User Manual

Virbox Protector (Standalone)

Version 2.4

Copyright & Trademarks

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 want to 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 Family members

Product	Description	Charge mode	Operation System & Platform support
Virbox Protector (Standalone)	Independent Protection tools, support to integrate work with the third party License Solution, or be used for software application protection, shielding and hardening;	Purchase according to program language and operation system	Following release version available: Windows, Linux, macOS, ARM Linux and Android system
Virbox Protector (Trial Edition)	For developer to trial and Evaluation only, the protected application by Virbox Protector (Trial Edition) will be valid & used within 7 days;	Free	Following release version available: Windows, Linux, macOS, ARM Linux and Android system
Virbox Protector (LM)	This version contain and provided by Virbox LM SDK, support macOS, Linux and Windows System only (it doesn't support ARM Linux, Android and other IoT system;	Free	Windows, Linux and macOS
Virbox Protector (LM) Pro	For Developer who want to protect the application in ARM linux, Android SO libs, ARM Unity3D, macOS and issue the license by use of Virbox LM SDK, pls use this version,	Purchase according to your System Environment	
Virbox Protector (Moway)	This is the Protector provided to the developer freely who select the Moway Dongle to be the	Free	Windows, Linux

	license container for their applications, it contained in the Moway SDK		
Virbox Protector (Moway) Pro	Enhance the encryption/protection feature to basic functions: support Obfuscation, Code Encryption, Virtualization	Purchase	Windows, Linux

Operation System and Language Supported

File type	Operation System	Architecture	Language
.NET	Windows	X86, X64	VB, C# etc.
.NET Core 3	Windows, Linux, macOS	X86, X64	C#, VB.NET
PE	Windows	X86, X64	C/C++, Delphi, PB, BCB
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, etc.
Java	Windows	X86, X64	Java

Virbox Protector Update History:

Date	Version Number	Important Updates
2020.05	V 1.5	▪ 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	▪ 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
2021.10	V2.4	▪ Add the function to encrypt AAB project's resource and assets; ▪ Add the hot update to the Android APK resource encryption to save time to protect for each version released ; ▪ Add the label to mark and protect critical functions or string, for .NET & Java Applications ▪ Add the function to File verification to APK files ▪ Add and extend to protect the JAR file with shell script in Java VME Protection mode ▪ Support to generate the shell method name in random; Optimization for existed feature and functions • iOS applications Protection doesn't depends on dSYM; • User experience improvement, for the functions which not support to be protected in the Native and .NET applications, Virbox Protector will prompt a alert message only and no error

		message output; • Optimize and accelerate protection time for Native and .NET applications; • Optimize and accelerate the parsing time for Native ARM applications;

Table of Contents

1 Overview

1.1 Virbox Protector Introduction

Virbox Protector, is the latest Protector and Encryption tool for 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, Code and Memory Integrity Check, etc. It is the powerful protection tools 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 Standalone?

   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 scenario of your software applied.
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 PE Edition, Virbox Protector .Net Edition, Virbox Protector Unity3D Edition, Virbox Protector Java Edition, Virbox Protector Android Edition, Virbox Protector ARM-Linux Edition. etc.

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://appsecurity.virbox.com/

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 With 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:

