of AutoCAD can only select "remote desk service dialog message box", and now only support win7 and server2008 or above version.
- Anti-virus AVAST may cause the start failed of the protected program, it will kill the thread of the protected software when it executed.
- Program with strong signature is not supported

7 FAQ

7.1 What is the difference between the soft license edition and dongle edition?

For Virbox Protector with soft license: the license allowed to be bind with one computer only and you may change the device up to 2 times.

For Virbox Protector Dongle based license: In addition to the software, you will also get a dongle that stored license of Virbox Protector. Any computer plugged dongle can use Virbox Protector.

7.2 What is the difference between the trial edition and standard edition?

For a Trial Edition,

- The protected software program will valid within 7 days for your internal testing and evaluation. after 7 days, when you run the protected software it will have the message pop up:” **This application is protected with trial version of Virbox Protector**”. The license of trial edition Virbox Protector will be expired after 30 days or 100 times usage. For standard edition, you can protect your functions without above limitation.
- Trial edition Virbox Protector can be used to test and protect the program for: Windows, Linux, Mac, ARM Linux, Android. For standard edition, you need to purchase the corresponding license with your program you are going to protect.

No matter which software area you come from, we have experts who understand the special challenges you are facing in your industry. We will help you solve those problem with what we have. And we have helped thousands of software enterprises from different industry to Protected the software and helped them realized software monetization. And we have established special Internet sales model and deeper customer relationships with our customer. We can also do this for you.