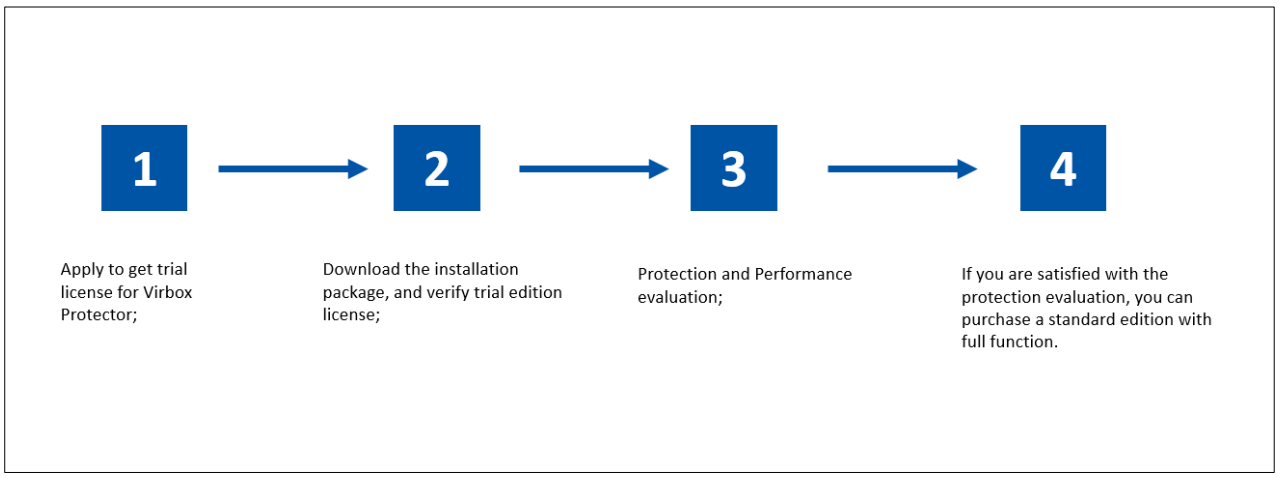


Figure 1-1

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
| └─ds
└─sdk
```

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 for PE application:
https://lm-global.virbox.com/detail/virboxProtector.html

Free trial License apply for Mobile application:
https://appsecurity.virbox.com/

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.

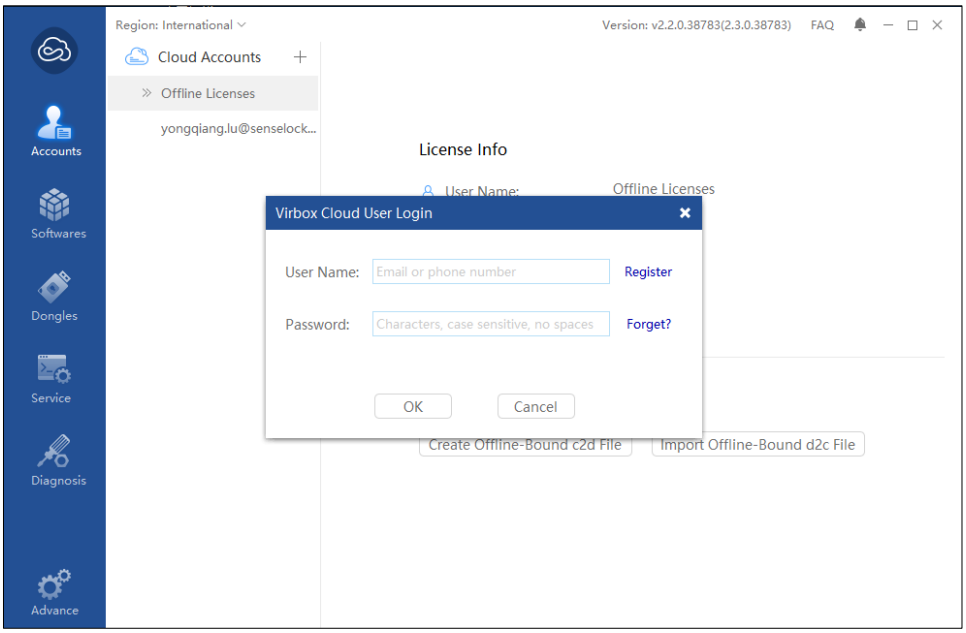


Figure 2-2

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

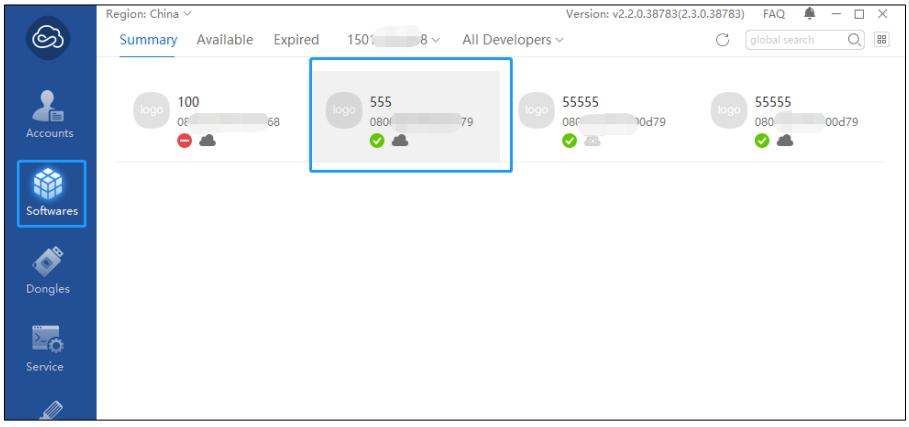


Figure 2-3

2.2.1.2 Sign in by the Virbox Protector GUI Interface

You can also sign in from the Virbox Protector GUI interface:

Sign in the authorized account:

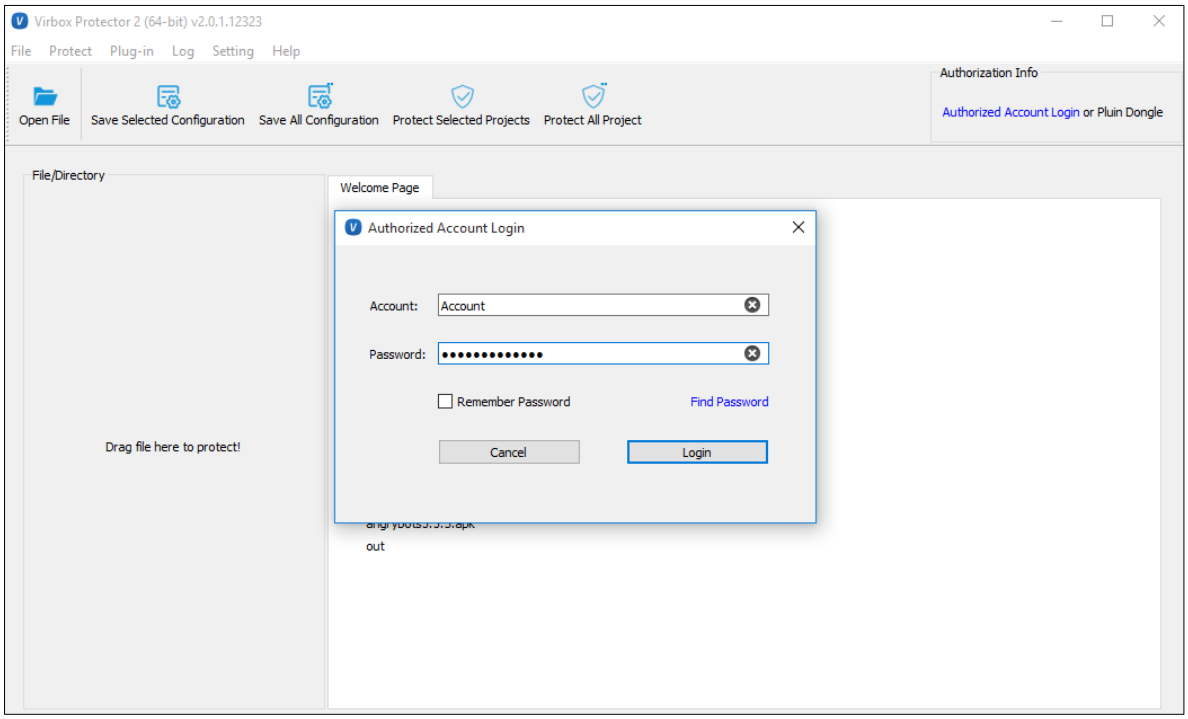


Figure 2-4

You can check the license detailed information by clicking here:

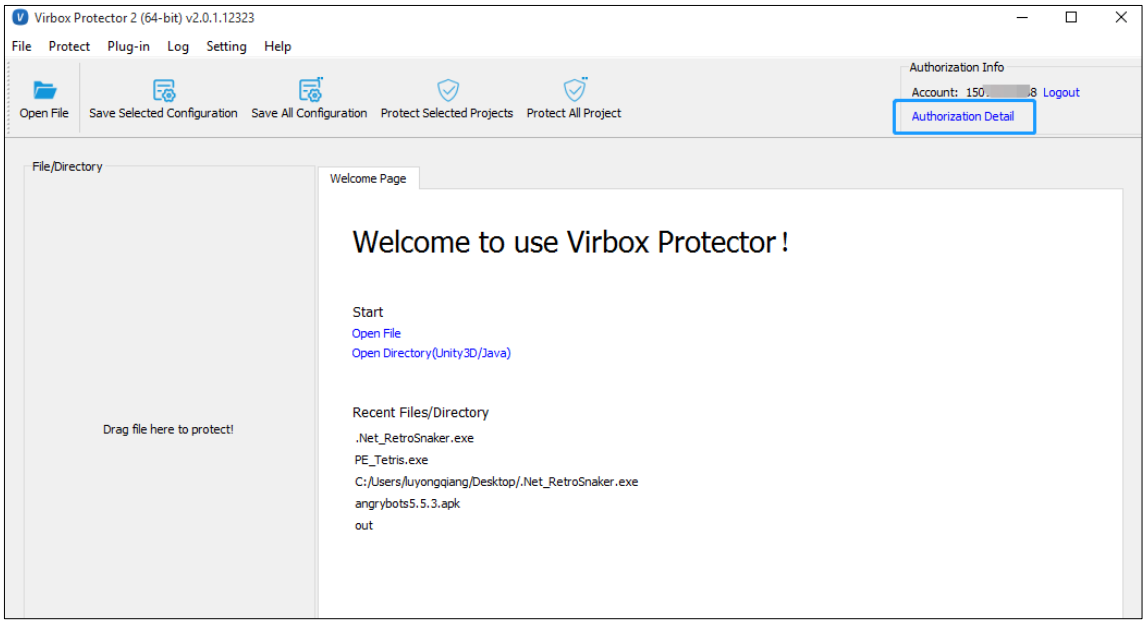


Figure 2-5

You can logout by clicking the “logout” button.

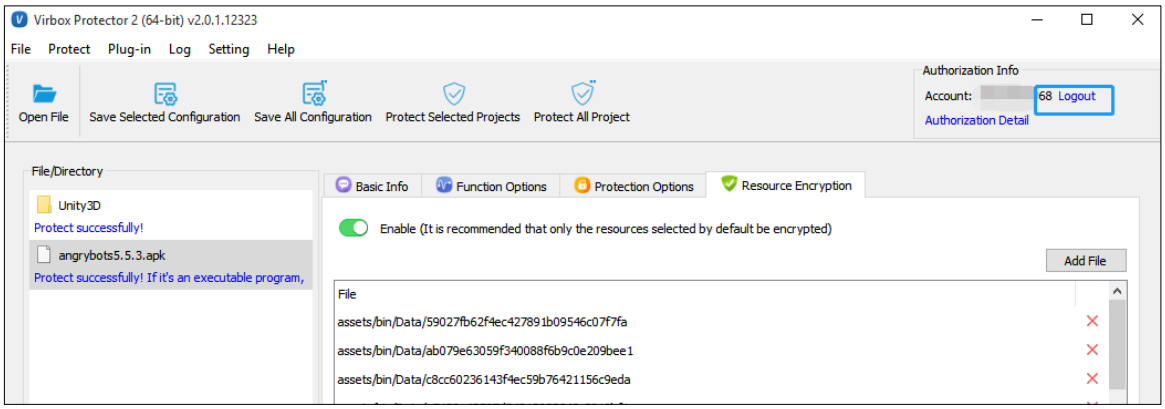


Figure 2-6

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.

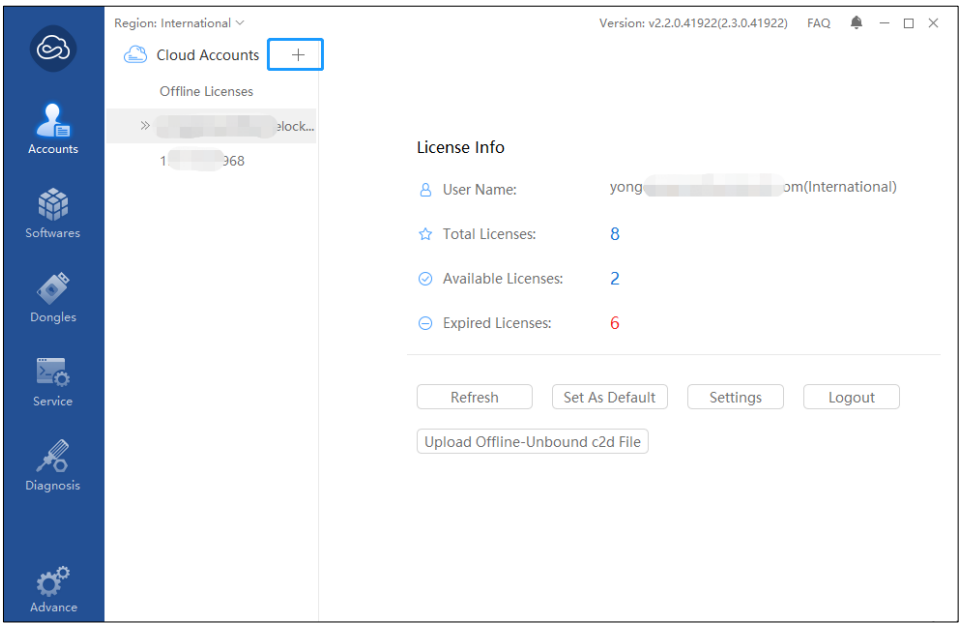


Figure 2-7

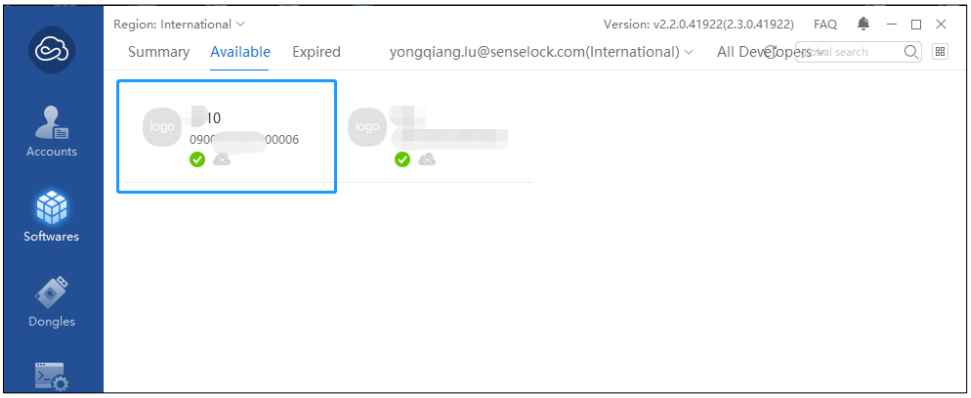


Figure 2-8

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",

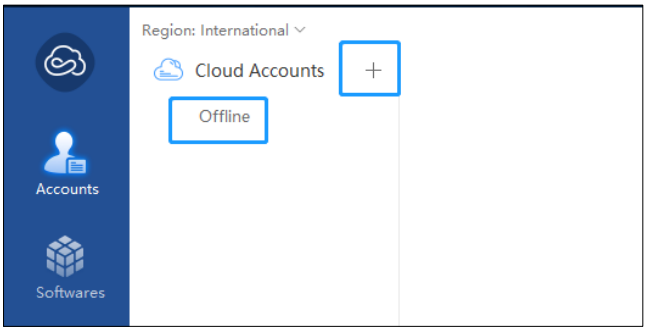


Figure 2-9

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

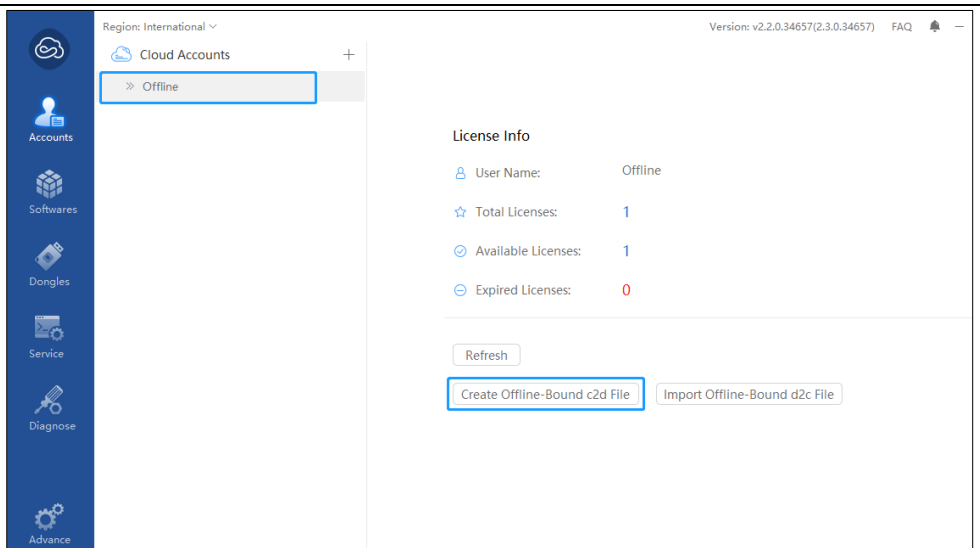


Figure 2-10

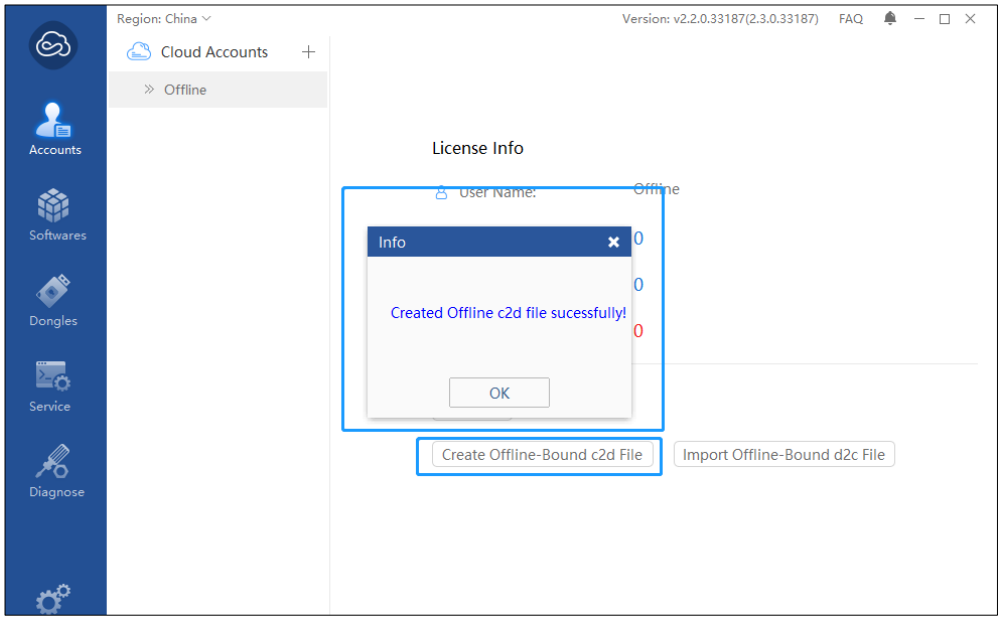


Figure 2-11

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.

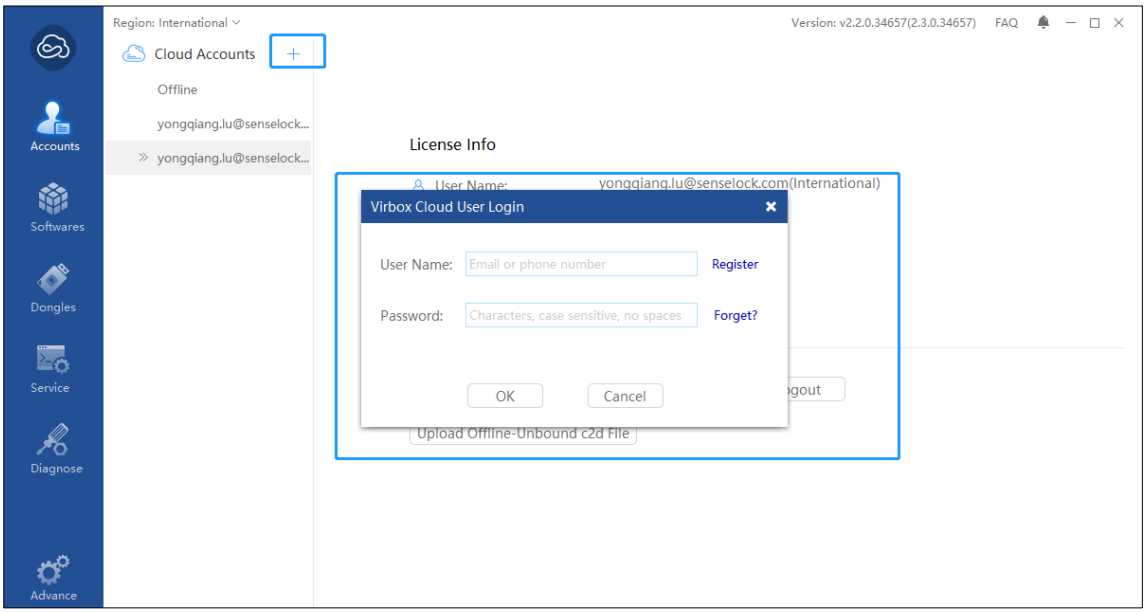


Figure 2-12

Click "**Software**",

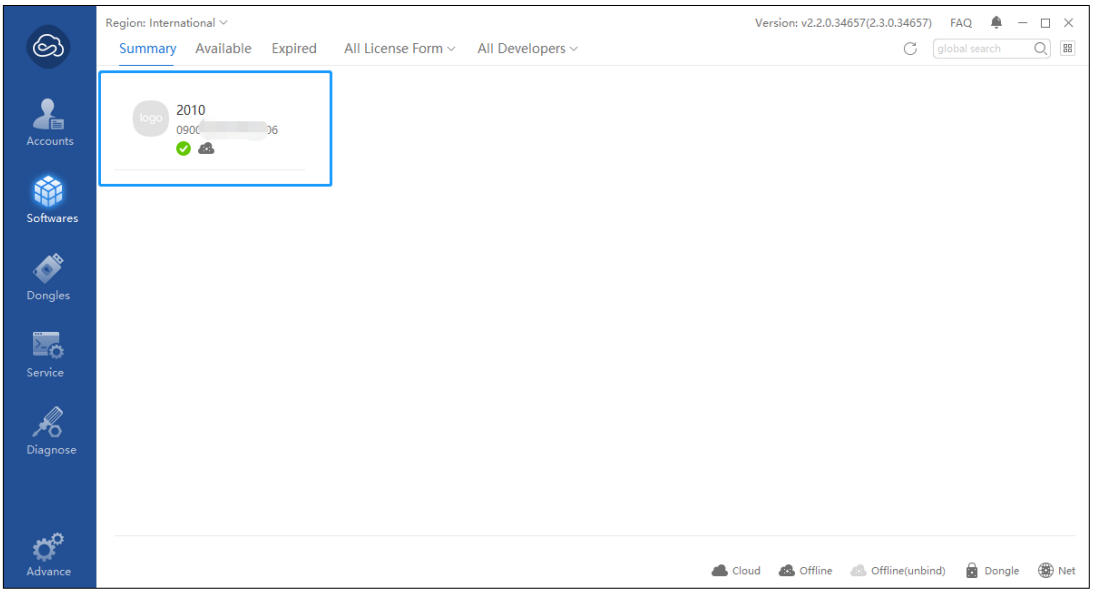


Figure 2-13

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

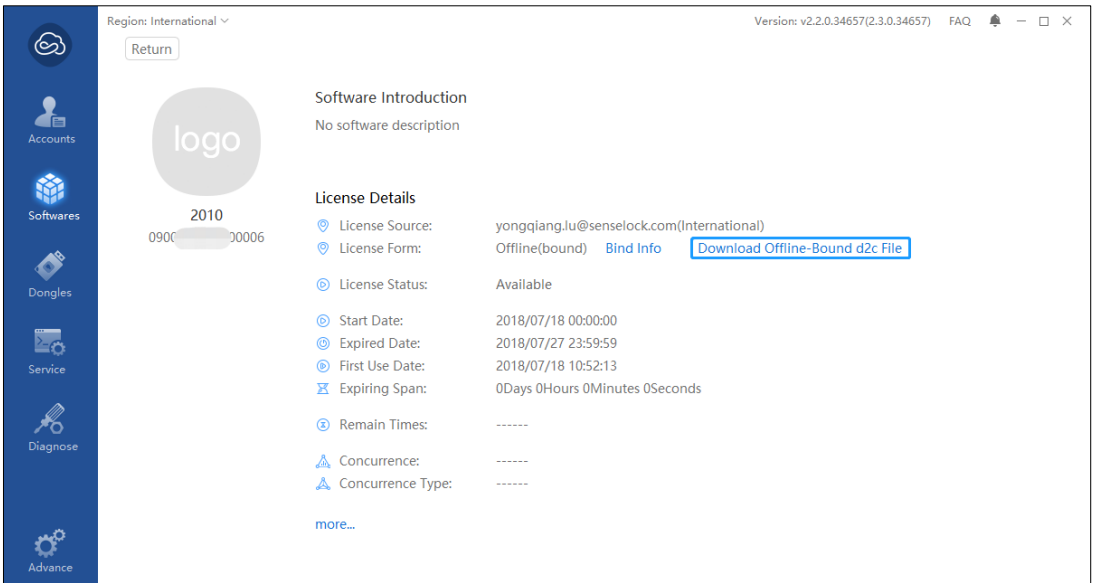


Figure 2-14

Click "**Download Offline Bound d2c file**"

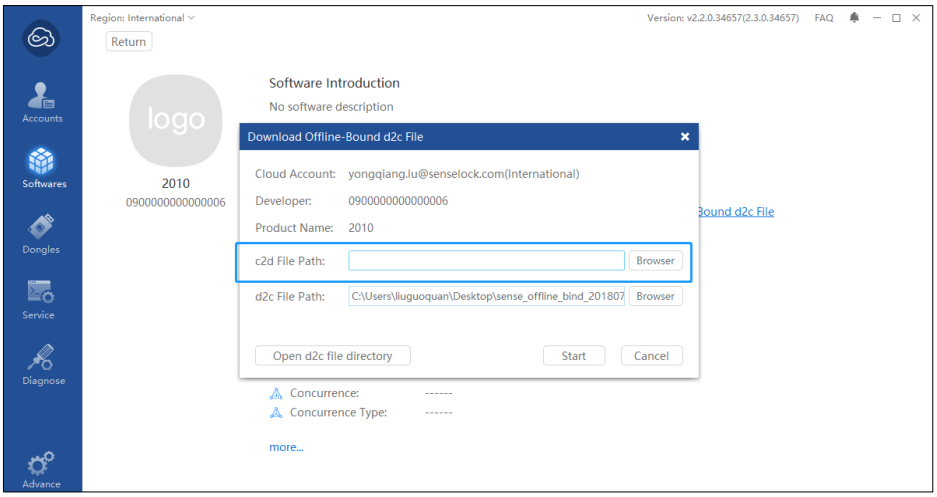


Figure 2-15

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

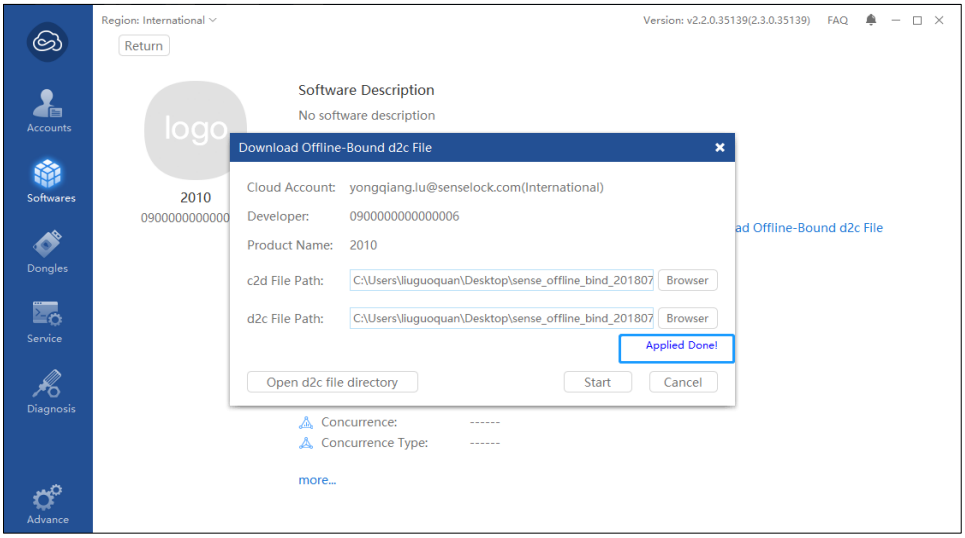


Figure 2-16

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.

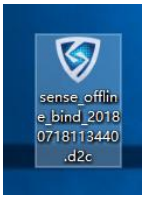


Figure 2-17

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.

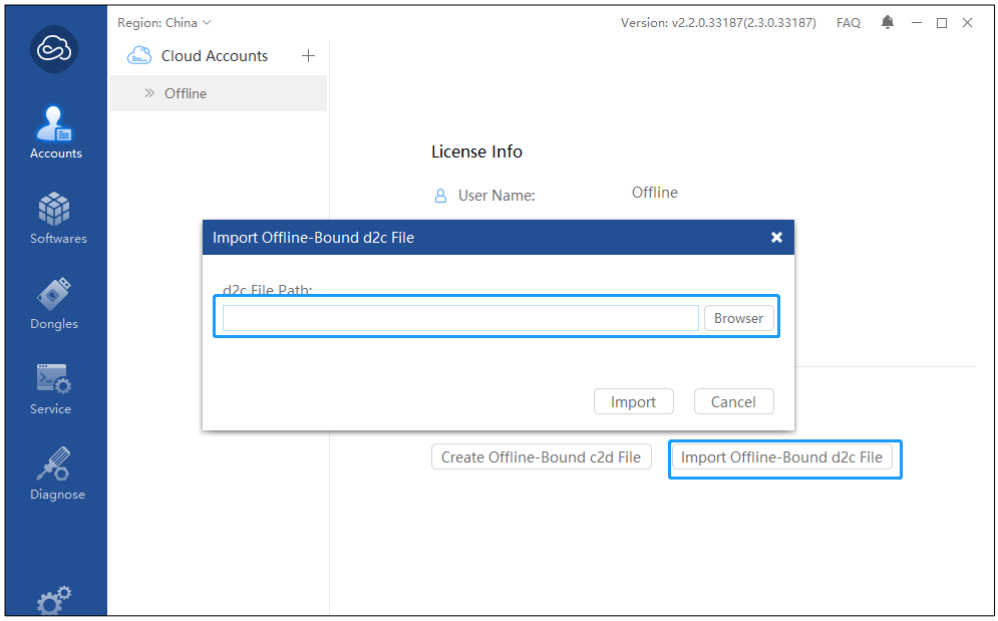


Figure 2-18

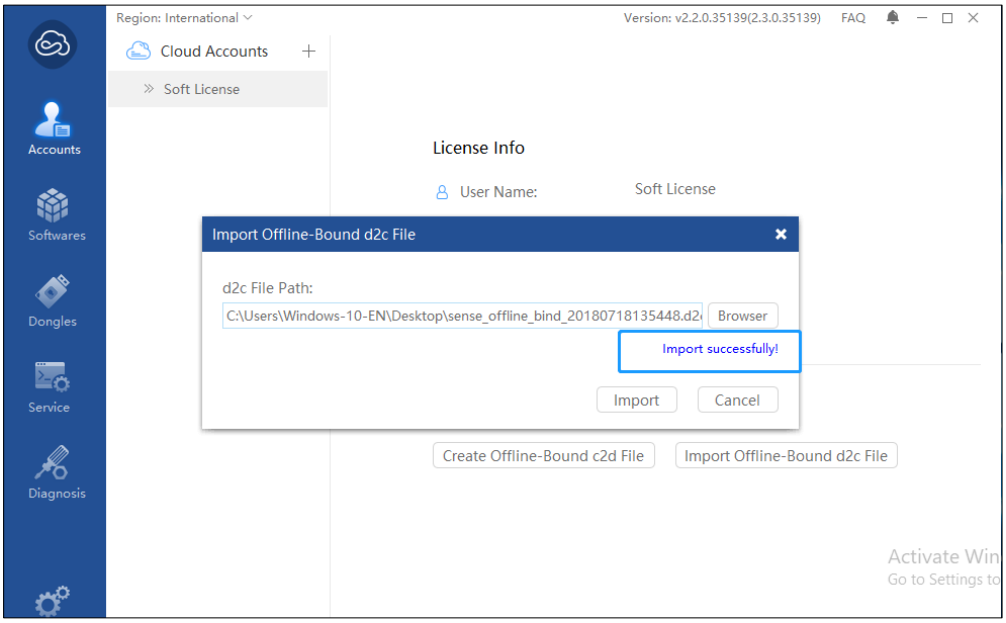


Figure 2-19

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 El5 dongle (For official user use dongle license)

If you purchased the Virbox Protector with a Virbox El5 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 subscripted. As the figure shown below:

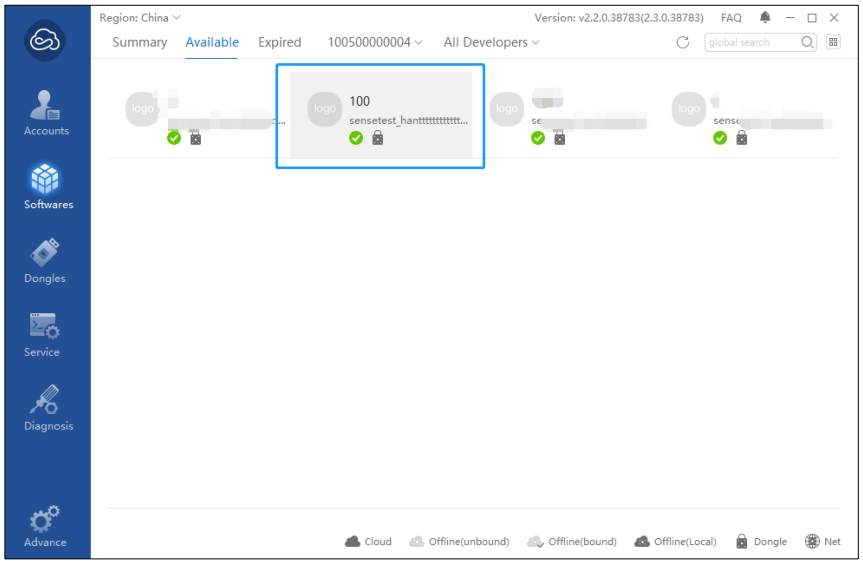


Figure 2-20

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

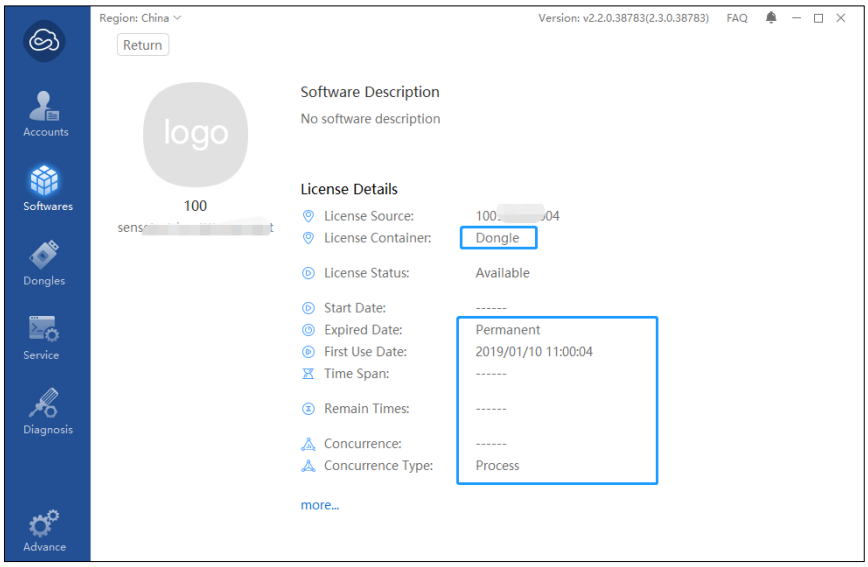


Figure 2-21

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 in this chapter.

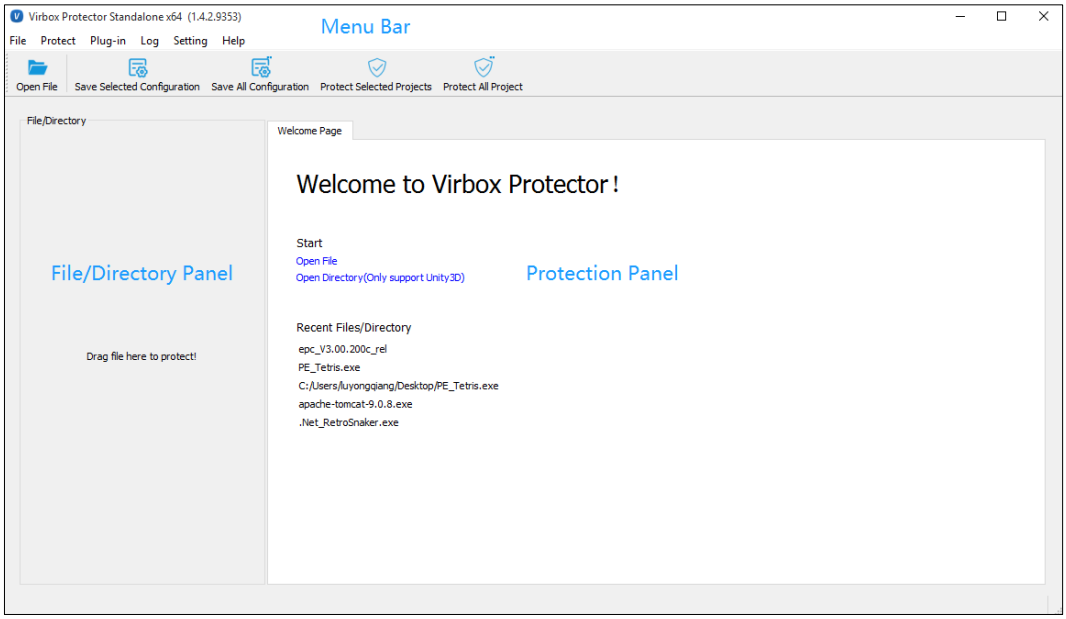


Figure 3-1

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. Apk, AAR and latest AAB format 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 Without Protection.

Parse all project

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

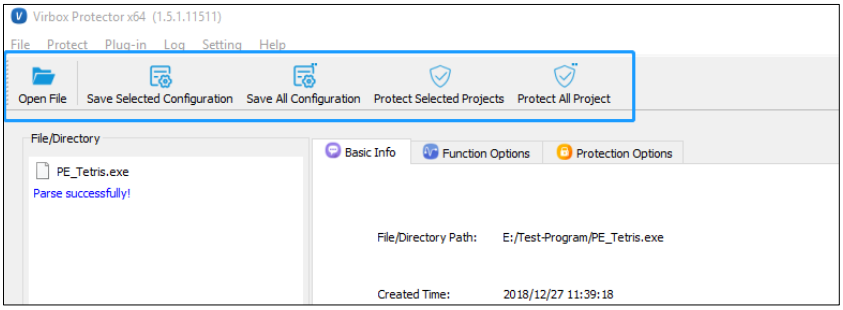


Figure 3-2

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.

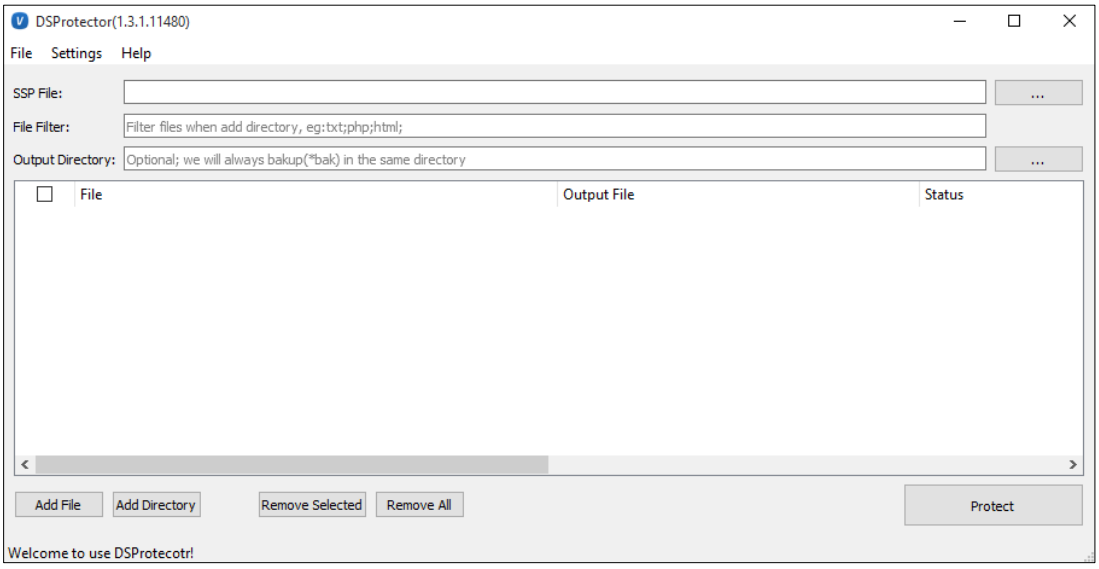


Figure 3-3

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

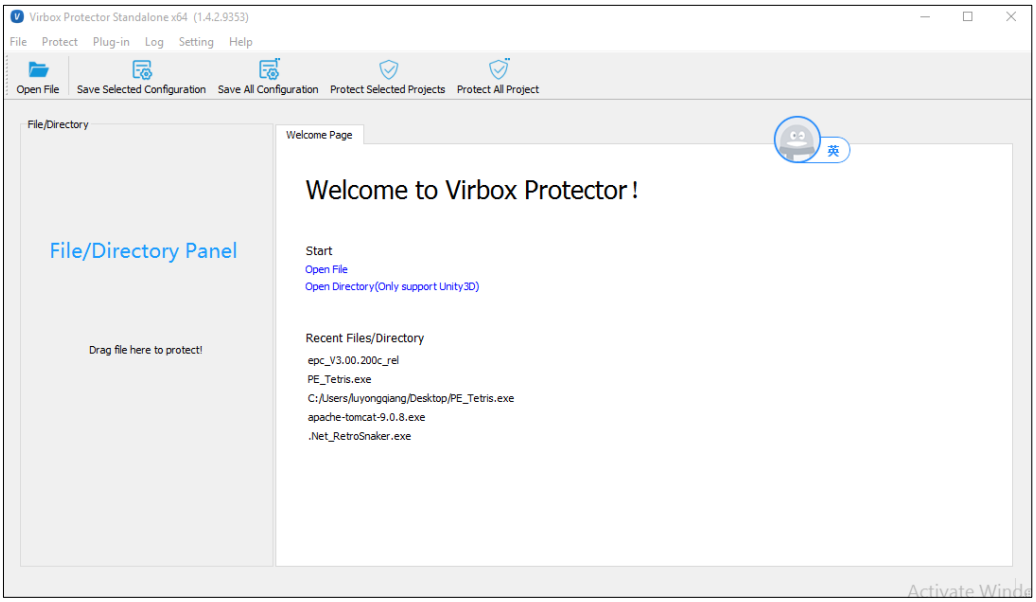


Figure 3-4

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

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 sub 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

Virbox Protector supports to protect the software application to the specified function's level and provides several protection mode for developer selection to protect the critical functions.

Add the Functions to be protected

Function Option, it 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 listed in the "***Function Option***" pane, 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 pane also.

As shown the figure below:

This pane 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.

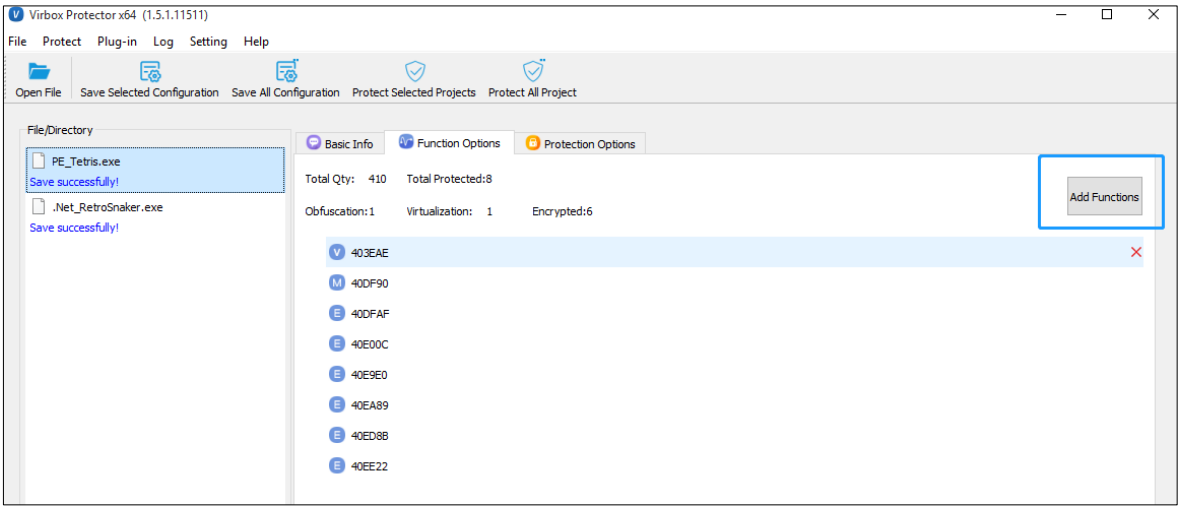


Figure 3-5

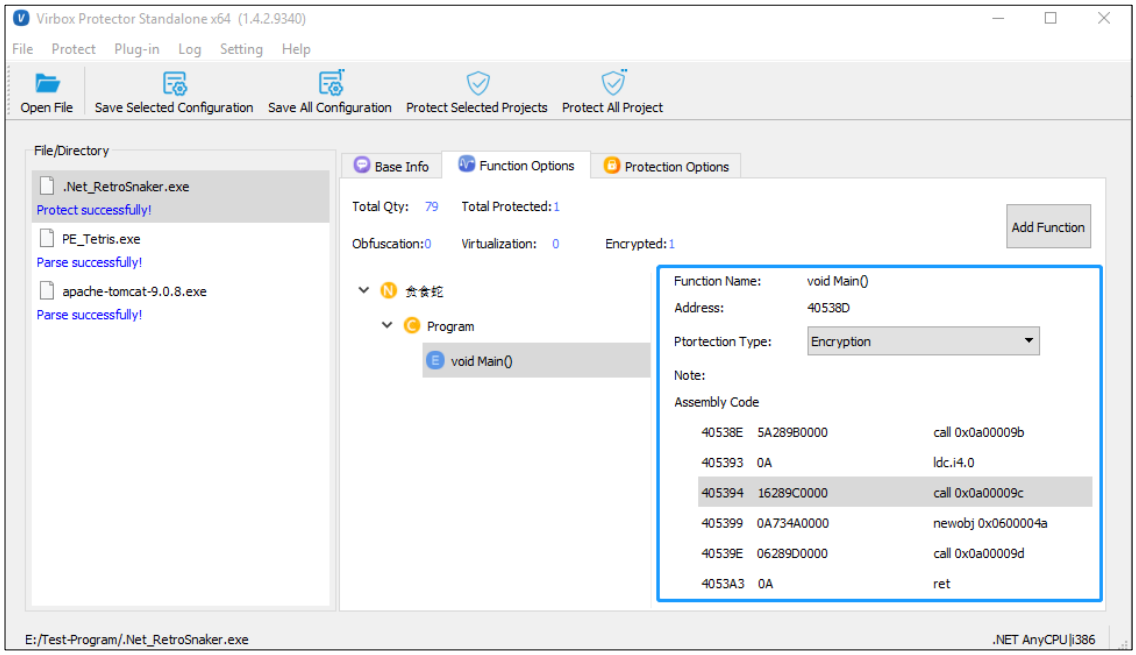


Figure 3-6

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 manage the balancing 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 With Protection/encryption.

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

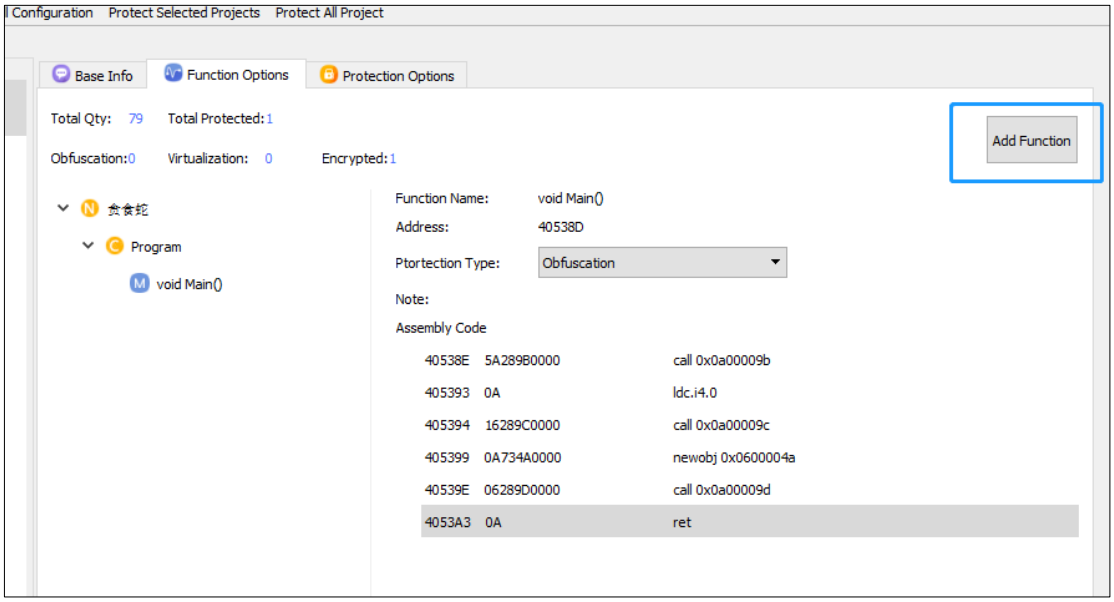


Figure 3-7

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

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

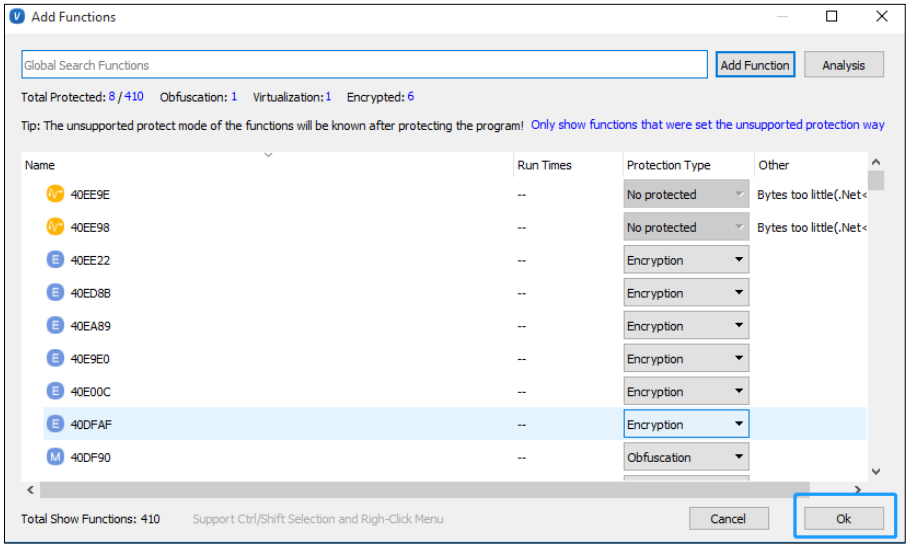


Figure 3-8

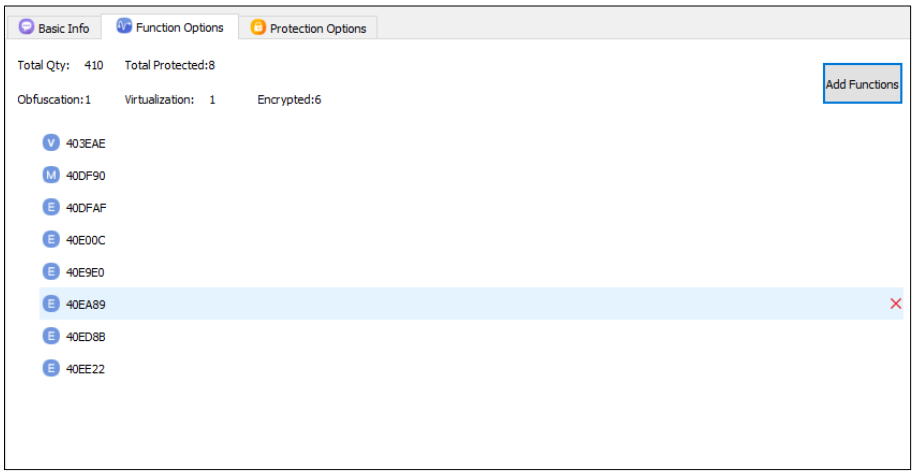


Figure 3-9

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

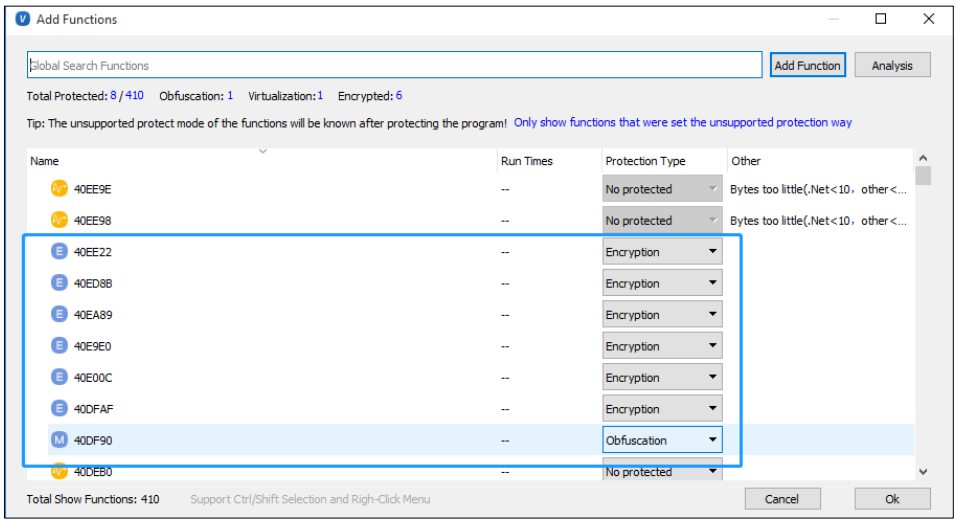


Figure 3-10

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

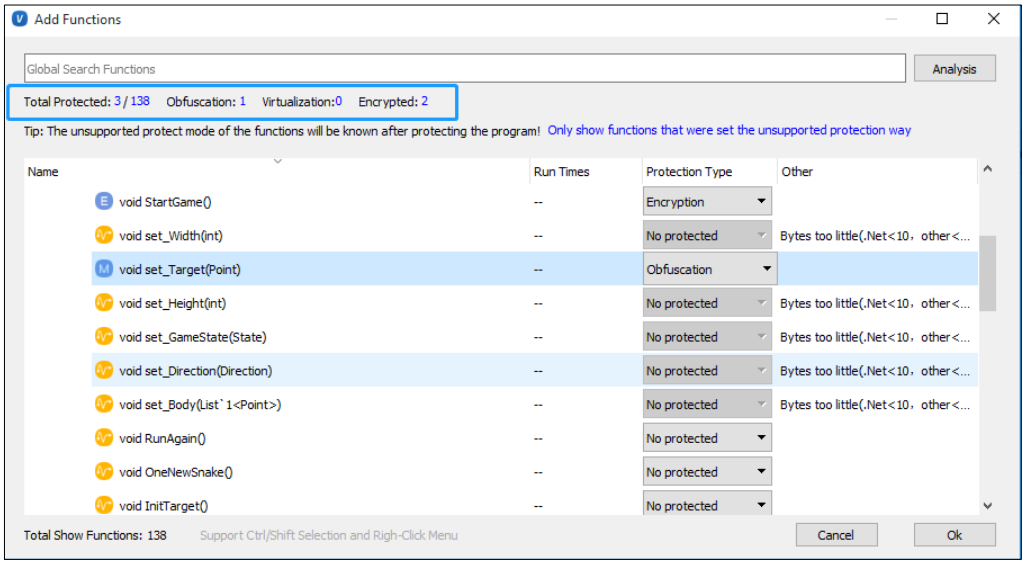


Figure 3-11

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 Protected**" option, since if you protect the functions which is called frequently, it will decrease software’s running performance when software is executed;

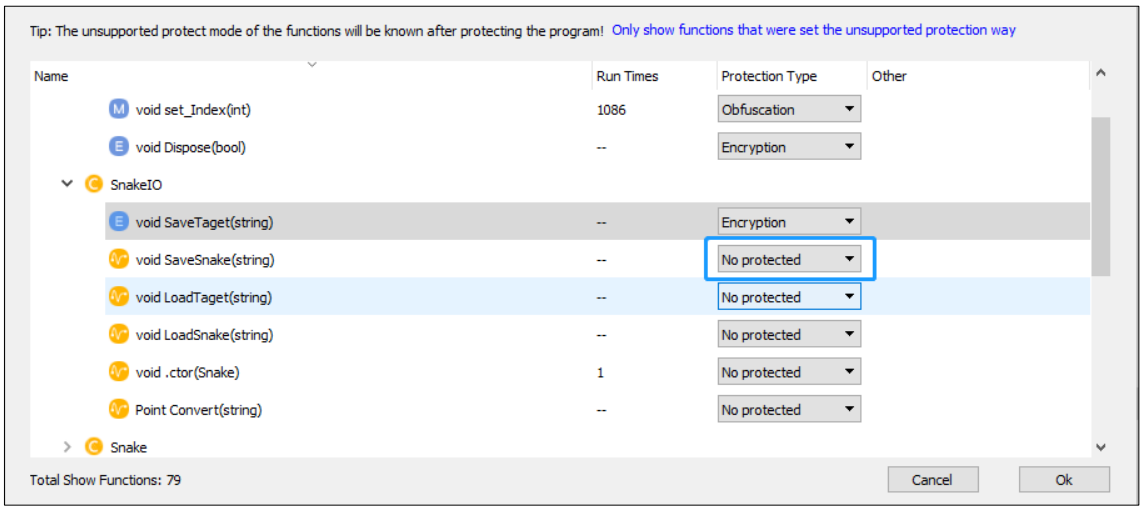


Figure 3-12

- 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 de compiling and make it more difficult to analysis the code.

  The Weakness:

  Partial Impact to the execution performance.

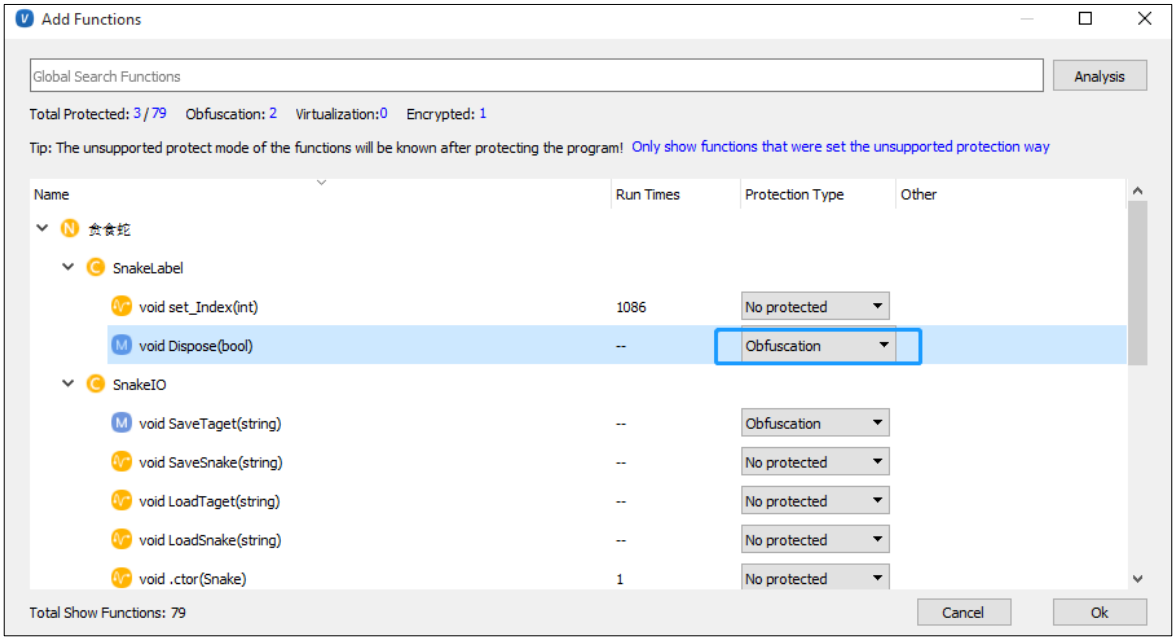


Figure 3-13

- 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

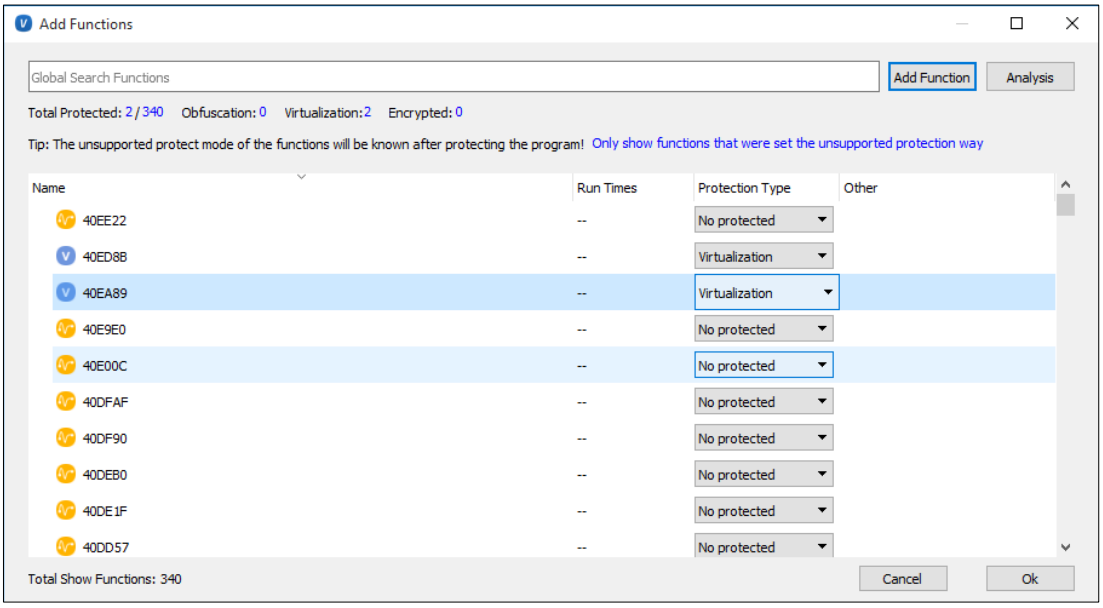


Figure 3-14

- **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.

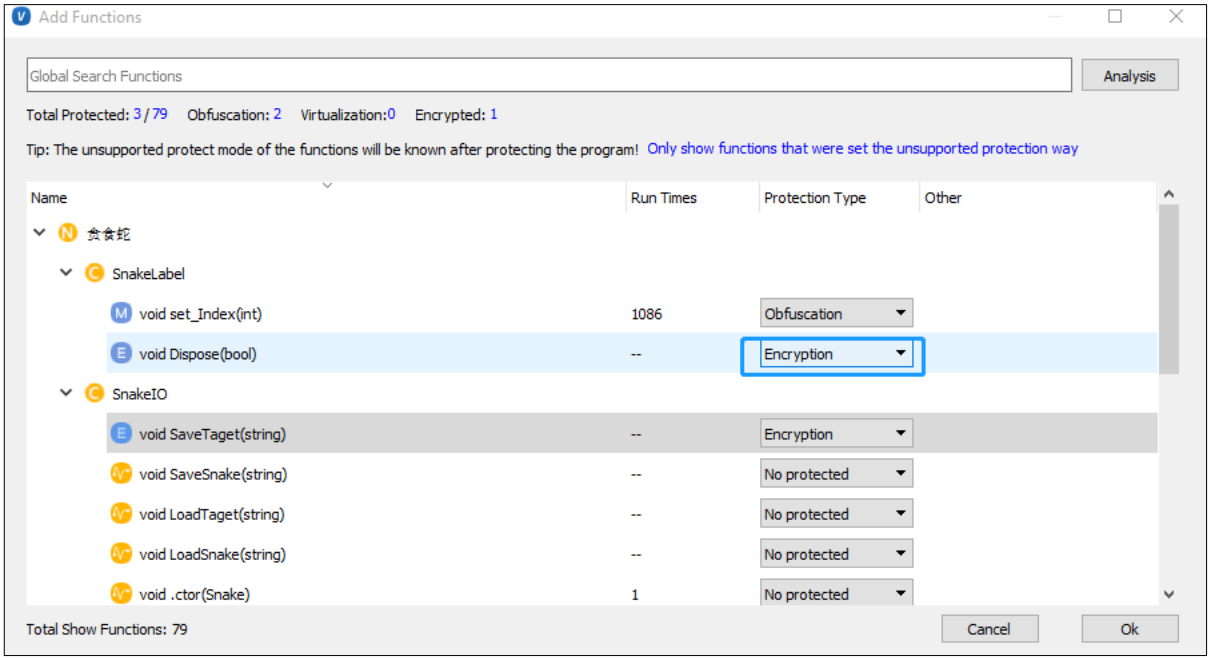


Figure 3-15

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.

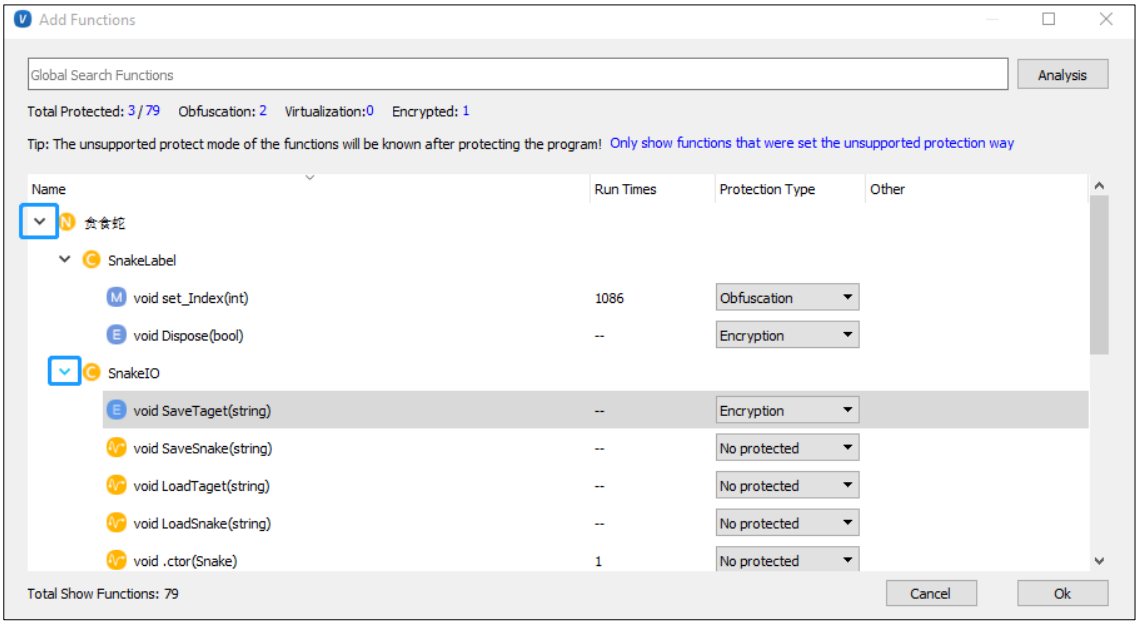


Figure 3-16

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.

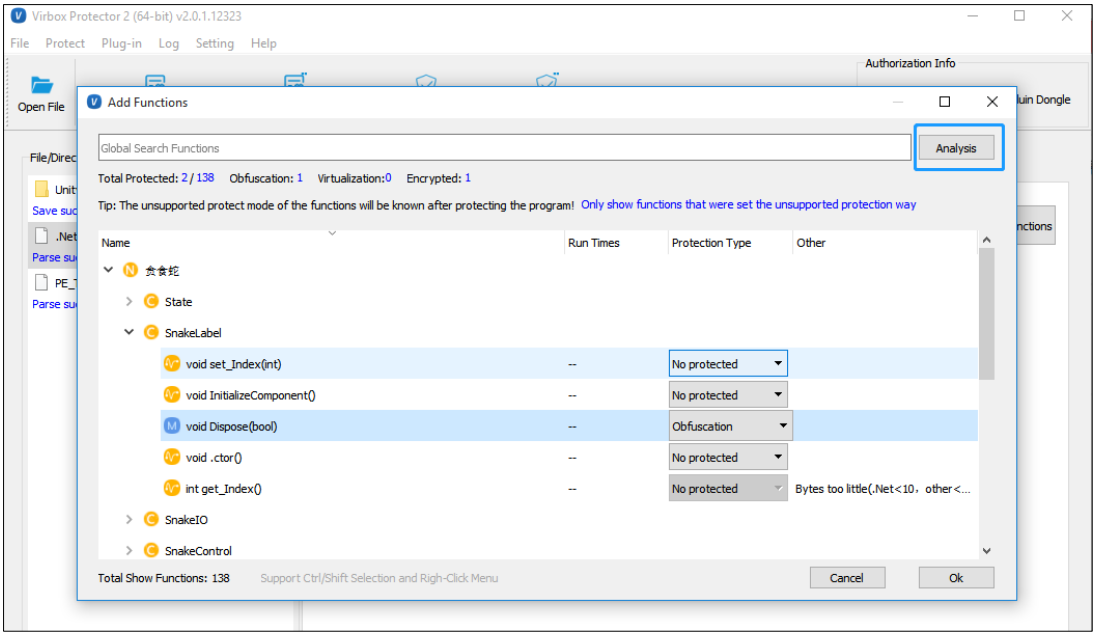


Figure 3-17

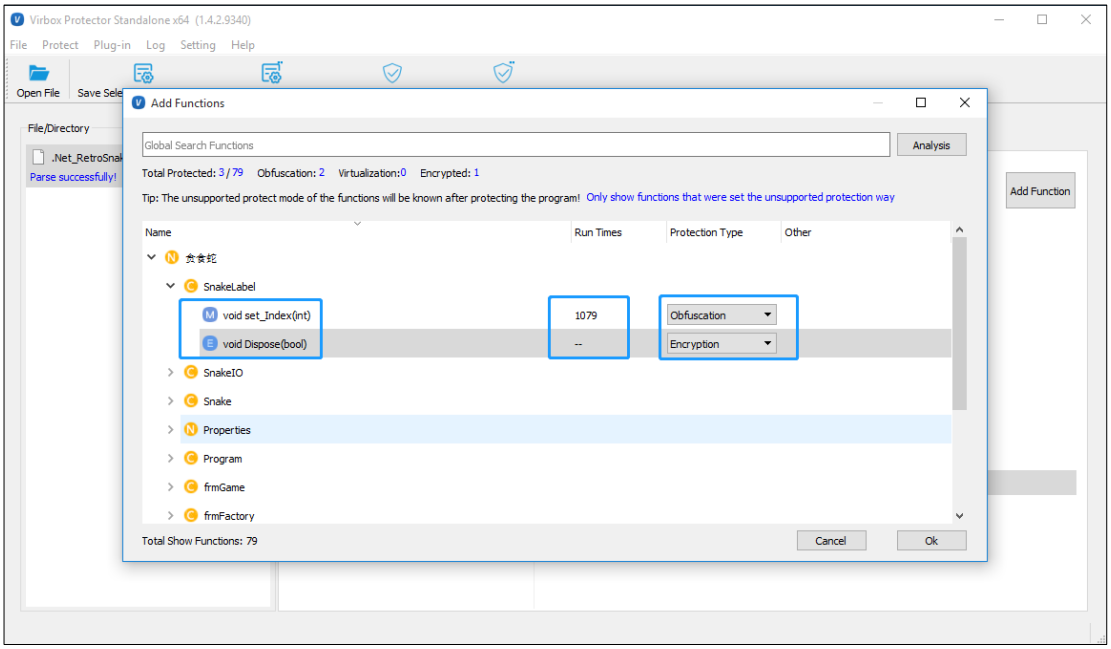


Figure 3-18

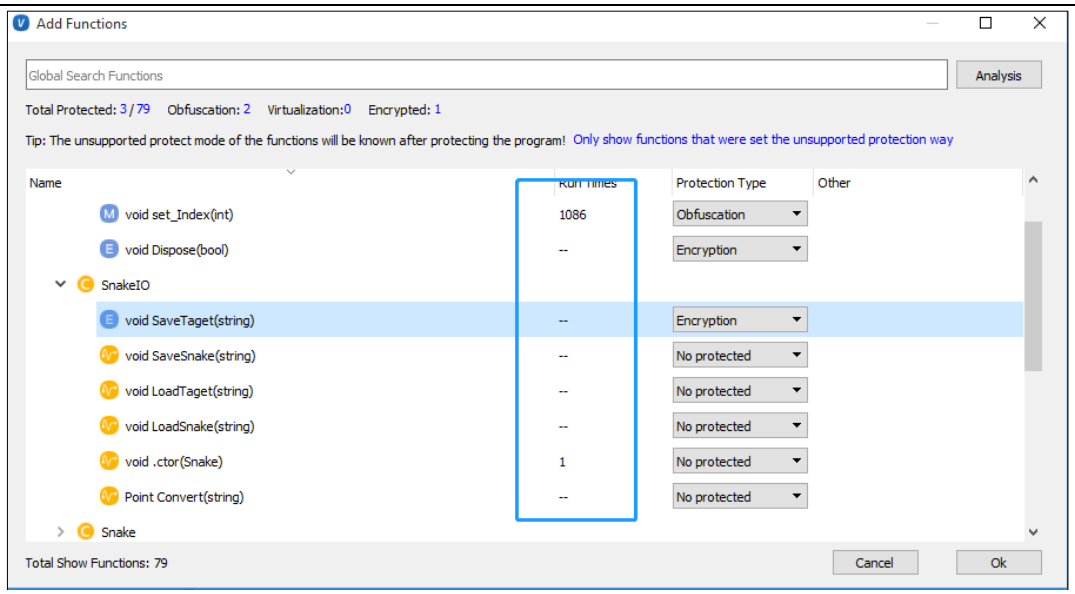


Figure 3-19

Function search:

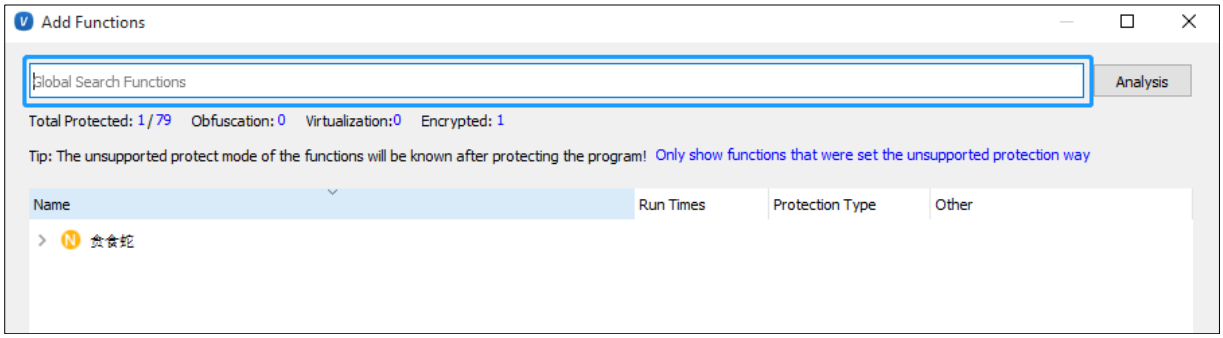


Figure 3-20

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 Options, developer can select and set protection option to your application in general, and also enable/disable the anti debug, Memo Check/Verification mode to defense the cracker to debug your application, and protect the data resource by using the Plugin unit: DS Protector. The Protection setting will be different for different program: The Protection option to PE (local program) and .NET application or Gaming software based Unity3D is different, Developer can select and setup these "**Protection Option**" in actual project.

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

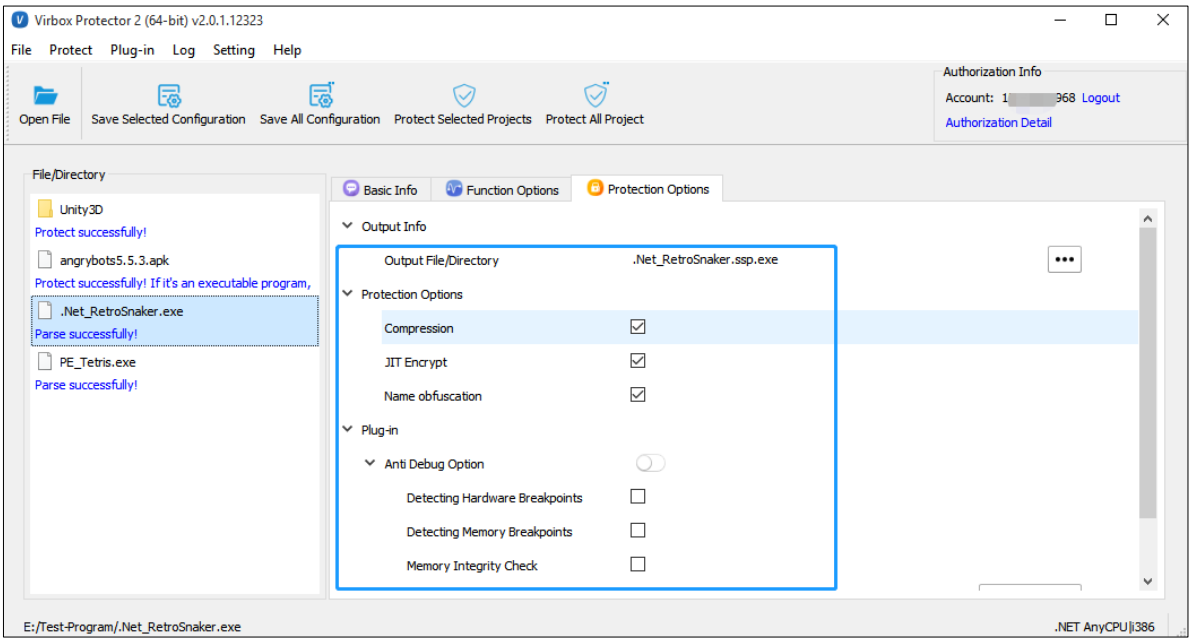


Figure 3-21

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

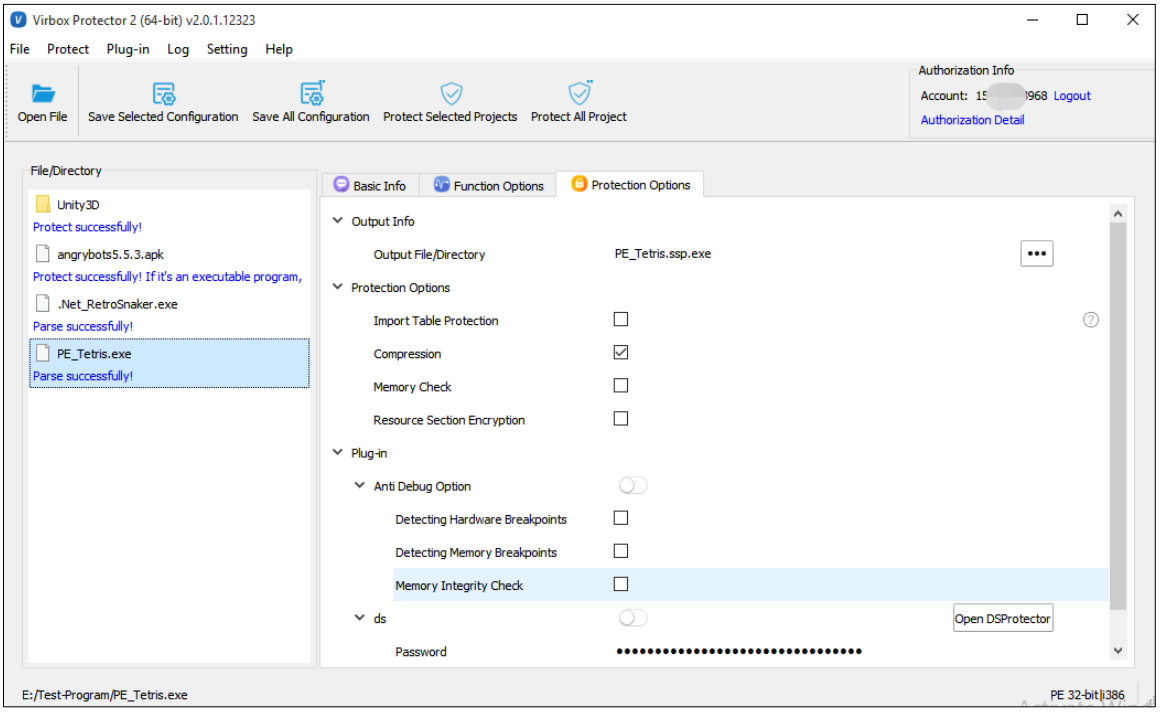


Figure 3-22

For Unity3D applications, following options will be displayed and selected:

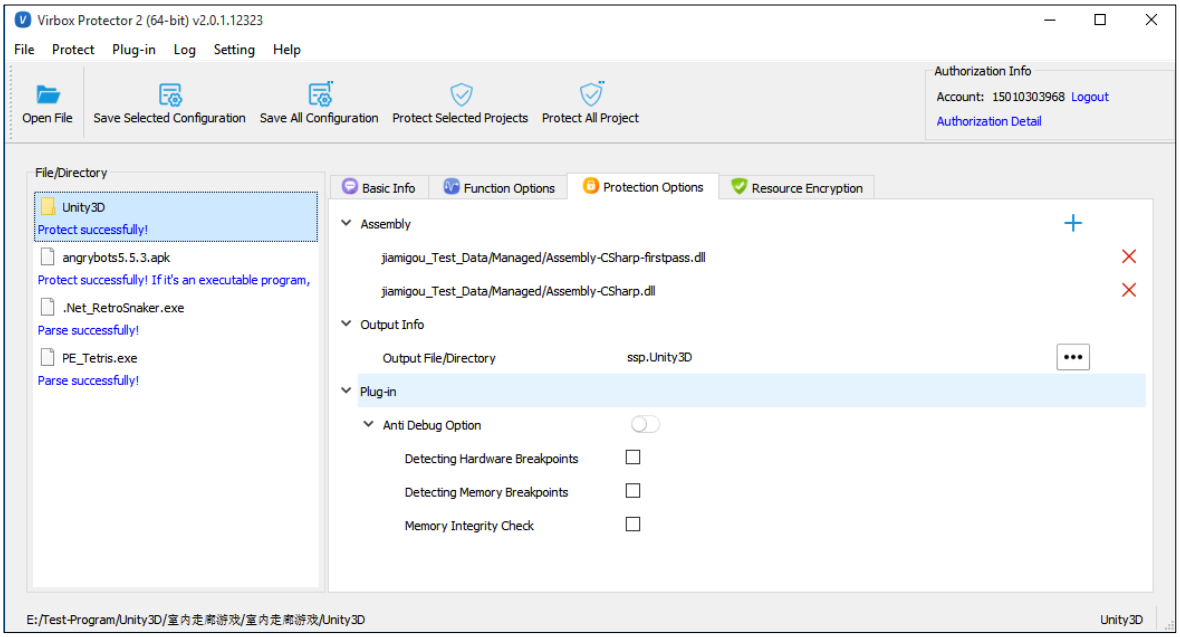


Figure 3-23

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. **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.

3. **Compression**: Compression means to compress the application With Protection and reduce size of application, it is also prevents static de compiling the software application by hacker who use static de compiling 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 de compiling 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.

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.

Note: Support .Net program only, not support IIS type program.

5. **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 patching or repackage your apps, 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 will be executed 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 when you call ***VBProtectVerifyImage,*** the memory check will be executed.

How to use this "Memory Check": drag your PE or ELF programs into the Virbox Protector, then this "Memo Check" Option will be showed in the "Protection Option" panel, click and select this option, then your program will be protected with the "Memo Check" option

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

6. **JIT encryption (applied for .NET):**

.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 de compiling and prevent the IL code being Dumped in memory.

JIT encryption will encrypt all of the method on default and enhance the security level of the source code With Protection.

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

7. **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.
   - 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.
   - Detecting memory breakpoint: This function will protect your software by exit the program if the program has been detected to be setting memory breakpoint.
   - 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 de compilingd, such as: gdb, IDA, etc tool.

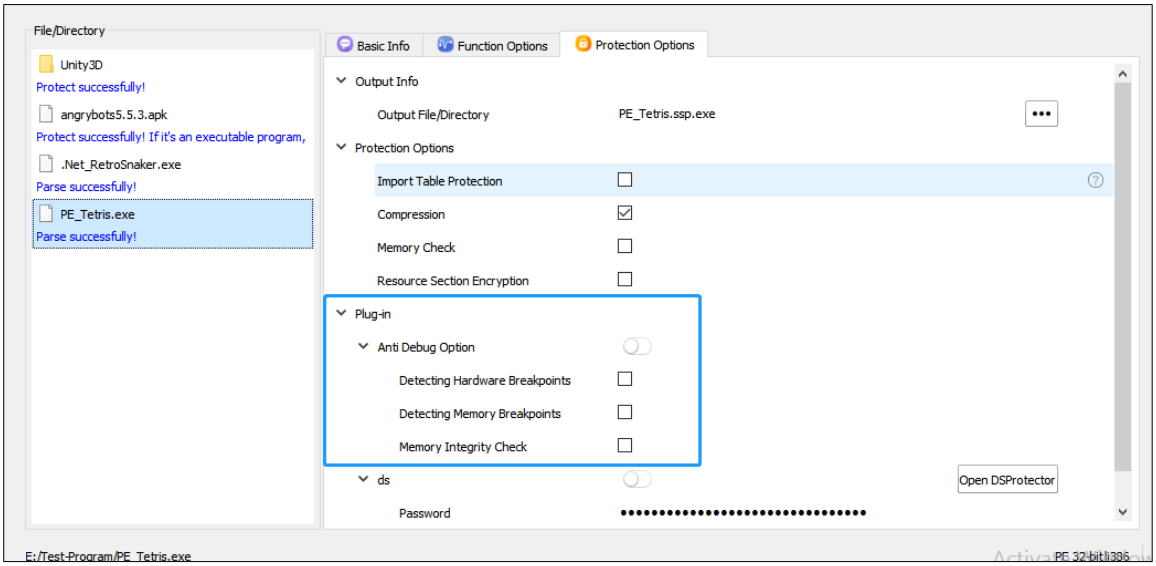


Figure 3-24

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:

Figure 3-25

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

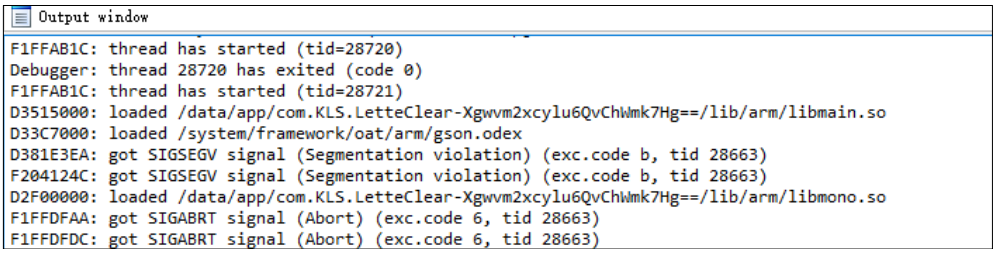


Figure 3-26

8. **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 it, 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.

9. **Ds (Data Source protection):** Encrypts the Data Source 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**".

- **DS button:** You can open DS Protector by clicking this button.
- **Password:** You can also set a password for the data resource protected by DS Protector: letters and numbers are supported, but it should not be longer than 64 characters.

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.

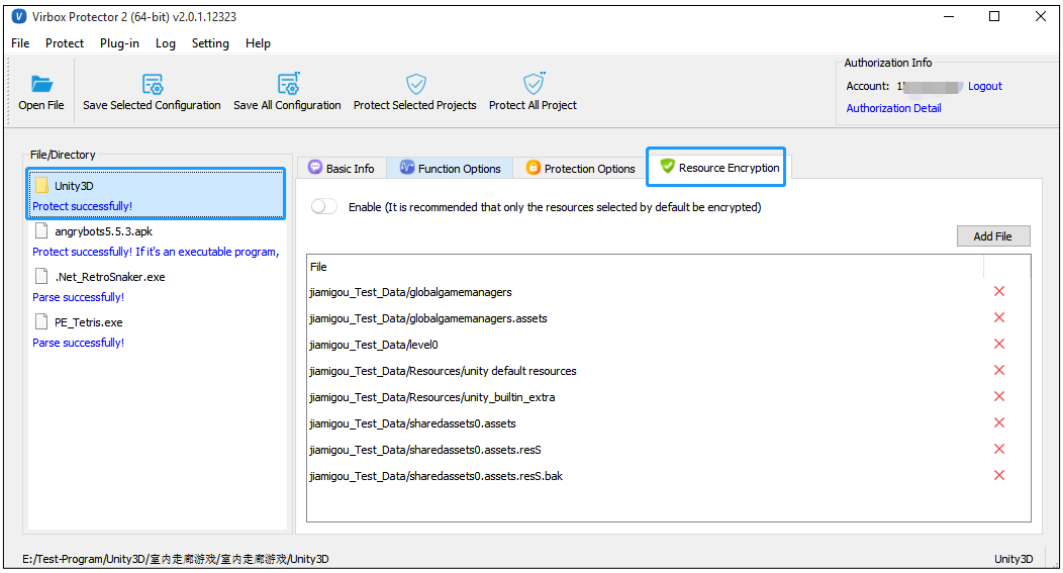


Figure 3-27

3.3.2.5 Status bar

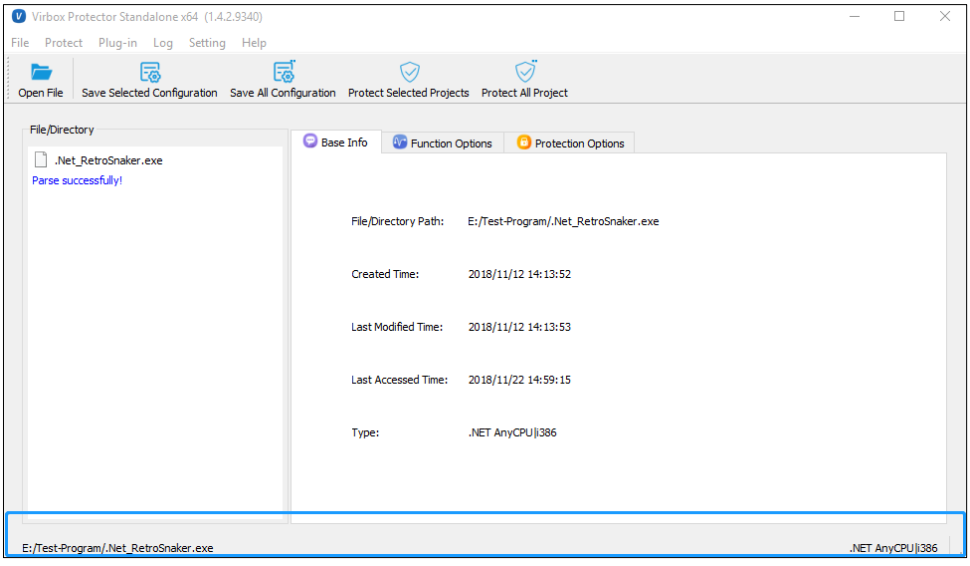


Figure 3-28

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 Native application

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:

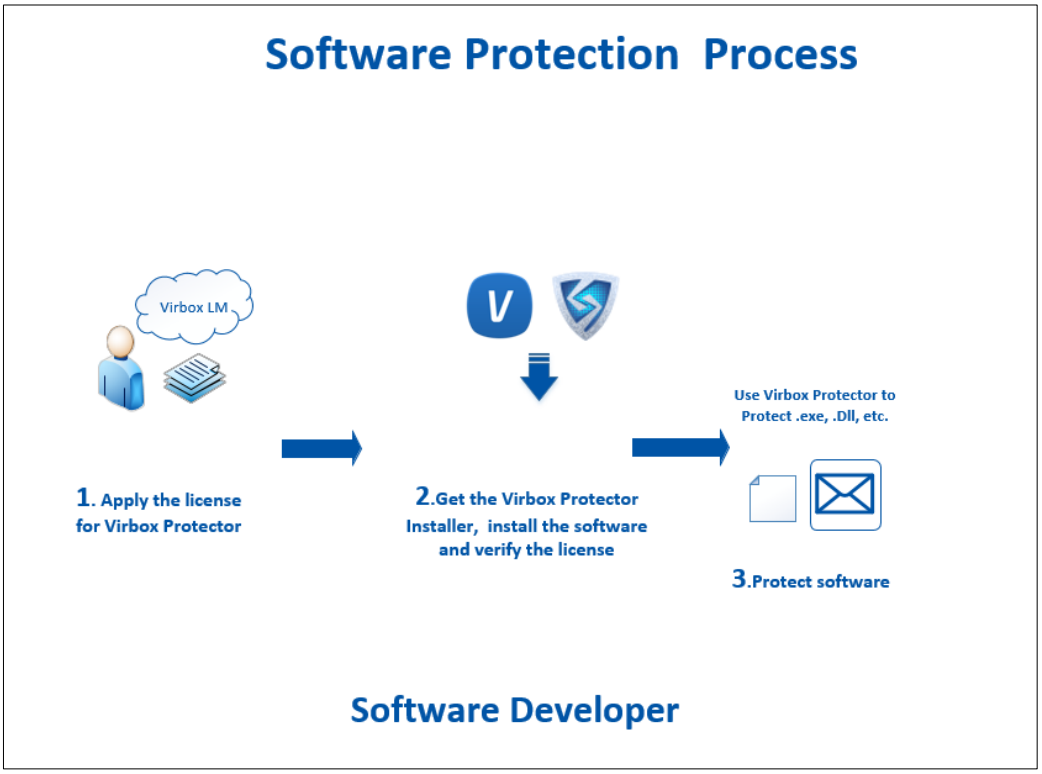


Figure 4-1

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 desktop 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, or Media player, etc.);
  - Use DS Protector to protect the source code or data resources (.py, media files, etc.);

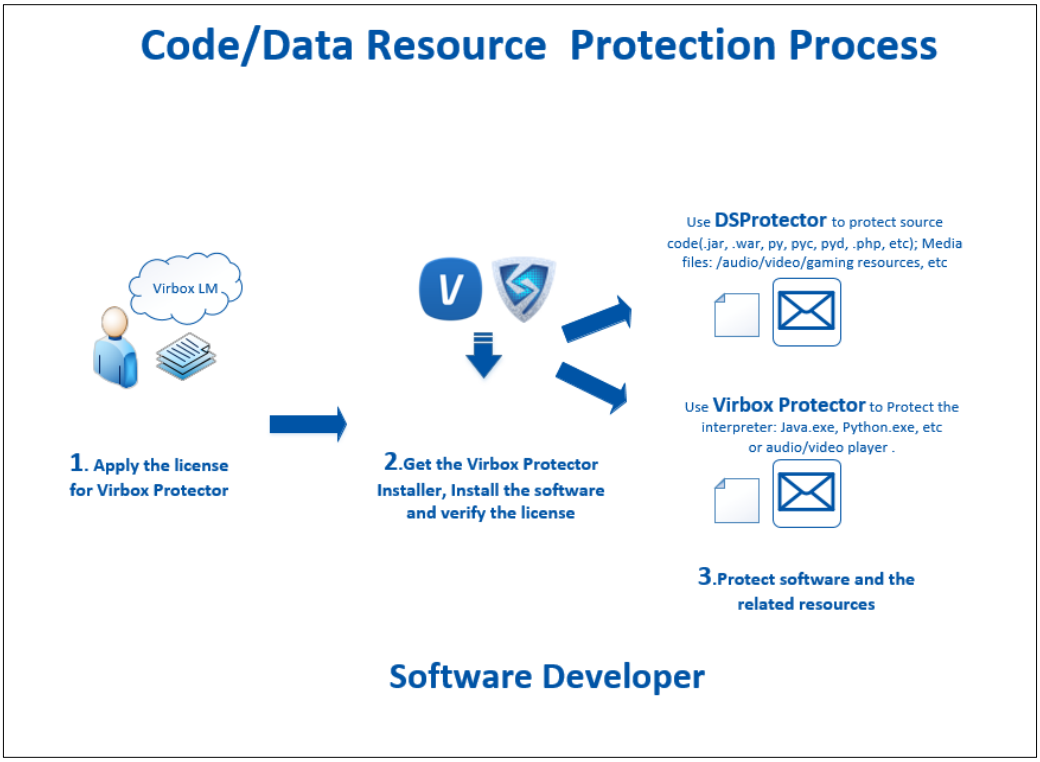


Figure 4-2

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:

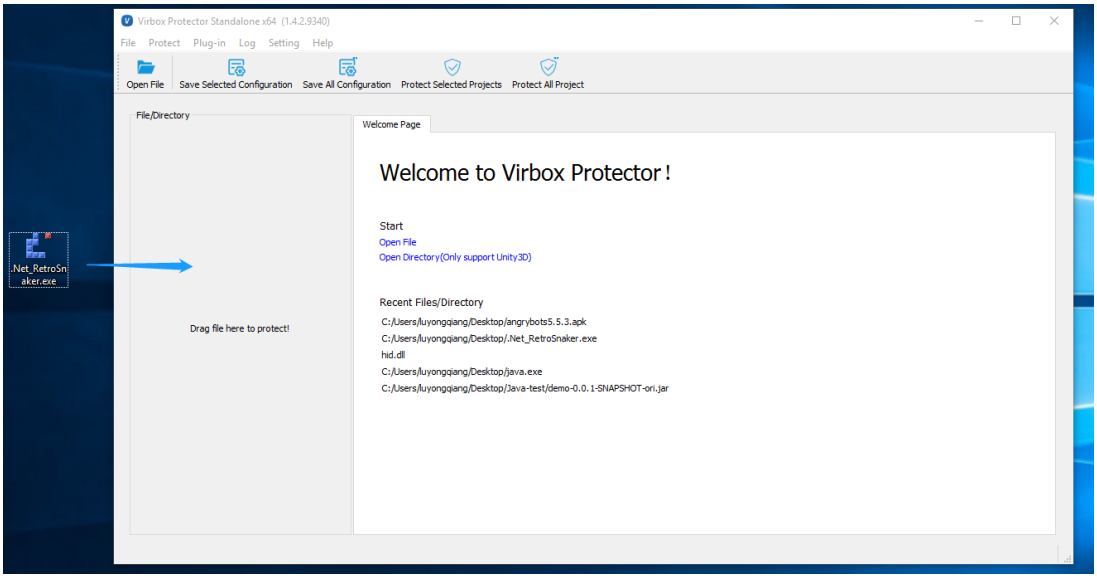


Figure 4-3

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

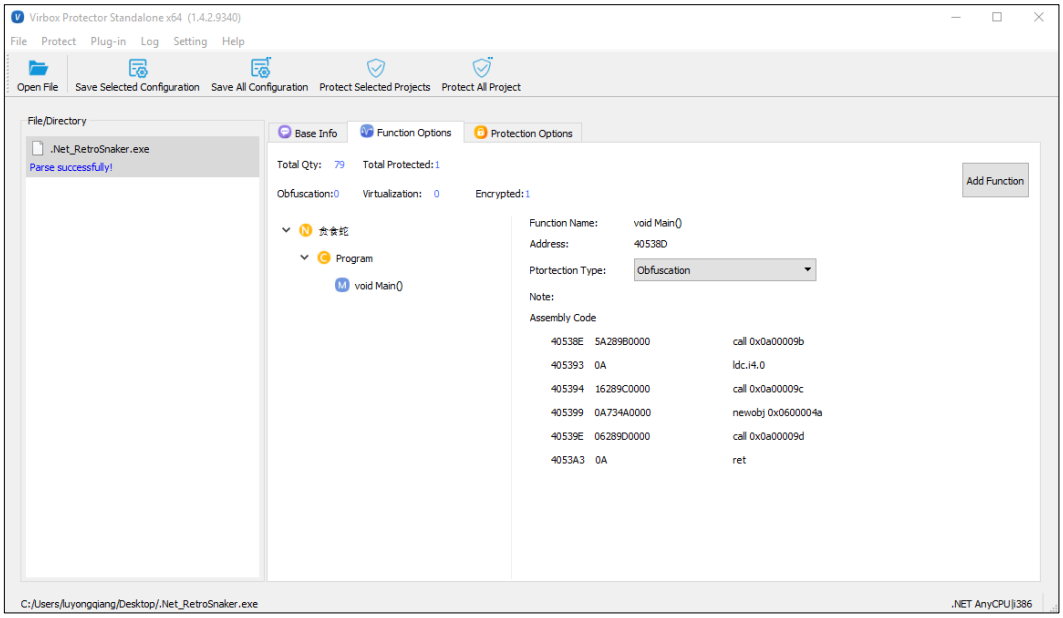


Figure 4-4

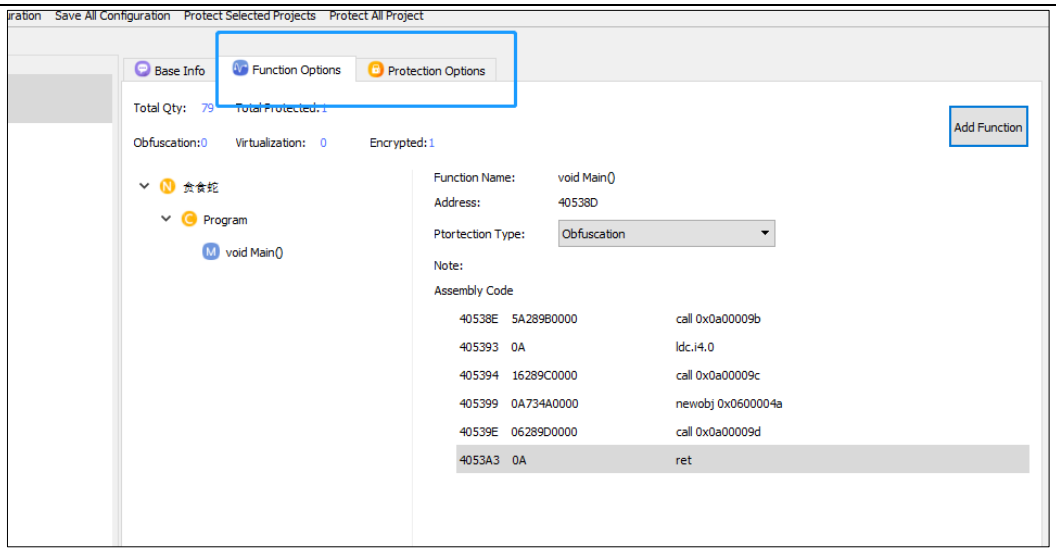


Figure 4-5

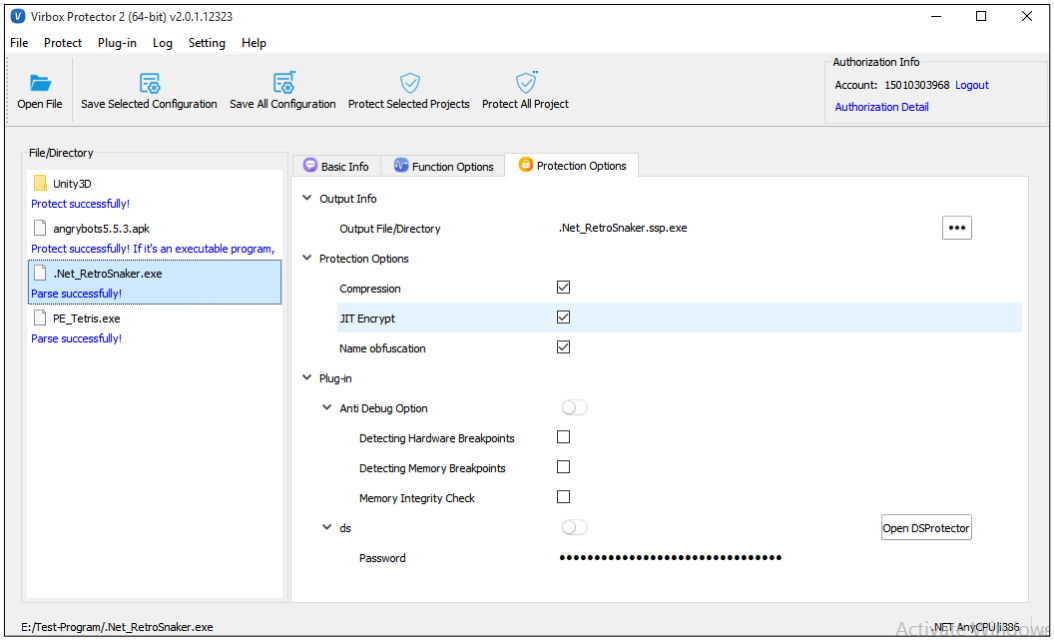


Figure 4-6

Select different technology to encrypt your function/method: **Virtualization, Advanced obfuscation, Code Encryption**. And also you can protect your program in overall by selecting these options: **Import table protection, 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**" tab, and you can protect the functions of your software by selection from the "**Function Options**" tab. 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 files.

5.1.1 *Protection Option*

Following protection option can be click and selected to protect local executable file in ***Protection Option*** Panel, Developer may use below options to implement the fundamental protection to software applications:

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 Appending data extension

What is Appending data extension?

Appending data is the data (video or database) being combined with the original PE executable by the compiler or packer tool, these data will be read by the original executable when executed and it 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 appending data into the protected program, error may happen when the program executed.

Hook is used in appending data extension to read the appending data normally, and encrypt the appending 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 de-compiling 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:

Without Protection:

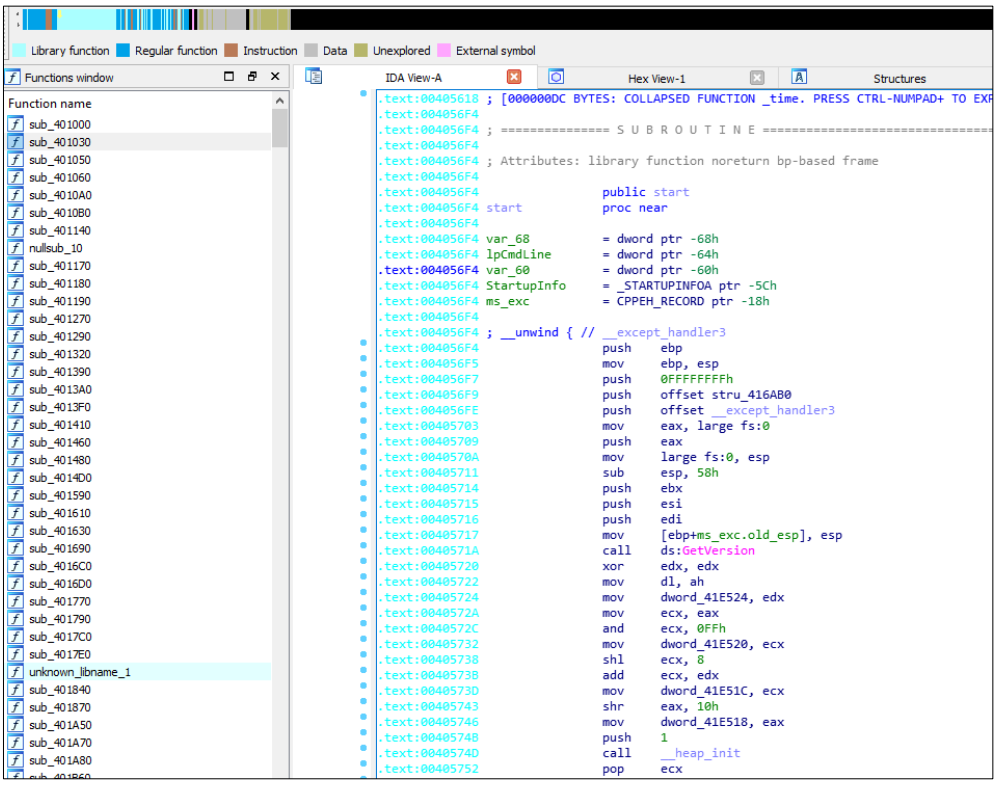


Figure 5-1

With Protection:

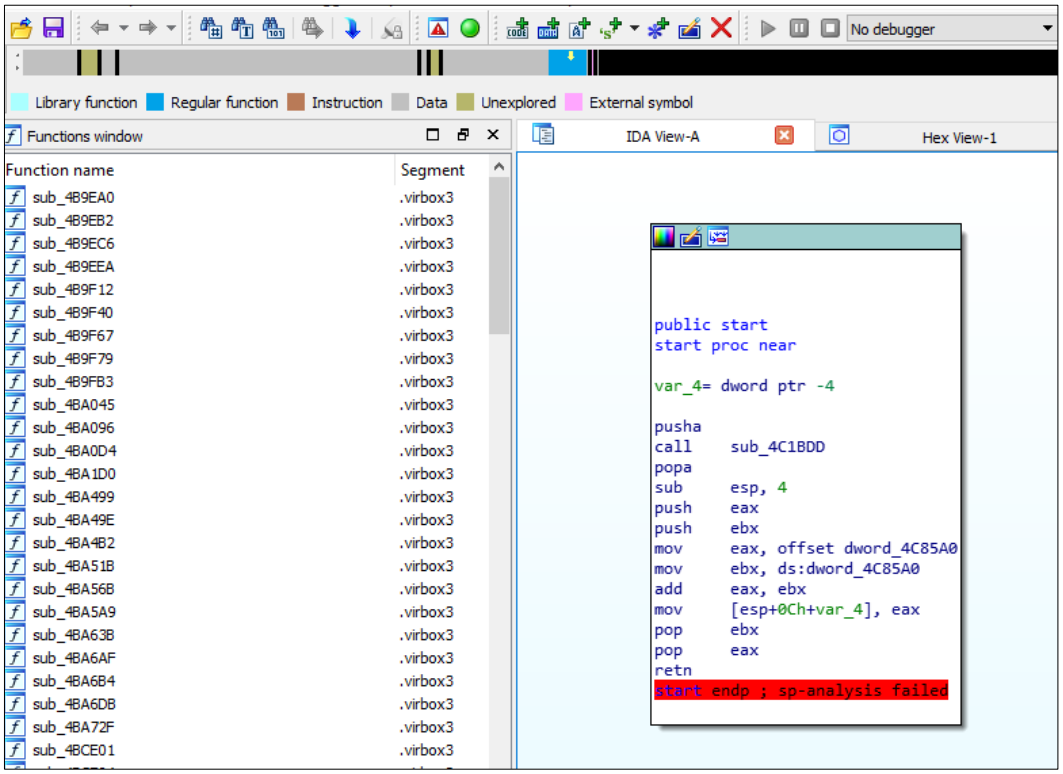


Figure 5-2

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.

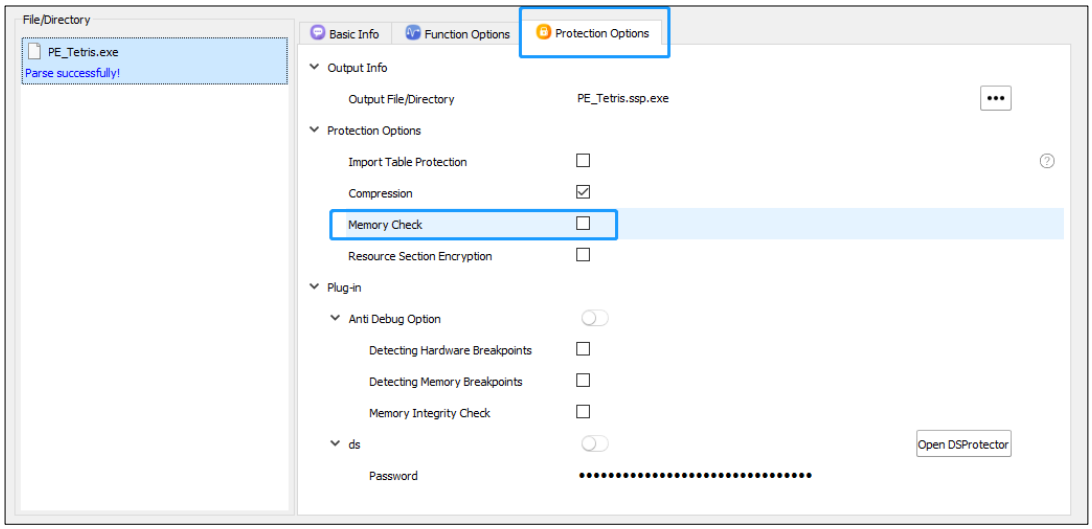


Figure 5-3

5.1.2 Functions Option: Protect the critical & Specified Functions

Following protection mode can be click and selected to protect the specified functions in your application

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 de-compiled/disassembled and make it more difficult to analysis the code.

The Weakness:

Execution Performance may impact.

Protection Comparison:

Without 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 5-4

With 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
.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, 1102B200h
.text:004144DE                 jle     short loc_41447C
.text:004144E0                 mov     cl, 10h
.text:004144E2
```

Figure 5-5

ARM architecture source code:

Without Protection

Figure 5-6

With Protection:

Figure 5-7

5.1.2.1.1 Anti-Run Trace:

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

1. Run Trace is the function provided by debugging tools, 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.

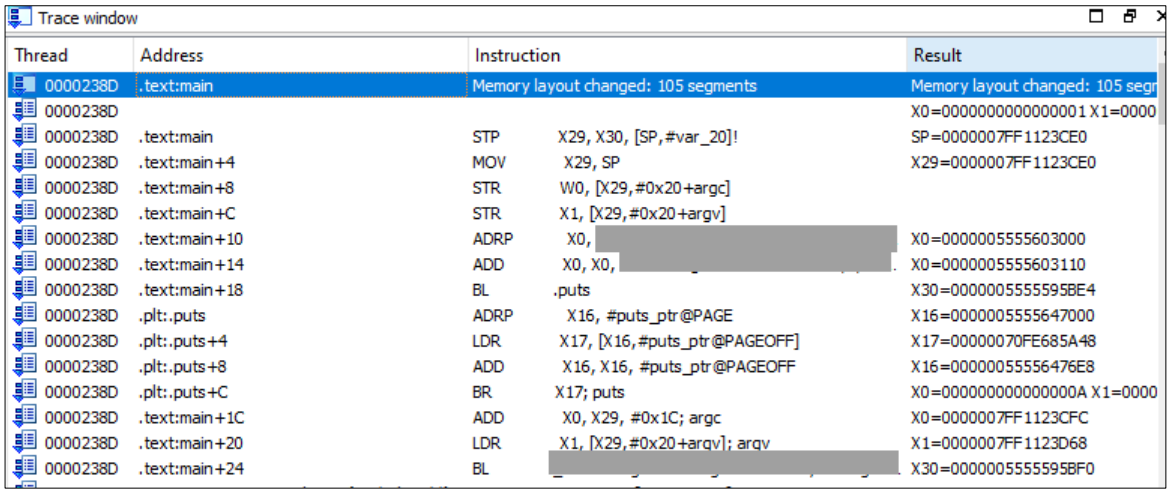


Figure 5-8

5.1.2.2 Code Virtualization

Protection Mechanism:

Virbox Protector will compiles instructions into virtual instructions executed in the randomly generated 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.

Note: Applied for X86, X64 and ARM architecture program.

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 will 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:

Without Protection:

```
.text:004056F4 start           proc near
.text:004056F4
.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    0FFFFFFFFh
.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 5-9

With 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, 0B8h, 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, 0CBh, 7, 0BBh, 7, 0AEh, 89h, 61h, 98h, 9Bh
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 5-10

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

Virbox Protector provides 2 ways for 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 functions 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 for 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:

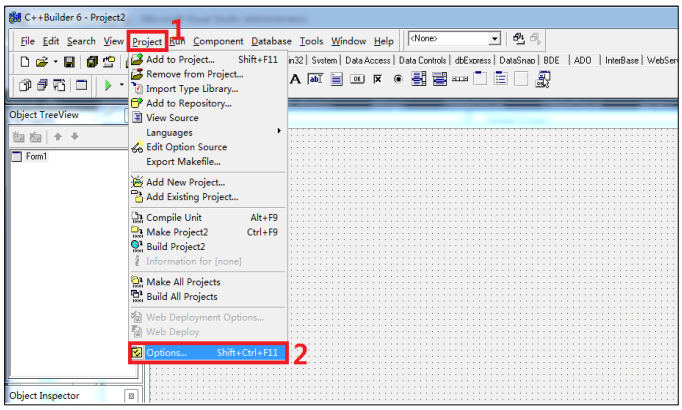


Figure 5-11

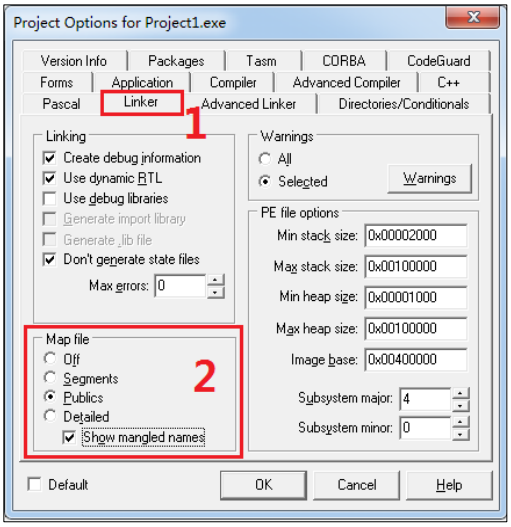


Figure 5-12

5.1.3.1.2 Generate map file for VC program

Project settings as shown below:

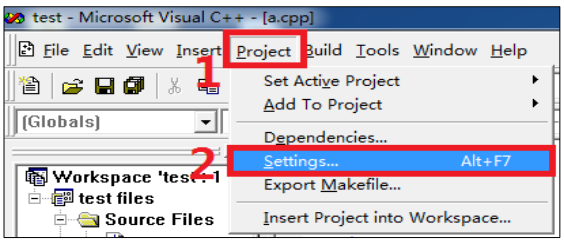


Figure 5-13

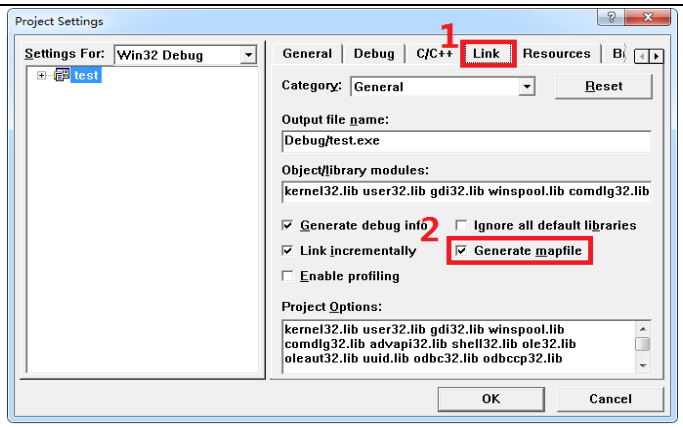


Figure 5-14

5.1.3.1.3 Generate map file for VS program

Project settings as shown below:

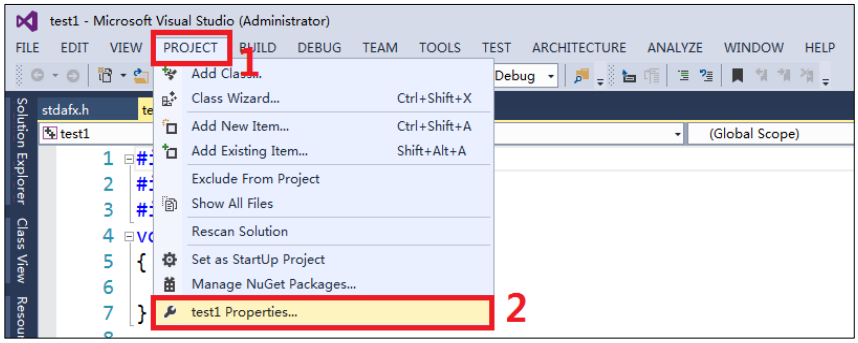


Figure 5-15

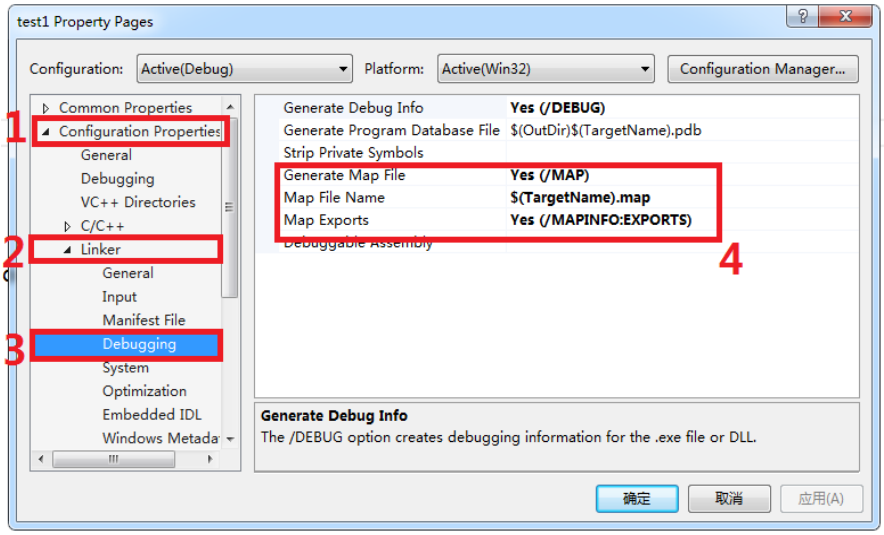


Figure 5-16

5.1.3.1.4 Generate map file for Delphi program

Project settings as shown below:

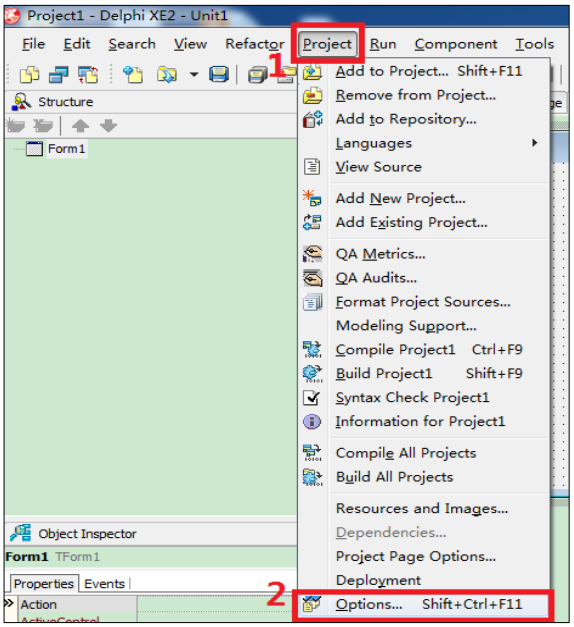


Figure 5-17

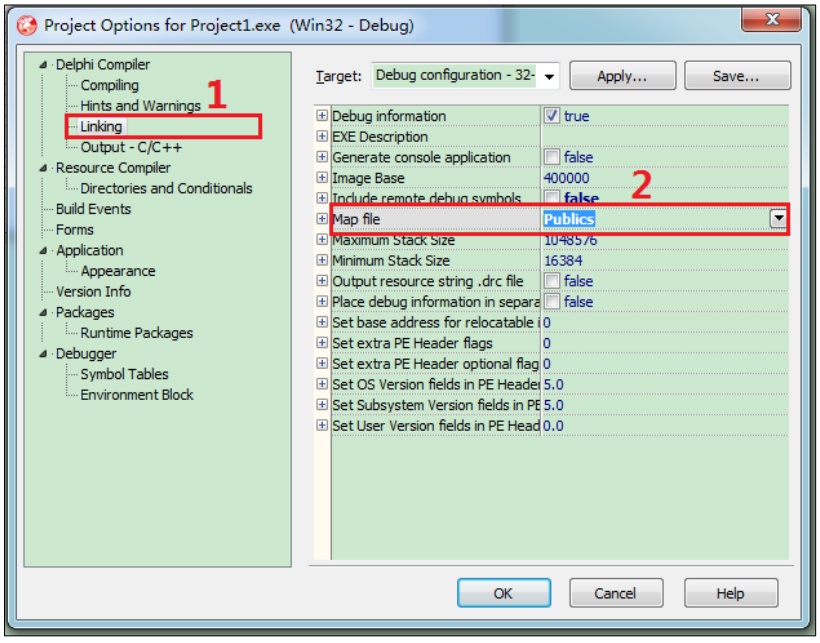


Figure 5-18

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.

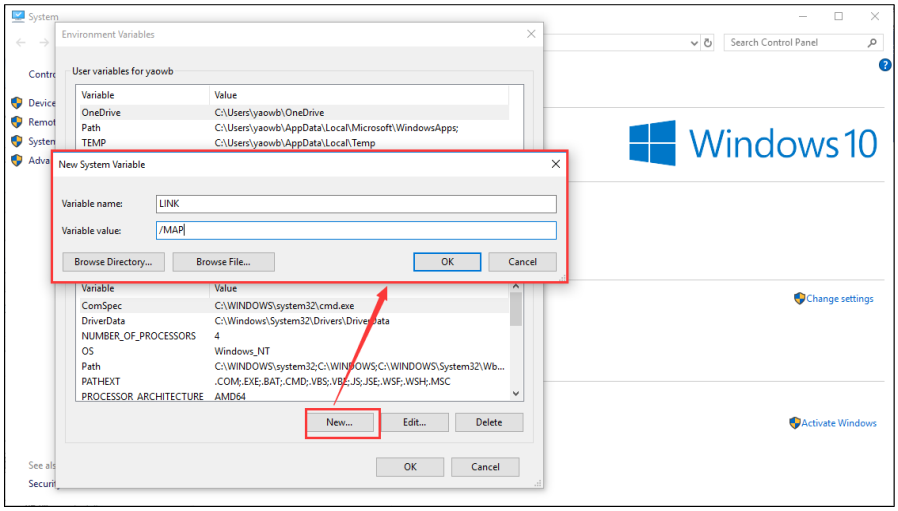


Figure 5-19

5.1.3.2 Using the SDK label API to mark the critical functions

This function and related SDK label API helps developer to mark and locate the important function that need to be protected by using SDK label API. 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 recognize & locate the critical code and protect those functions which use SDK labeled.

5.1.3.2.1 Using SDK Label and Protect your functions

Till now Virbox Protector supports to add these kind of label to mark the functions you want to protect:

VBProtectBegin: Normal & conventional 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. The SDK label can only be loaded statically and not supported to be loaded dynamic link lib (i.e. 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 Encryption of String: How to encrypt and decrypt the sensitive 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 Decryption supported:
   - String Decryption:

     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 be encrypted 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 via Virbox Protector GUI tools:

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".

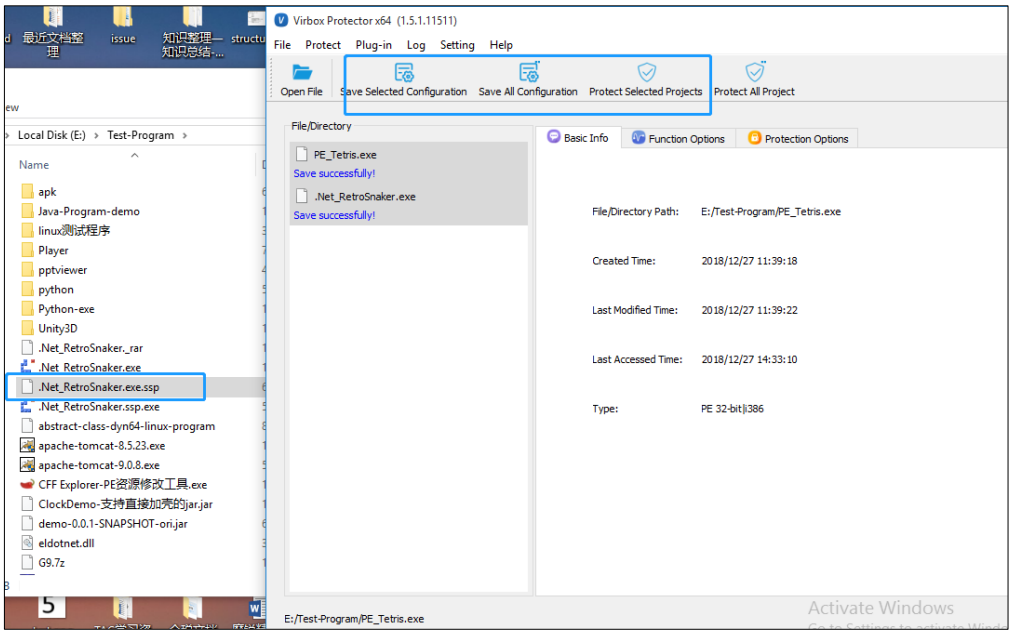


Figure 5-20

5.1.3.4 Protect software with command line

Virbox Protector support developer to protect their applications with command line mode:

virboxprotector_con.exe

Command line parameter :

Command	Description (Local Dongle)	Description(Cloud License)	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 No. of functions and protection mode in the GUI tool
- If no configuration file is generated, only the default entry functions will be protected

Open a terminal window in Linux system, 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 5-21

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

Command: absolute path of VirboxProtector_con absolute path protected file name -o '/absolute path of output file/ouput file name

- 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 '/h
ome/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 5-22

- 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 '/h
ome/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 5-23

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 to protect Unity3D*: Path of VirboxProtector_con/VirboxProtector_con /path of the program to be protected/the filename -u3d –o Path of output file/new file name*

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 '/h
ome/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particle
s2018.1.9f1'
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 5-24

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 '/h
ome/sense/Desktop/Particles2018.1.9f1' -u3d -o '/home/sense/Desktop/ssp.Particle
s2018.1.9f1'
SenseShield Virbox [version: 1.5.0.10808]
Copyright(c) SenseShield Technology Co., Ltd.  All rights reserved.


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

Figure 5-25

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.

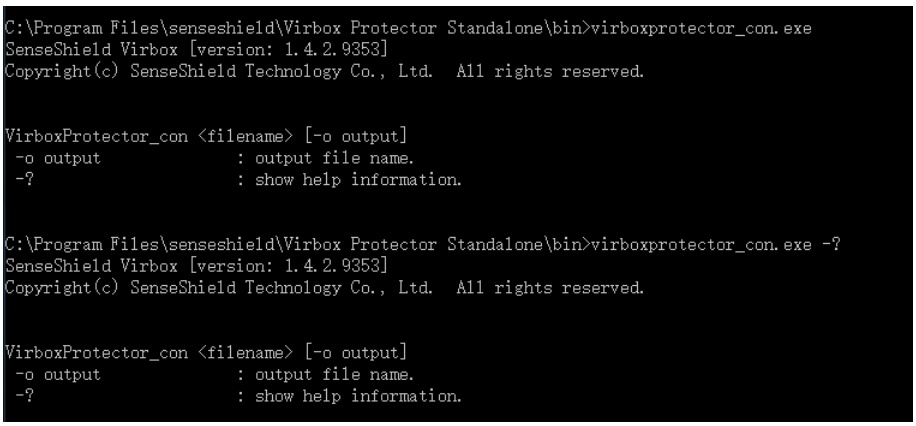


Figure 5-26

When you use the command line but 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/protect filename*

-o path of output file/file name

Figure 5-27

After the license is verified, you can complete protection.

Figure 5-28

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/ Virboxprotector_con the path of the program that to be protected/file name –u3d -o path of output file/filename*

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

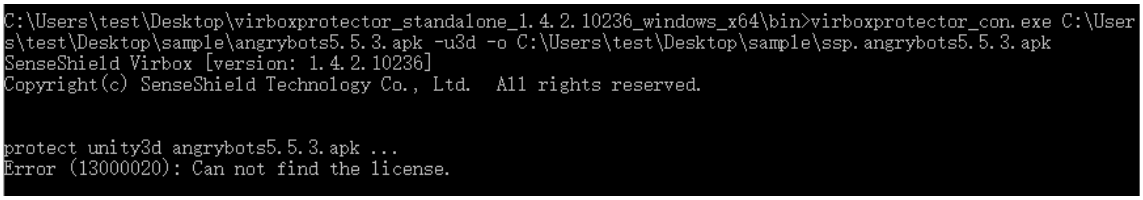


Figure 5-29

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 5-30

5.2 Protect the .Net application

Drag the .NET file you want to protect into Virbox Protector first, then Virbox Protector will start parsing..

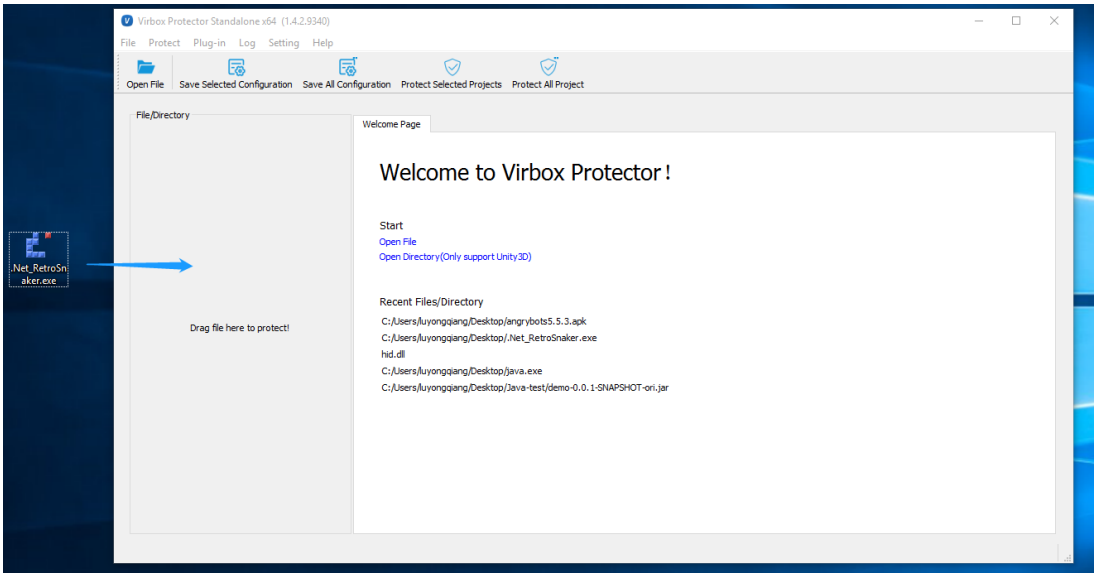


Figure 5-31

5.2.1 Protect the .NET application in fundamental

Protection Option pane to .NET application as shown below:

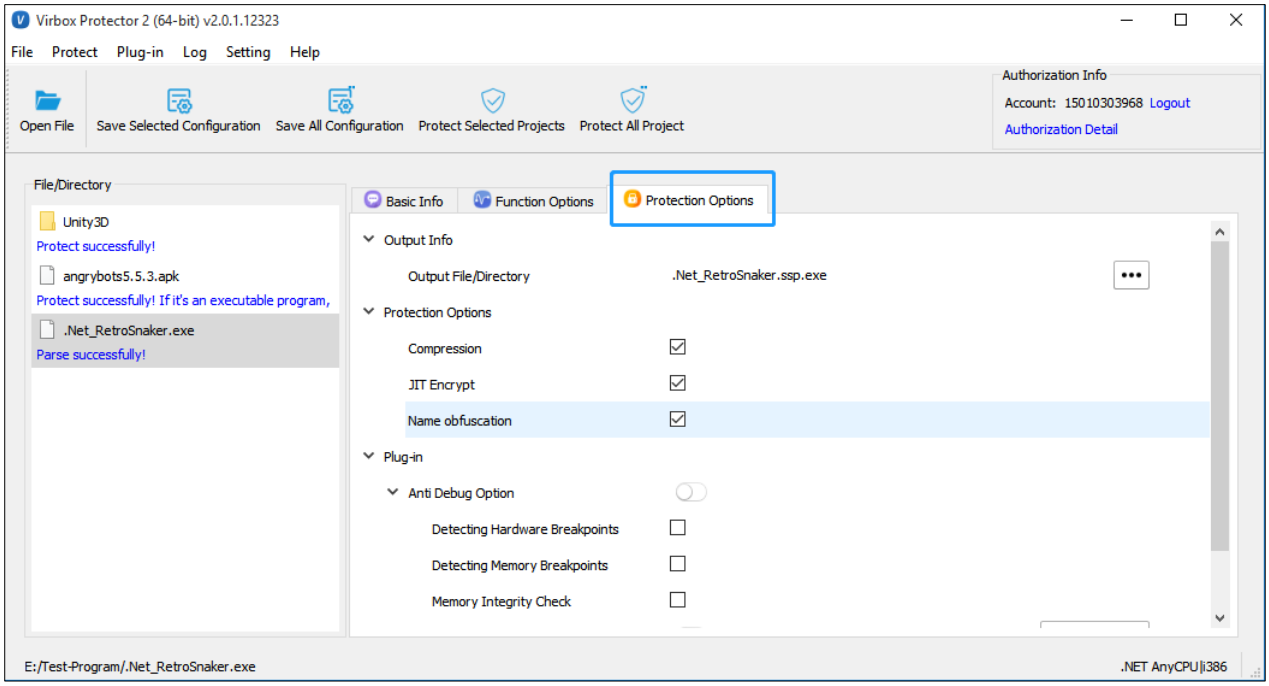


Figure 5-32

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 will not be changed.

Protection Comparison:

Without protection:

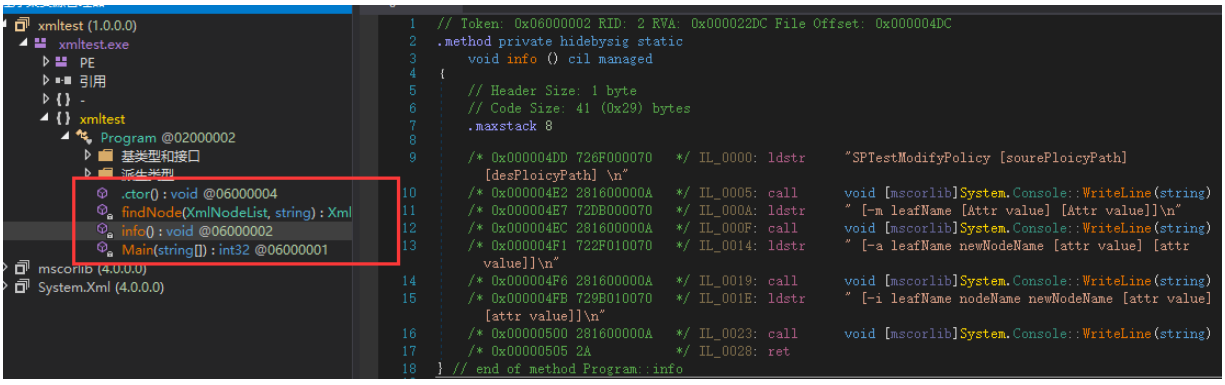


Figure 5-33

With Protection:

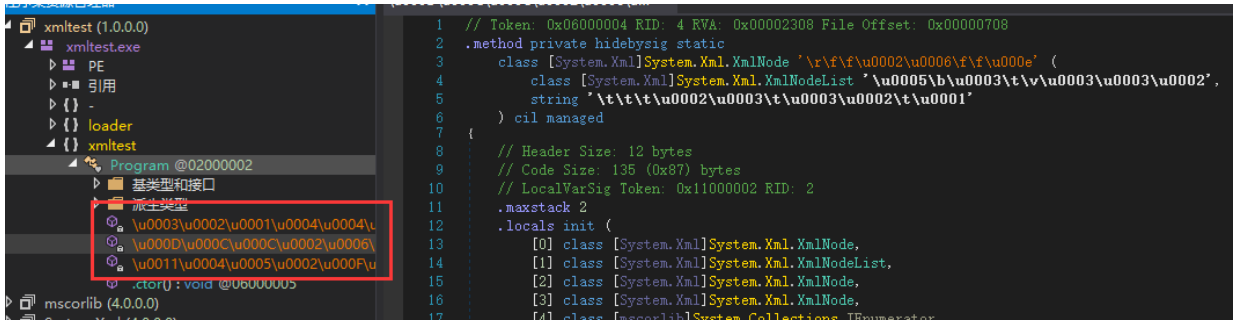


Figure 5-34

5.2.1.2 Compression

The main purpose of the compression is not only to compress the .NET applications, 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 de compiled by DnSpy, ILSpy and .Net Reflector. The compression of Virbox Protector provides 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 de-compiling 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:

Without Protection:

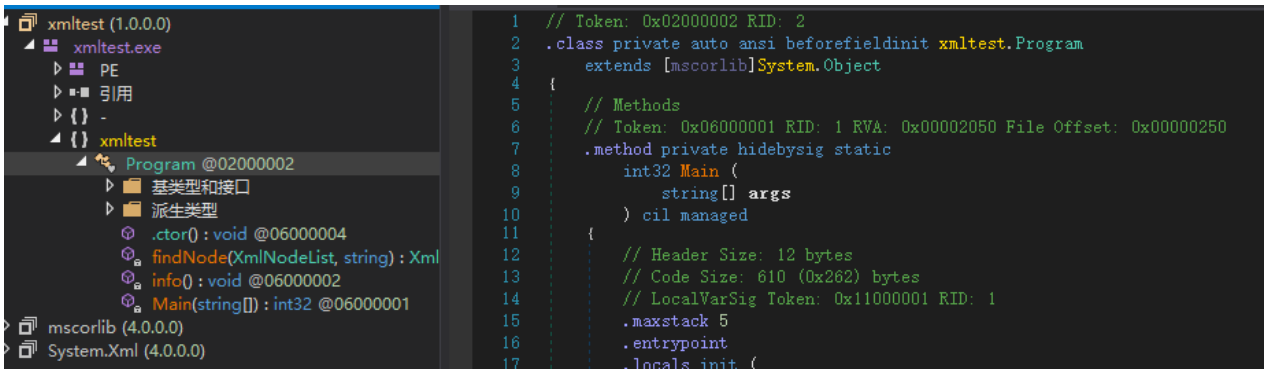


Figure 5-35

With protection:

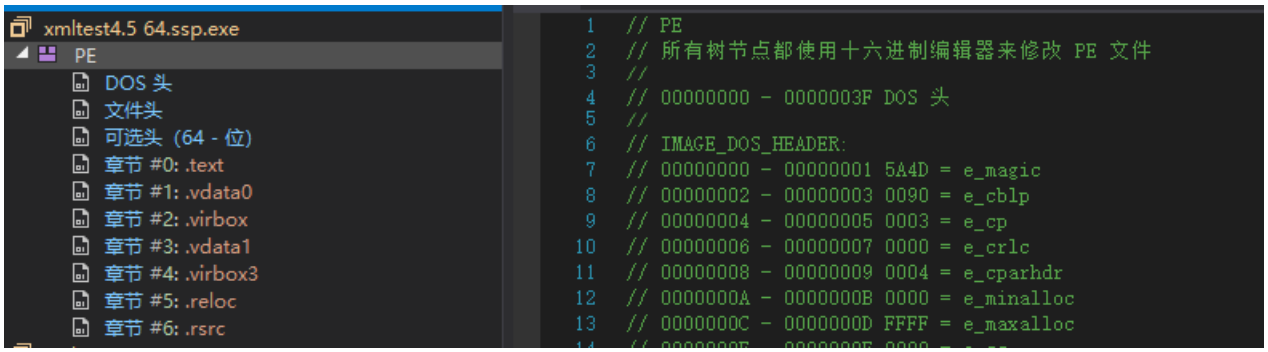


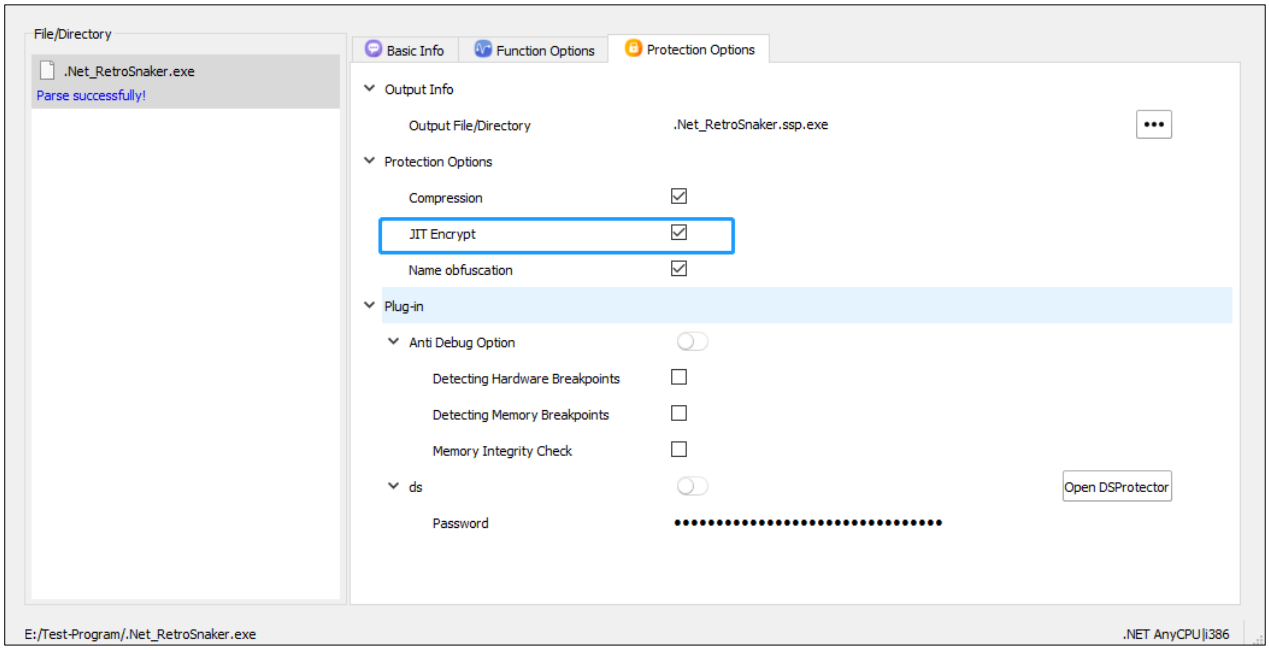
Figure 5-36

5.2.1.3 JIT Encryption

- .Net JIT encryption means it encrypt all of the IL instructions of method in the .Net Program, and the instructions will be decrypted only when the JIT compiling proceed in the .Net Virtual Machine, This can be used to prevent static decompiling and prevent the IL code being Dumped in memory.
- JIT encryption encrypt all of the method in default and enhance the security level of the source code With Protection.
- JIT encryption support inheritance, event, reflection, recursive call which is not supported by code encryption.

Using 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 assembly with a mechanism which consists of an Assembly's identity, that means the Assemblies can be assigned a cryptographic signature. Strong Name contains the name of the .net assembly, version number, culture identity, and a public key token.
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 Without Protection/encryption and add the Strong name With Protection/encryption again.

5.2.2 Protect the critical Functions in .NET program

Virbox Protector support developer to select and encrypt specified functions, critical or contains critical algorithms, and select the encryption mode to each specified functions, let's go to:

Function Option **Tab**:

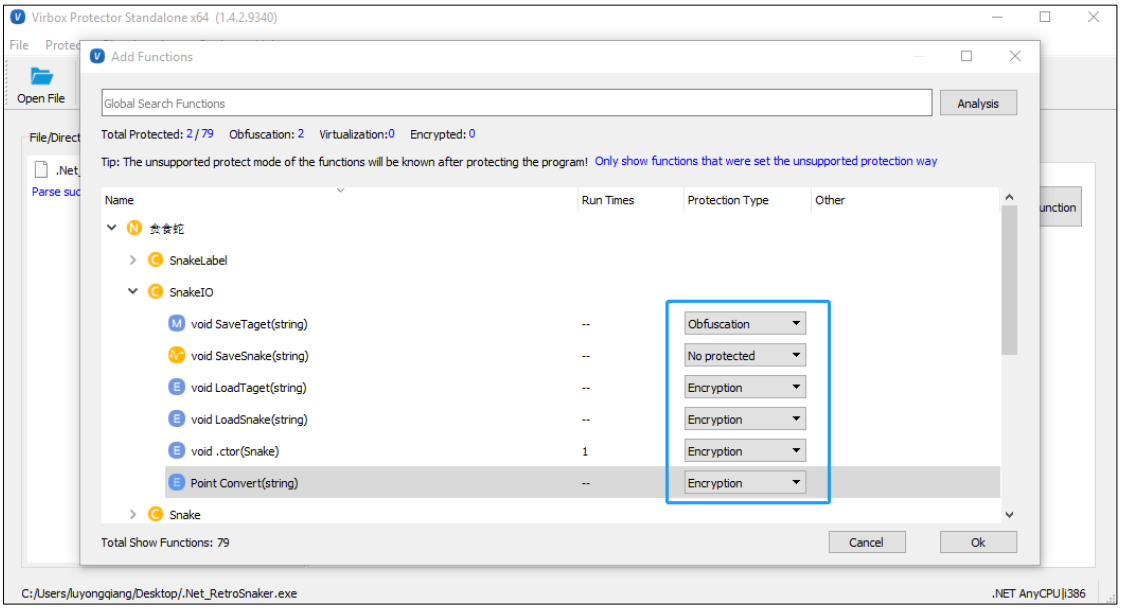


Figure 5-37

Function Level Protection:

Developer may select following "Protection Option" to protect specified "functions":

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:

Without protection:

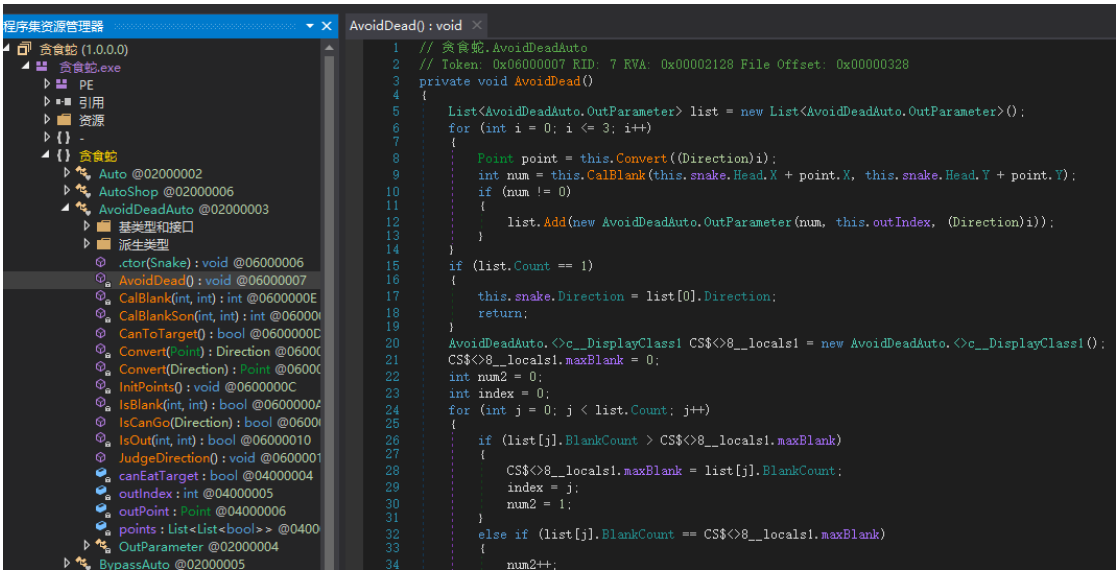


Figure 5-38

With protection:

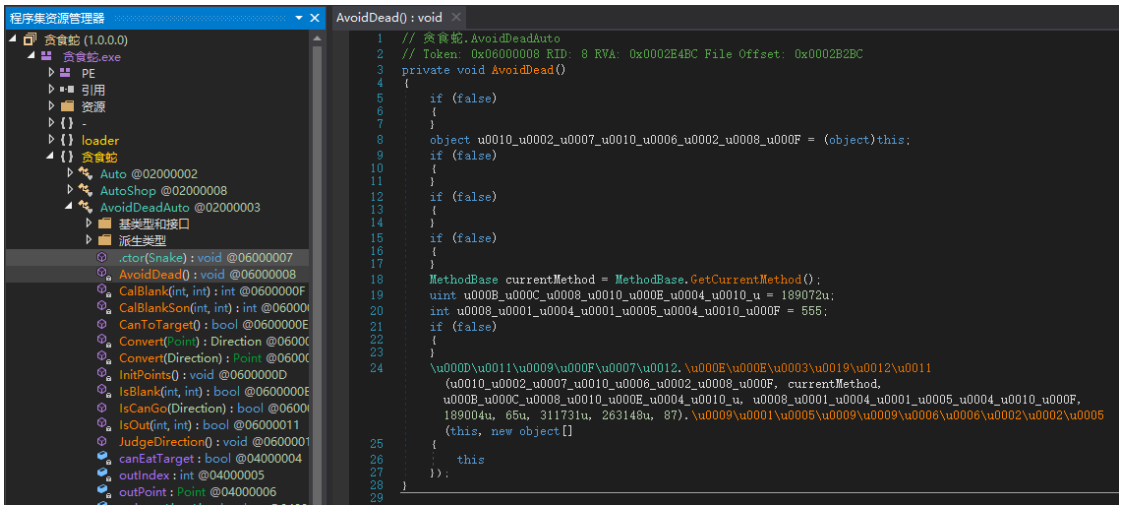


Figure 5-39

Following scenario doesn't supported to code encryption:

For C# program, if following the message pop up: “Some of the functions has been set the protection option unsupported, and please modify the protection option you selected. Error code: 0xA000A000” , pls go back to "Function Protection" tag to modify the protection mode:

Following scenario doesn't supported to "code encryption":

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

Protection Mechanism:

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

The Benefit:

Prevent from de compiling and make the hacker more difficult to analysis the code.

The Weakness:

Slightly Negative impact to software performance.

Limited protection to software.

Protection Comparison:

Without protection:

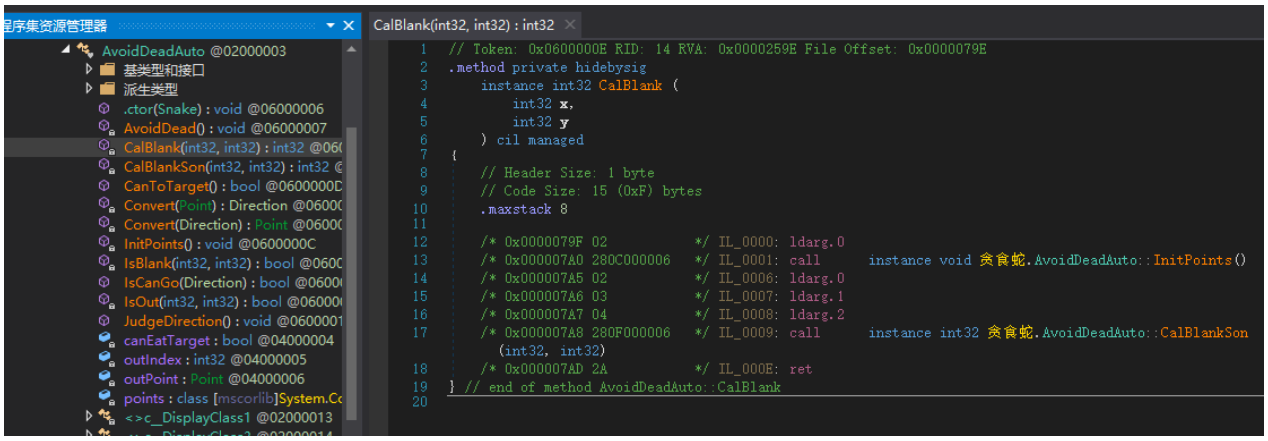


Figure 5-40

With Protection:

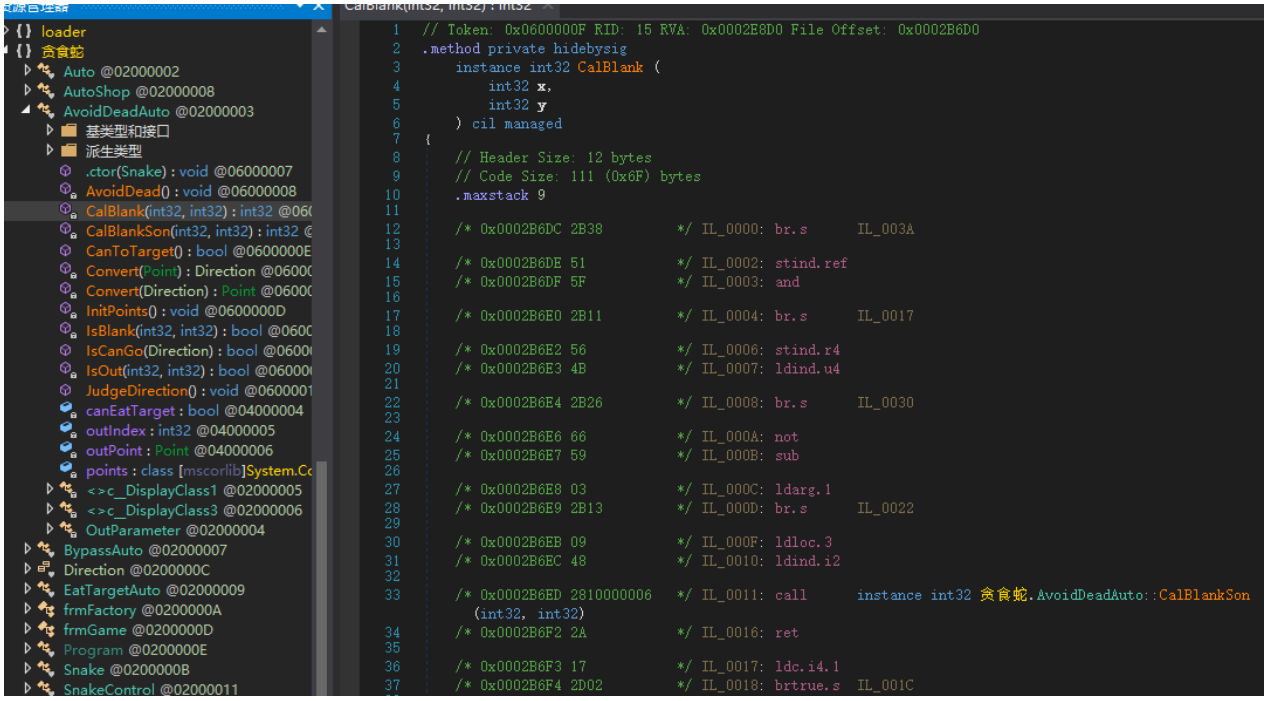


Figure 5-41

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

Then click "**Protect all project**" button **to complete** protection:

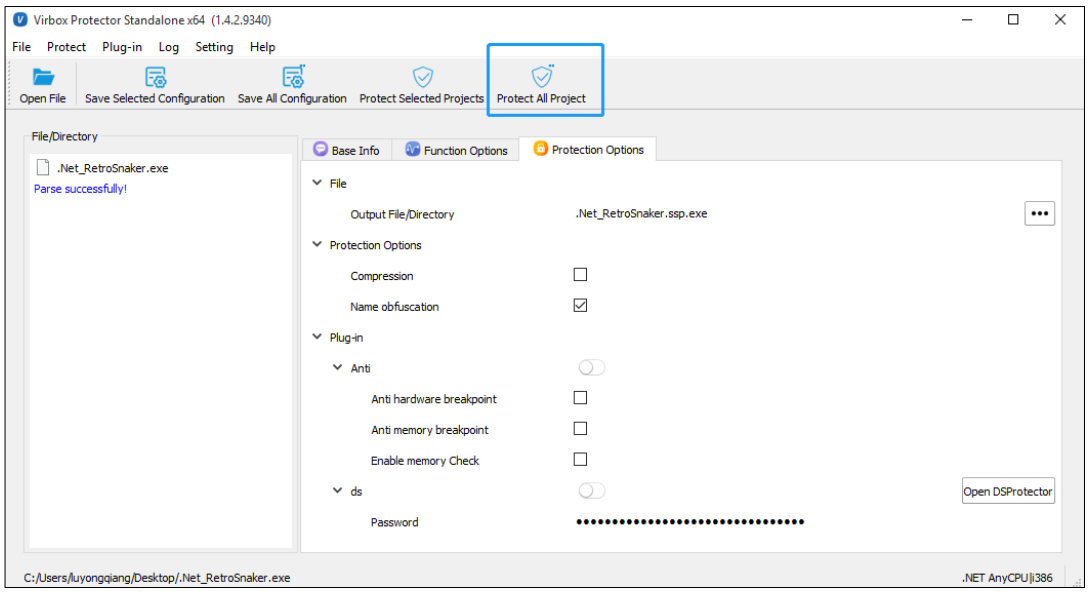


Figure 5-42

5.2.2.3 Using label to mark the critical function be protected

Virbox Protector support to protect the critical functions with 2 protection modes:

Code Encryption and Code Obfuscation

Developer may reserve and set the protection mode to the function will be protected in coding, and it can be quoted and viewed in the code, so, when the compiling completed, developer drag the apps into the Virbox Protector, the GUI will show the protection mode set in the code accordingly, here is label sample for code:

```
//Label
namespace Virbox{
    //Code Obfuscation
    class Mutate : System.Attribute
    {
    }
    //Code Encryption
    class Encrypt: System.Attribute
    {
    }
}
public class main
{
    [Virbox.Mutate]//Code Obfuscation
    public static void test1(string[] args)
    {
        System.Console.WriteLine("hello Virbox.Mutate!");
    }
    [Virbox.Encrypt]//Code Encryption
    public static void test2(string[] args)
    {
        System.Console.WriteLine("hello Virbox.Encrypt!");
    }
    public static void Main(string[] args)
    {
        test1(args);
        test2(args);
    }
}
```

5.3 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 class file 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 de-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 de compiling engineering.

Virbox support directly protect the Jar archive, War archive and APK. ***Virbox Protector*** encrypt the byte code of every method to prevent the source code from being de-compiled, it is quite easy for developer to use Virbox Protector to protect Jar, War projects in Windows, Linux, ARM Linux platform are supported to be protected.

The Protection Mode: Virbox protector support to protect Java project with 2 kinds of Mode, Java VME and Java BCE (with different license), developer may select one of protection mode to protect the Jar/War Project.

Comparison to Java VME and Java BCE:

- With Java VME protection, Developer protect and encrypt the Java's method with Virtualization, the most secured protection mode with highly security;
  With Java BCE protection, Developer protect and encrypt the bytecode of each method of Java class file, the bytecode will be only decrypted in execution;
- The execution for Java VME and BCE are different;
  With VME protection, the execution to the protected Jar/War is same;
  with BCE protection, the execution to protected Jar/war file need relevant sjt_agenet.jar to execute;
- Encryption Process is different:
  VME supports to protect/encrypt the Jar/War archive directly;
  with BCE mode, developer need to save the Jar/war file into a folder and drag/add the folder into the Virbox Protector;
- With VME Protection, the Protected Jar/War projects support the scenario which Java Jar file be called by other program; BCE mode doesn't support similar scenario;

5.3.2 Protect Java with Virtualization (Java VME)

Java VME: Protect the Java code with Virtualization, JAVA VME is highly secured protection function provides by Virbox Protector (Developer need to purchase this function separately), Java VME transform the JVM bytecode into the private VM instructions, when Java code executed, the java program will go to executed in

Native VM environment, and provides most secured environment to executed and effective to defend the retrieving and cracking by current decompiling tools available in the market.

Note:

1) Not support to protect the embedding Java code currently;
2) For the Jar archive which use the springframework, it doesn't support to protect the "class" which under the *org/springframework/boot/loader*, only support to protect the class file of the core class file (for example, the class file under the *BOOT-INF/classes*.
3) Following type of functions doesn't support to be protected with VME mode (Virtualization): Constructors, destructors and reflection;
4) There are 2 different license for Virbox Protector (Java VME & Java BCE) to protect Java program. you need to contact and get commercial license from Virbox Sales Team.

5.3.2.1 JAVA VME Protection process:

1. Drag the Jar archive directly into the Virbox Protector:
2. Click "Add Functions" and Select the functions and set the "Virtualization" to the functions which you want to protect from the ***"Function Option"*** tab;

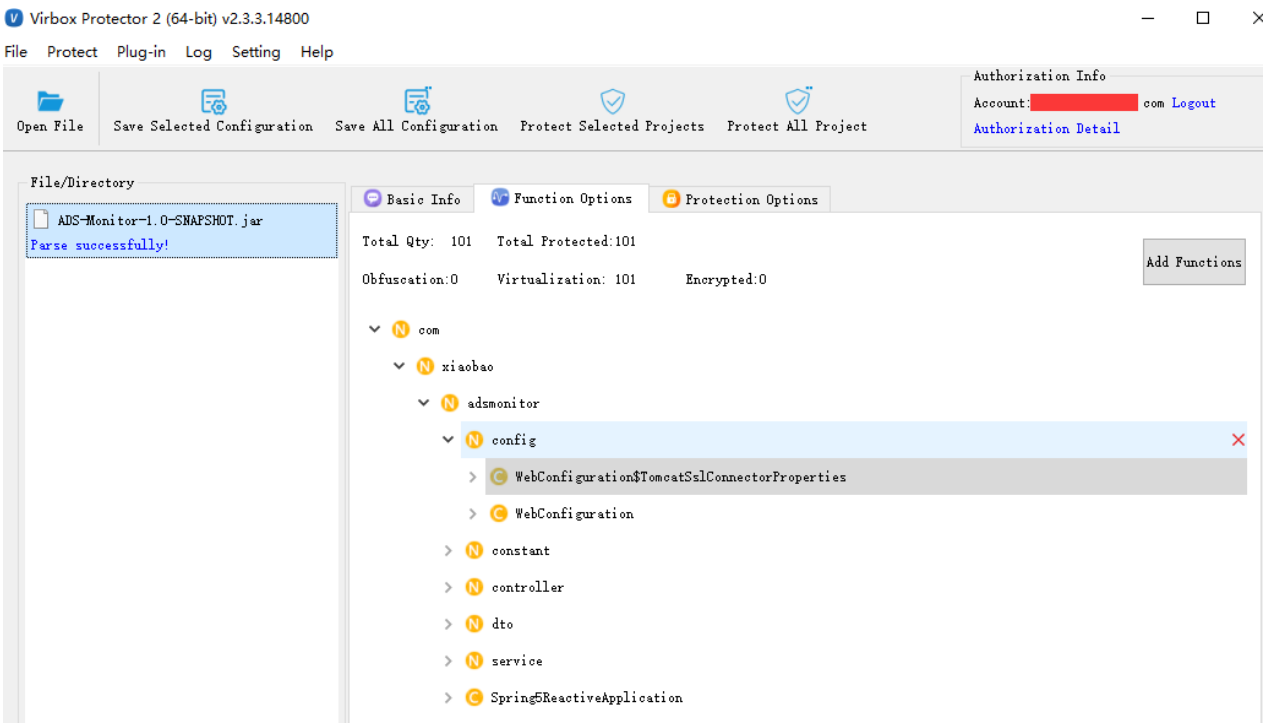


Figure 5-43

3. Click "Protect Selected Project" to complete the protection process.
   a new, protected Jar archive and relevant .ssp file will be generated.

- Note: the abc.jar is original jar project; the abc.ssp.jar is the protected jar archive; and the abc.jar.ssp is the configuration file respectively;

5.3.2.2 VME Protection Result:

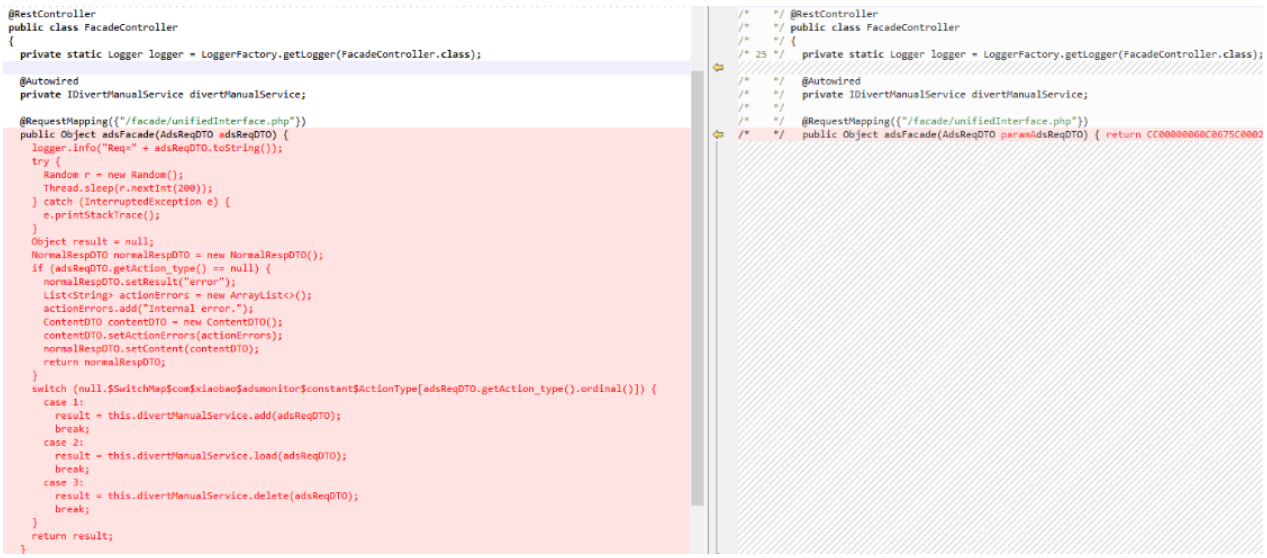


Figure 5-44

5.3.2.3 Use Command line to protect java code with VME protection mode

5.3.2.3.1 Use Virbox Protector GUI to generate the configuration file

5.3.2.3.2 Open a terminal windows, goes to the sub directory which the " *virboxprotector_con*" located, and input *virboxprotector_con*, to view the help information

5.3.2.3.3 Use following command:

virboxprotector_con <The jar which need to be protected> -o <the jar which output>

5.3.2.4 Using Label to mark the critical function with Code Virtualization protection

Virbox Protector supports to protect the critical functions with "Code of Virtualization" protection mode, developer may label the critical function in coding, and quoted in functions, so, when the source code compilation completed, drag the program into the Virbox Protector GUI tools, then Virbox Protector will show protection mode to the functions in coding. here is sample:

5.3.2.4.1 Create VBVirtualize.java

```
package virbox;
public @interface VBVirtualize
{
}
```

5.3.2.4.2 How to call

```
import virbox.VBVirtualize;
@VBVirtualize //can be add to the class, then all of method can be protection on default
public class Main {
	public static void main(String[] args) {
		System.out.println("hello");
		test_vir();
	}
	@VBVirtualize //can be to the method, protect this method only
	public static void test_vir()
	{
		System.out.println("test_vir");
	}
}
```

5.3.3 Protect to Jar archive (BCE)

Besides of Java Protection with VME mode, Virbox Protector support developer to protect Jar Archive with BCE mode in simple way: drag the folder which the jar project located to the Virbox Protector and click to select the protection option to complete the protection process. after protection, a new and protected Jar archive and sjt_agent.jar will be generated.

Before start protection, pls save your Jar file into a folder first, then,

1. Drag the Jar archive directory into the Virbox Protector and assign the output directory in "Protection Option"

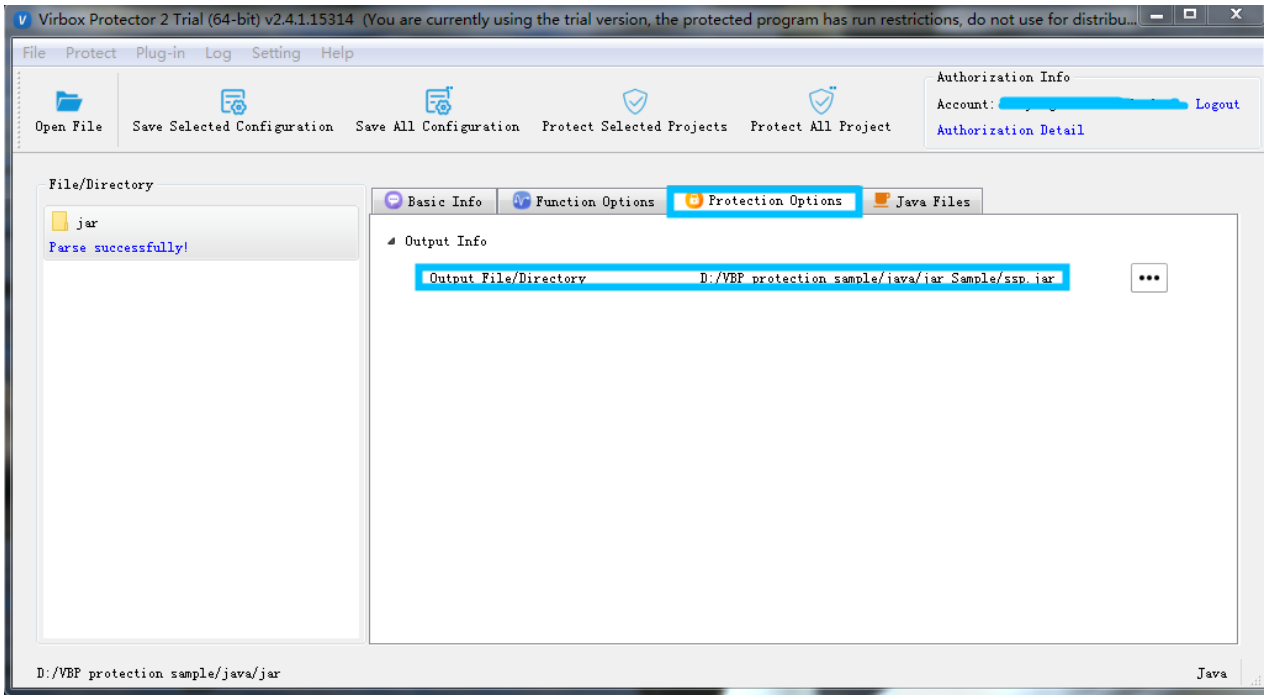
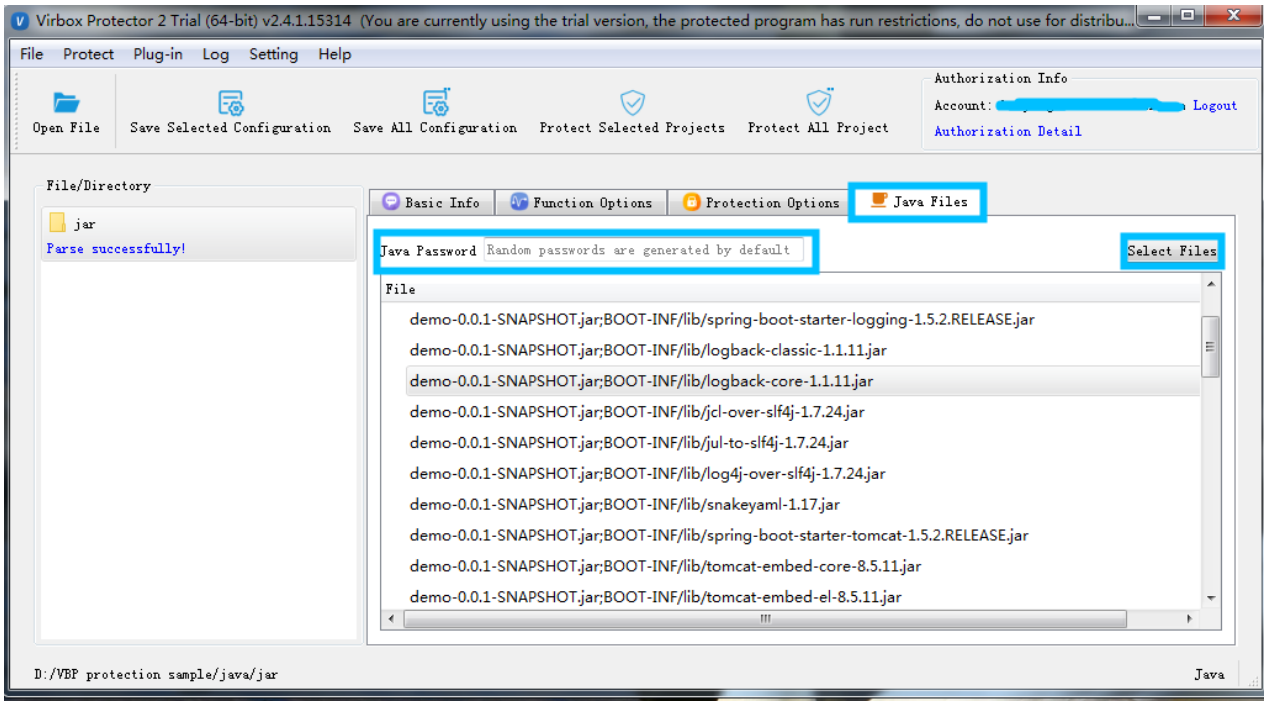


Figure 5-45

2. Select Jar file:

Go to "Java File" tab, and select the java file to be protected, and input password (pls remember and keep this password), if the password not input, the system will generate a random password for you. this password will be required when you updated the jar archive, and no need to update the sjt_agent.jar.



3. Click "Protect Selected Project" to complete the protection, then, a new directory will be generated

Figure 5-46

then go to the jar _sample folder, which contained encrypted Jar archive and sjt plugin, as shown the figure below:

Figure 5-47

5.3.3.1 Deployment

Windows:

Directly run the protected Jar archive

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

   Command：

   java -javaagent:sjt_agent.jar-jar ***.jar

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

   Command:

   java -javaagent:C:\Users\test\Desktop\sjt\sjt_agent.jar -jar ***.jar

Linux System:

Directly run the protected program:

- If the sjt library are in the same directory with Jar archive, you can run the following command in the current directory:
  - Command:

*java -javaagent:sjt_agent.jar -jar ***.jar*

- If the sjt library is not in the same directory with the jar archive, you need to assign the absolute directory:
- Command:

  *java -javaagent:/home/sense/Desktop/sjt_so/sjt_agent.jar -jar ***.jar*

macOS System:

Directly run the protected program:

- If the sjt library are in the same directory with Jar archive, you can run the following command in the current directory;
- Command:

  *java -javaagent:sjt_agent.jar -jar ***.jar*
- If the sjt library is not in the same directory with the jar archive, you need to assign the absolute directory:
- Command:

  *java -javaagent:/Users/sense/sjt/sjt_agent.jar -jar ***.jar*

Jar archive Protection performance comparison:

Without Protection:

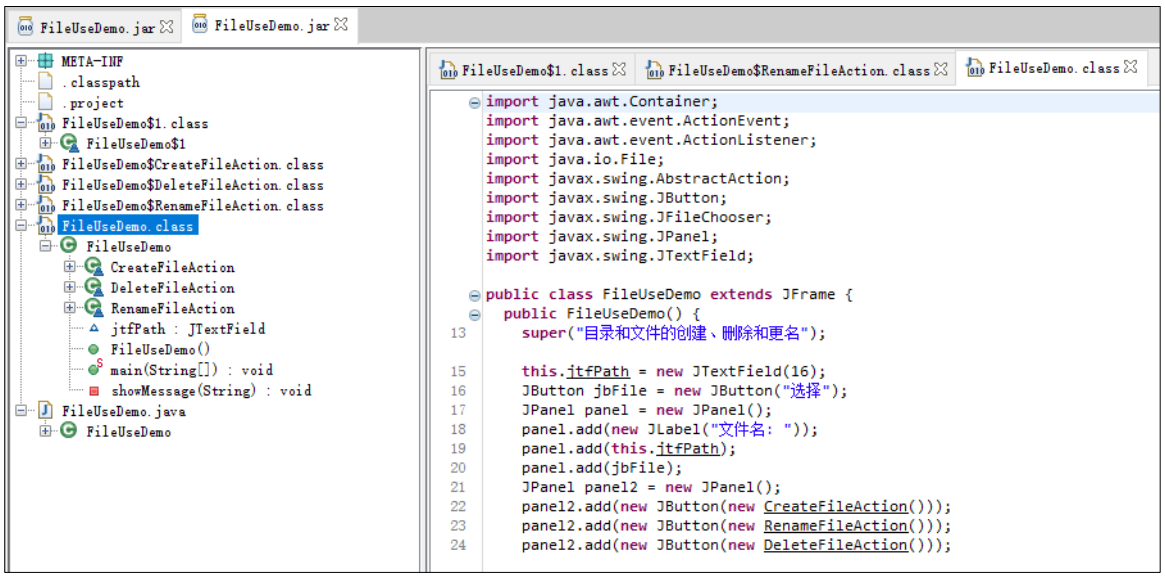


Figure 5-48

With Protection:

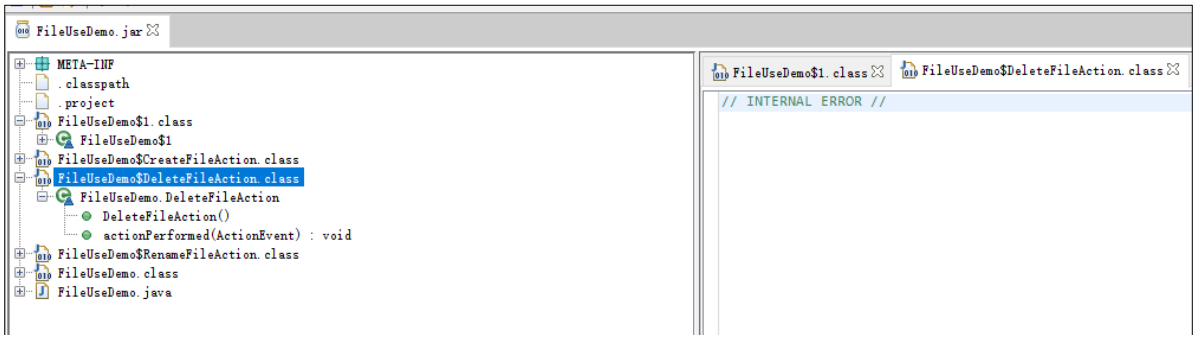


Figure 5-49

5.3.4 War archive protection:

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

1. Drag War archive into Virbox Protector to protect War Archive;

   (The war protection process is same as the process of Jar archive protection, see above Jar protection process.)

2. After successfully encrypted, ***sjt*** plugin and the protected War archive will be generated.

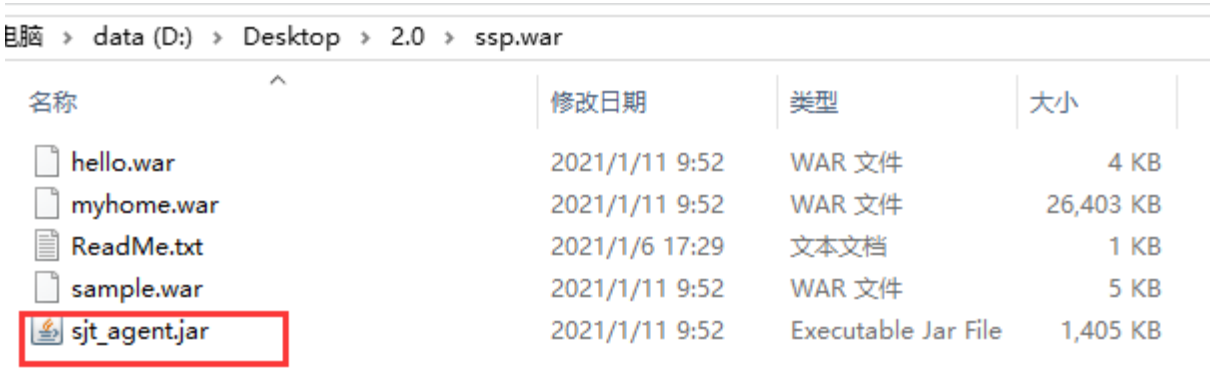


Figure 5-50

Deployment：

5.3.4.1 Windows system:

Use following ways to configure the system environment, you can select one of them:

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

Create the *setenv.bat* in the *tomcat\bin* directory, for example:

a) Create "*setenv.bat*" in the *tomcat\bin* directory, set the environment variable such as (absolutely

path):

set CATALINA_OPTS=%CATALINA_OPTS% -javaagent:sjt_agent.jar

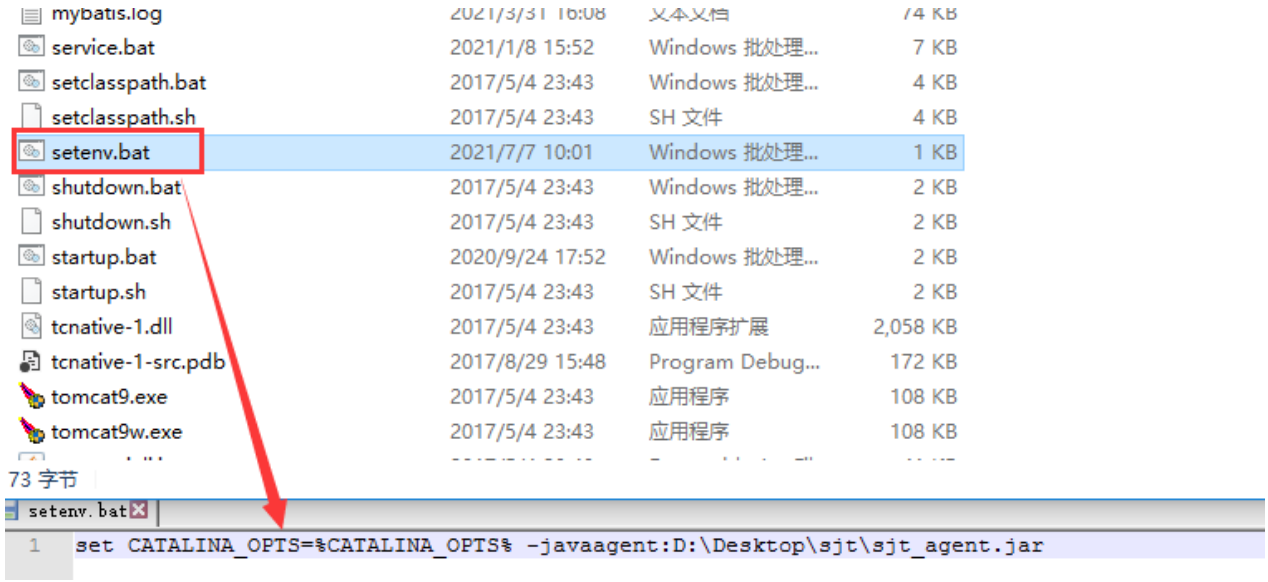


Figure 5-51

b) Put the encrypted "war archive" into the location: *.\apache-tomcat\webapps* and start the tomcat service.

2. Start the tomcat when system service started

a) First you need to uninstall the tomcat service, use the console command: *service.bat uninstall* to uninsall tomcat service;

Figure 5-52

b) Add the sjt lib in the parameter of *JvmOptions* in the *service.bat*, as shown as snapshot below:

```
"%EXECUTABLE%" //IS//%SERVICE_NAME% ^
    --Description "Apache Tomcat 9.0.0.M21 Server - http://tomcat.apache.org/" ^
    --DisplayName "%DISPLAYNAME%" ^
    --Install "%EXECUTABLE%" ^
    --LogPath "%CATALINA_BASE%\logs" ^
    --StdOutput auto ^
    --StdError auto ^
    --Classpath "%CLASSPATH%" ^
    --Jvm "%JVM%" ^
    --StartMode jvm ^
    --StopMode jvm ^
    --StartPath "%CATALINA_HOME%" ^
    --StopPath "%CATALINA_HOME%" ^
    --StartClass org.apache.catalina.startup.Bootstrap ^
    --StopClass org.apache.catalina.startup.Bootstrap ^
    --StartParams start ^
    --StopParams stop ^
    --JvmOptions "-Dcatalina.home=%CATALINA_HOME%;-Dcatalina.base=%CATALINA_BASE%;-Djava.io.tmpdir=%CATALINA_BASE%
    \temp;-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-Djava.util.logging.config.file=%CATALINA_BASE%
    \conf\logging.properties;-javaagent:D:\Desktop\sjt\sjt_agent.jar;%JvmArgs%" ^
    --Startup "%SERVICE_STARTUP_MODE%" ^
    --JvmMs "%JvmMs%" ^
    --JvmMx "%JvmMx%"
if not errorlevel 1 goto installed
echo Failed installing '%SERVICE_NAME%' service
goto end
:installed
echo The service '%SERVICE_NAME%' has been installed.

:end
cd "%CURRENT_DIR%"
```

Figure 5-53

c) Then use the command: *service.bat.install* in the console windows to install;

```
D:\apache-tomcat-9.0.0.M21\bin>service.bat install
Installing the service 'Tomcat9' ...
Using CATALINA_HOME:    "D:\apache-tomcat-9.0.0.M21"
Using CATALINA_BASE:    "D:\apache-tomcat-9.0.0.M21"
Using JAVA_HOME:        "C:\Program Files\Java\jdk1.8.0_66"
Using JRE_HOME:         "C:\Program Files\Java\jdk1.8.0_66\jre"
Using JVM:              "C:\Program Files\Java\jdk1.8.0_66\jre\bin\server\jvm.dll"
The service 'Tomcat9' has been installed.
```

Figure 5-54

d) then start "tomcat" service;

e) put the protected "war archive" into the folder of ".*apache-tomcat**webapps*, then start tomcat service.

3. Start service when you using tomcat9.exe

a) First step is to start the tomcat9w.exe

b) add the *sjt lib* in the Java Options list, as shown in the snapshot below:

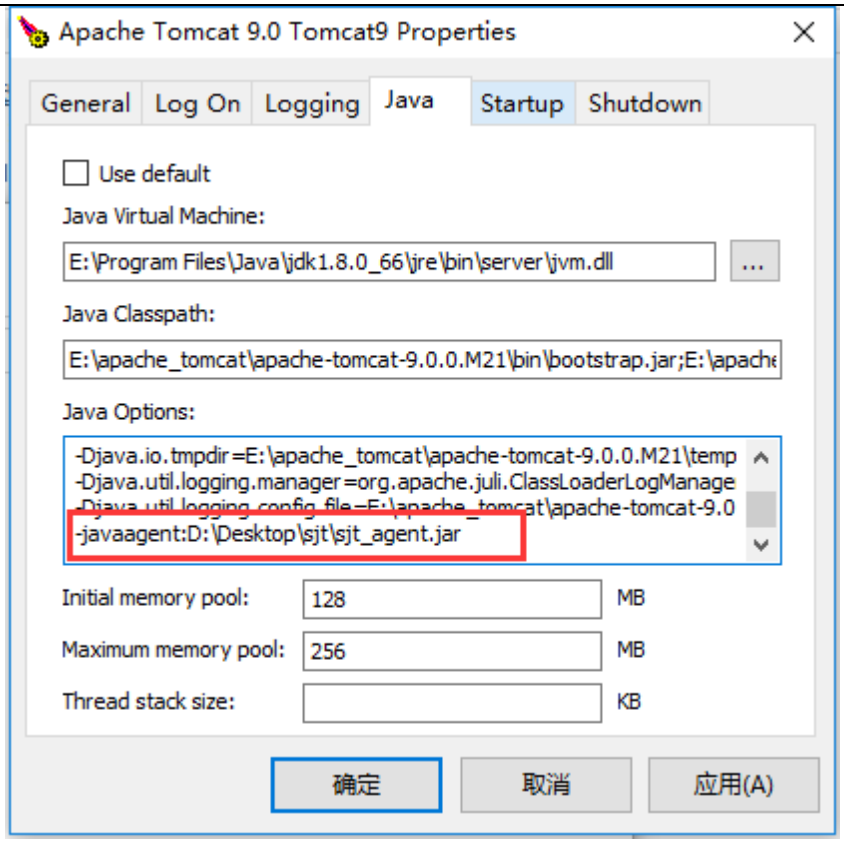


Figure 5-55

c) Execute tomcat9.exe to start tomcat service and put the protected war archive into the folder .*apache-tomcawebapps*, then start the tomcat service.

5.3.4.2 Linux System:

Set Setenv.sh in the tomcat directory*:*

▪ Create a new *setenv.sh* in the *tomcat\bin* directory, the absolute path environment variable can be set as follows: *CATALINA_OPTS="$CATALINA_OPTS -javaagent:sjt_agent.jar*

as shown below:

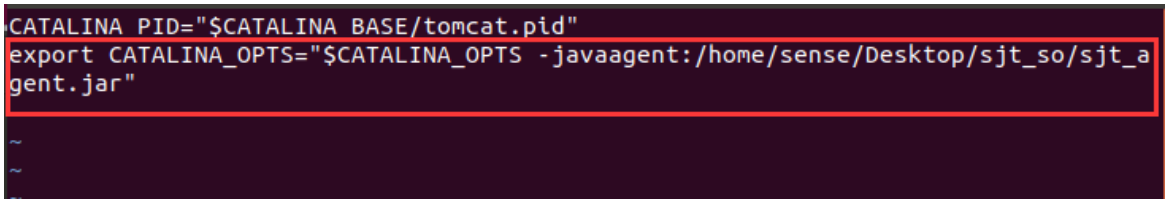


Figure 5-56

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

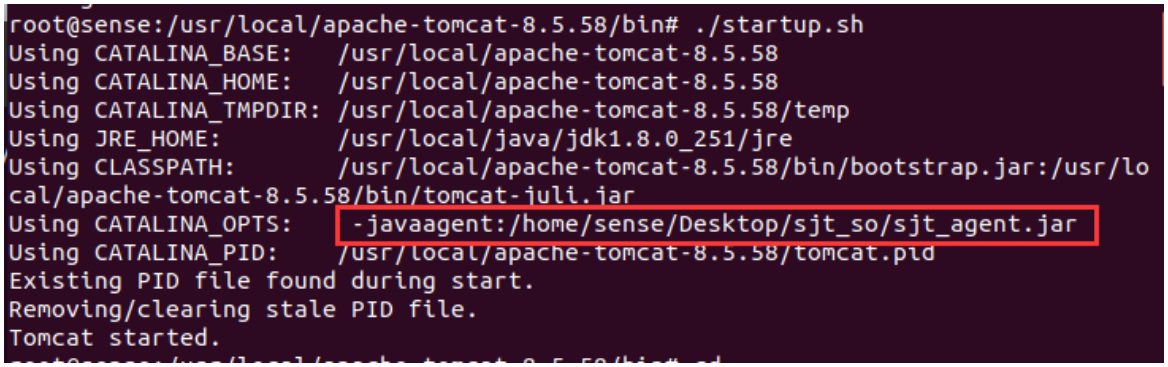


Figure 5-57

- Put the encrypted war archive in the directory: .*apache-tomcat**webapps*

  If the War archive can be parsed correctly, the webpage can run correctly.

Please note: 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.

5.3.4.3 macOS system

- Create a new ***setenv.sh*** in the ***tomcat\bin*** directory, the full path environment variable can be set as follows:
- *CATALINA_OPTS="$CATALINA_OPTS -javaagent:/Users/sense/sjt/sjt_agent.jar*
- Start tomcat service, you can view the parameter of *CATALINA_OPTS* set
- Put the encrypted war archive in the directory: .*apache-tomcat**webapps*

  If the War archive can be parsed correctly, the webpage can run correctly.

5.3.4.4 Protection comparison:

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

Without Protection:

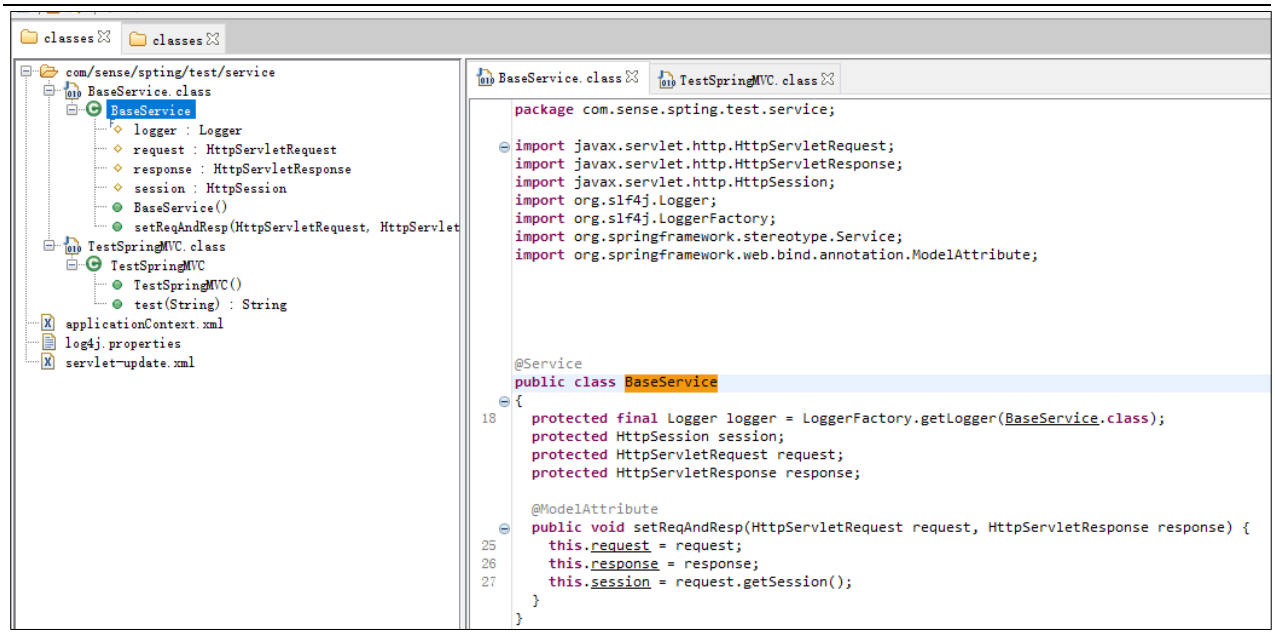


Figure 5-58

With Protection:

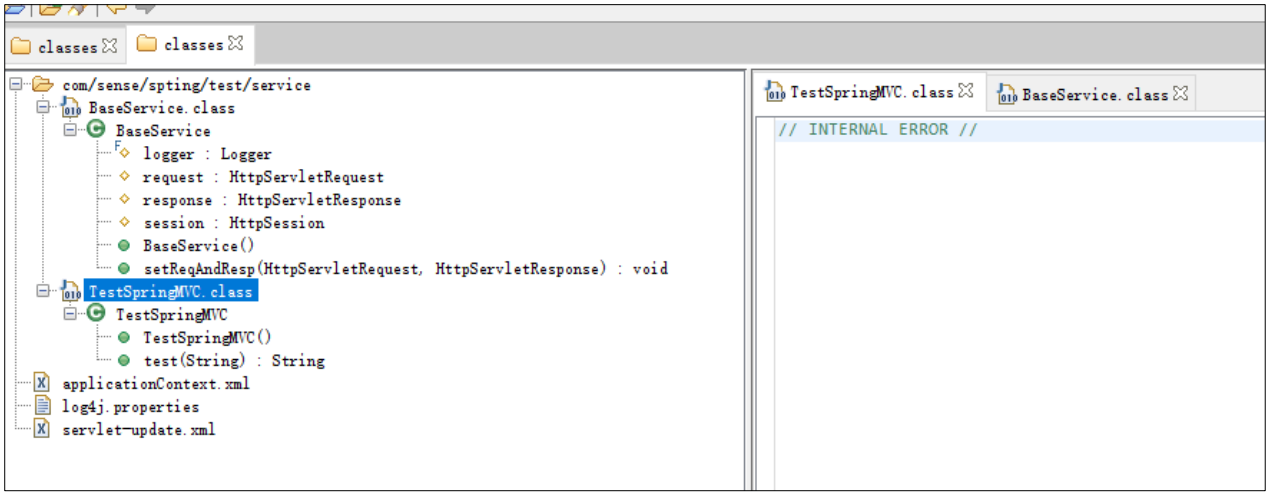


Figure 5-59

5.3.5 Using command line to protect Java

As Java program is much different from the other languages programs, you need to protect/encrypt to all of java directory. here are steps to protect Java program by use of command line mode:

1. User Virbox Protector GUI tool to generate the configuration files;

2. Open the terminal windows, go to the directory which ***virboxprotector_con*** located, input the "***virboxprotector_con***" to execute the Virbox Protector CLI tool, you can view the help information

accordingly;

3. Use the Command to protect your java program:

virboxprotector_con -java <The directory which need to be protected> --password <password> -o <output directory>

Note: Virbox provides different kind of edition of Virbox Protector tool which running in different platform/environment, pls. contact us for detail and get license.

5.4 Unity 3D Program Protection

Virbox Protector supports to protect the Unity3D program in Mono and IL2CPP VM environment.

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 to 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 mechanism 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. Interpret it and execute the program.

If you protect the Unity3D program with traditional software protection tool, it would not protect the main code source. With the Virbox Protector, It 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.

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 de compiling;
2. Add the program set file: the c# program set in the managed directory which is developed by the software developer.

5.4.3 Protect Unity3D in Windows, Linux, macOS Environment

5.4.3.1 Protect with Virbox Protector GUI tools

1. Drag the whole U3D directory to the Virbox Protector

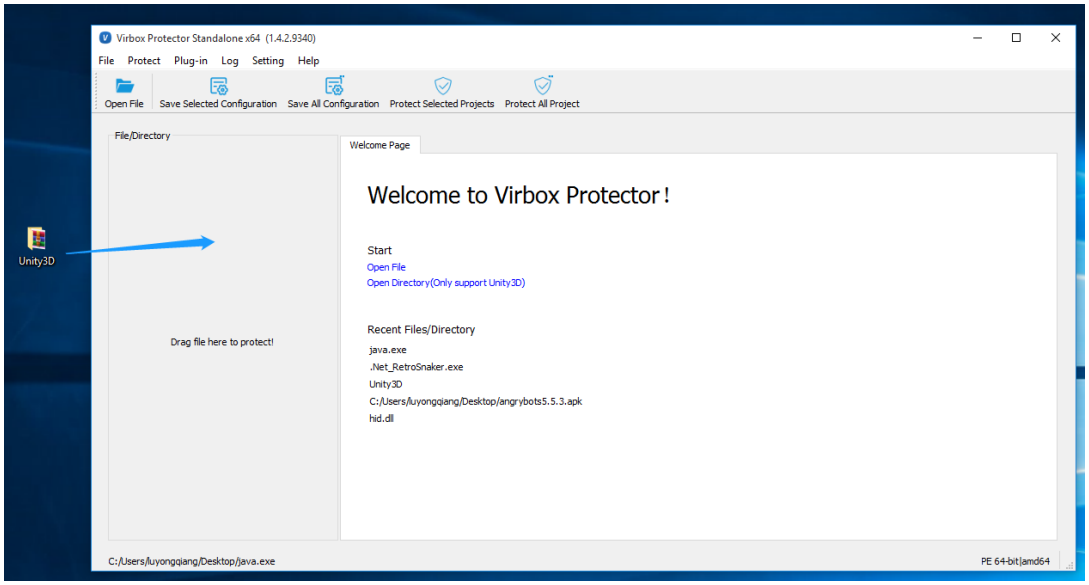


Figure 5-60

2. Virbox Protector support to protect Unity3D with 2 kinds of "compilation framework": ***mono*** type and ***IL2CPP*** type;

For Unity3D with Mono frame, The **Assembly-CSharp-firstpass.dll** and **Assembly-CSharp.dll** will be loaded on default in the “Protection Options” tab, you may also add the customized C# assembly set in the /Managed directory.

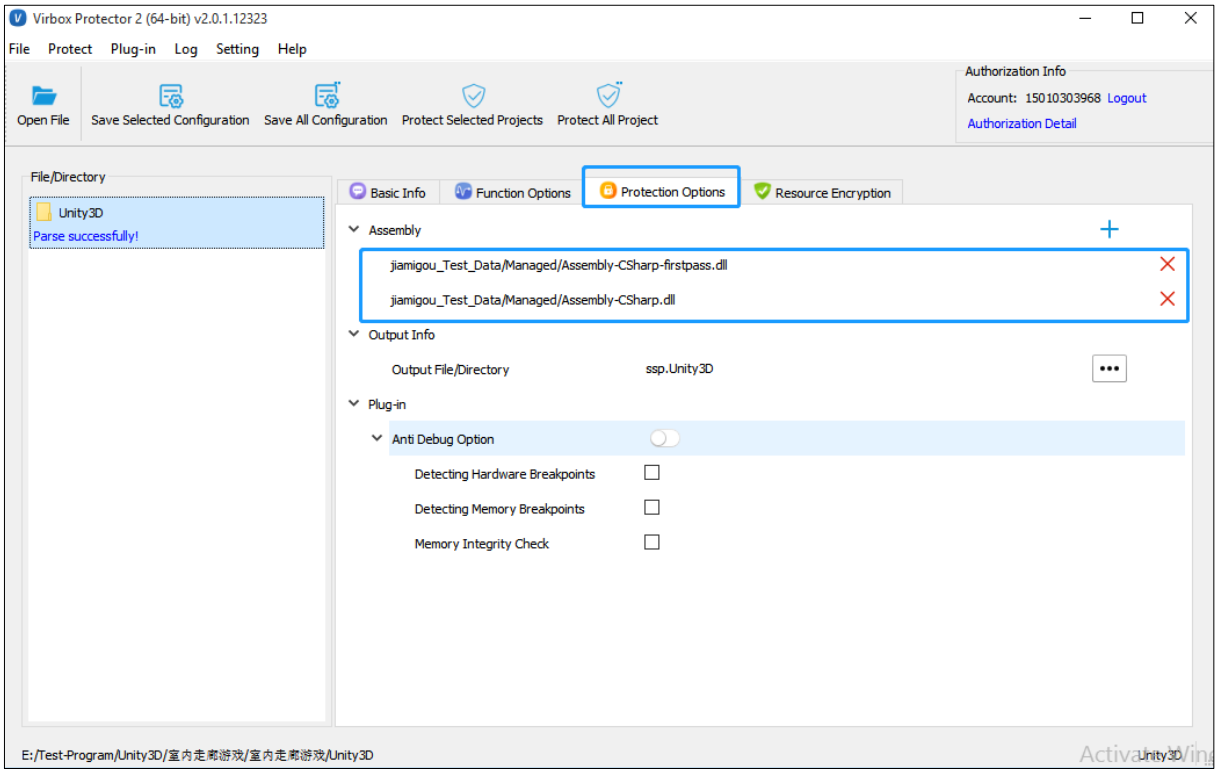


Figure 5-61

For Unity3D with IL2CPP frames, Virbox Protector will protect the **glabl-metadata.dat** on default. see snapshot as shown below:

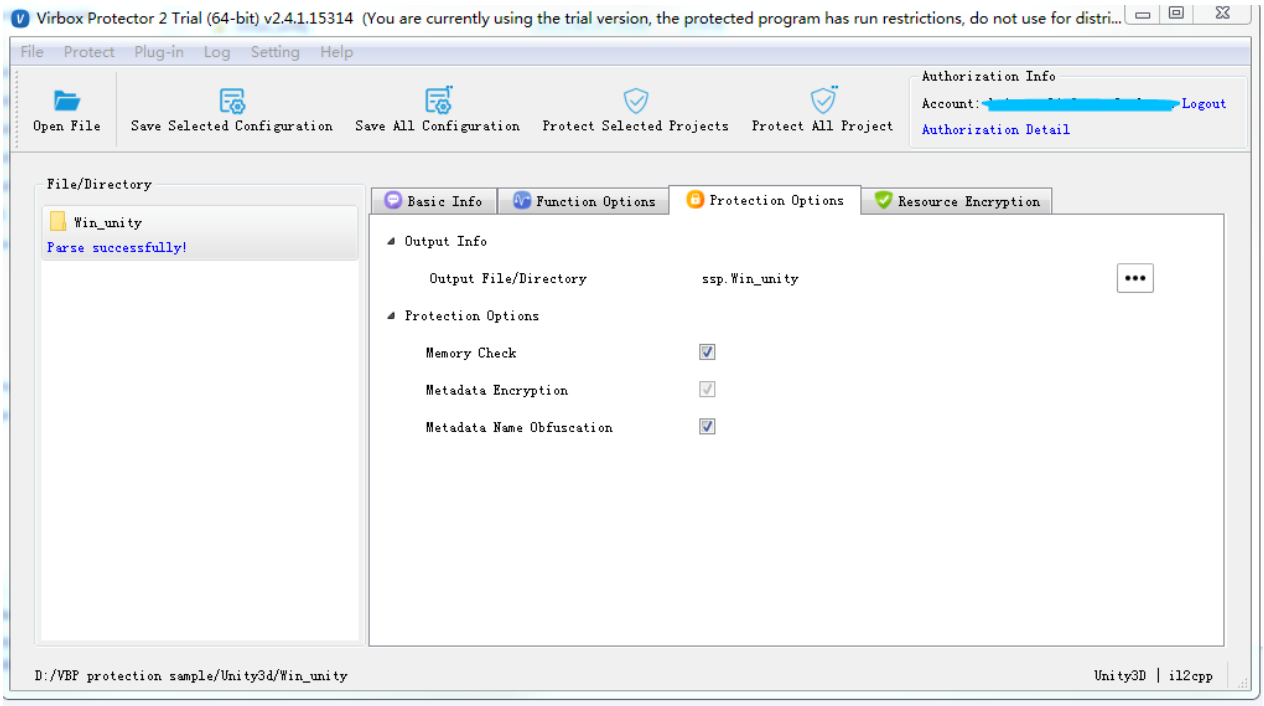


Figure 5-62

3. Protect resource file: Virbox Protector support to protect the Unity3D resource also, You can go to the “Resource Encryption” option (as shown in the picture in below), to protect the data resource, video resource

in your Unity3D program:

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

Pls input the password and keep the password.

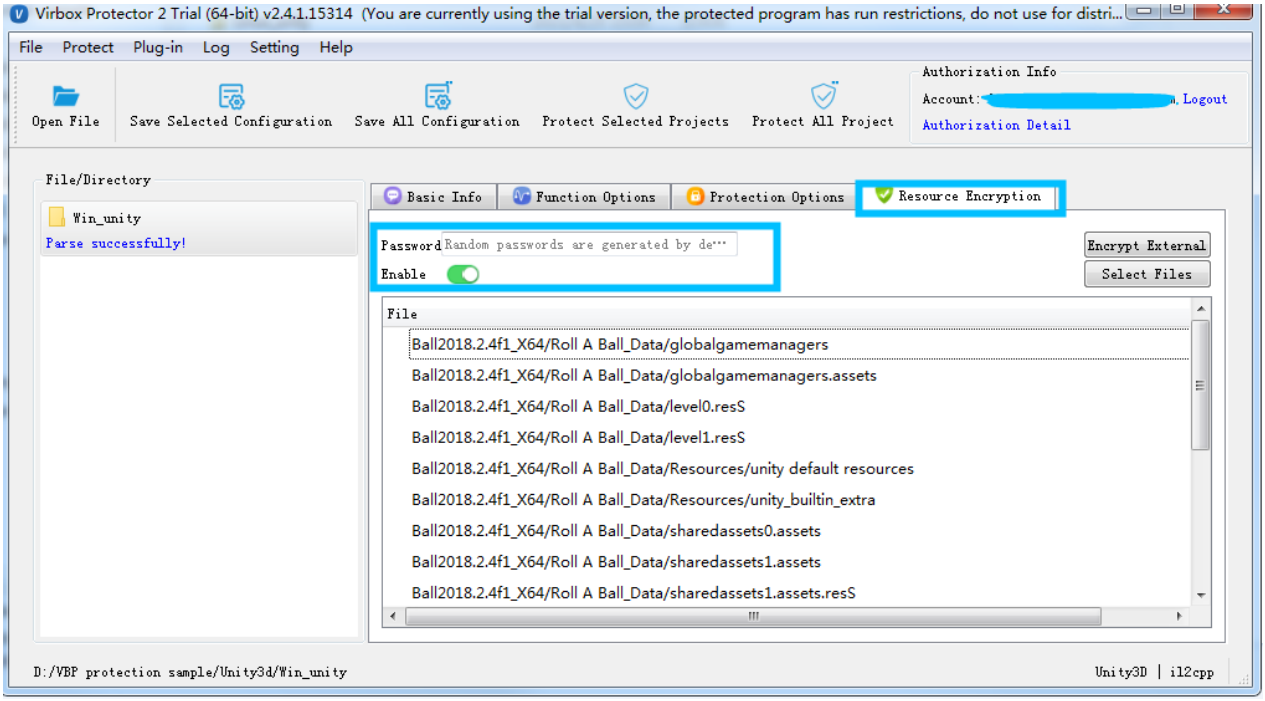


Figure 5-63

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**".

4. Click "Protect Selected Projects" to complete the protection process, Virbox Protector will generate a new folder, same directory with the original folder named SSP.TEST (TEST is the original Unity3D directory name), the picture shown below.

Two more files would be generated:

Figure 5-64

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 With Protection, you can distribute this file to the software user in future.

5.4.3.2 Hot Update the Protected Resource file

In most of cases, Unity3D developer may update the Unity3D resource and need to re protect these updated resource frequently. Virbox Protector (By using the DS protector) supports to protect the Unity3D resource file with "external Protection" way, that means, Virbox Protect (DS Protector) may protect these Resource file separately. and developer use the updated protected resource file to replace the old resource file. and no need every time developer have to protect both Unity3D program & resource file and save developer's time.

Note:

- Keep the ***consistent password*** when you protect the updated resource file. otherwise the Unity3d program may not open the updated protected resource file;
- Currently, This function only support following Unity3D applications: Unity3D mono in Windows and Unity3D mono in Linux Environment.

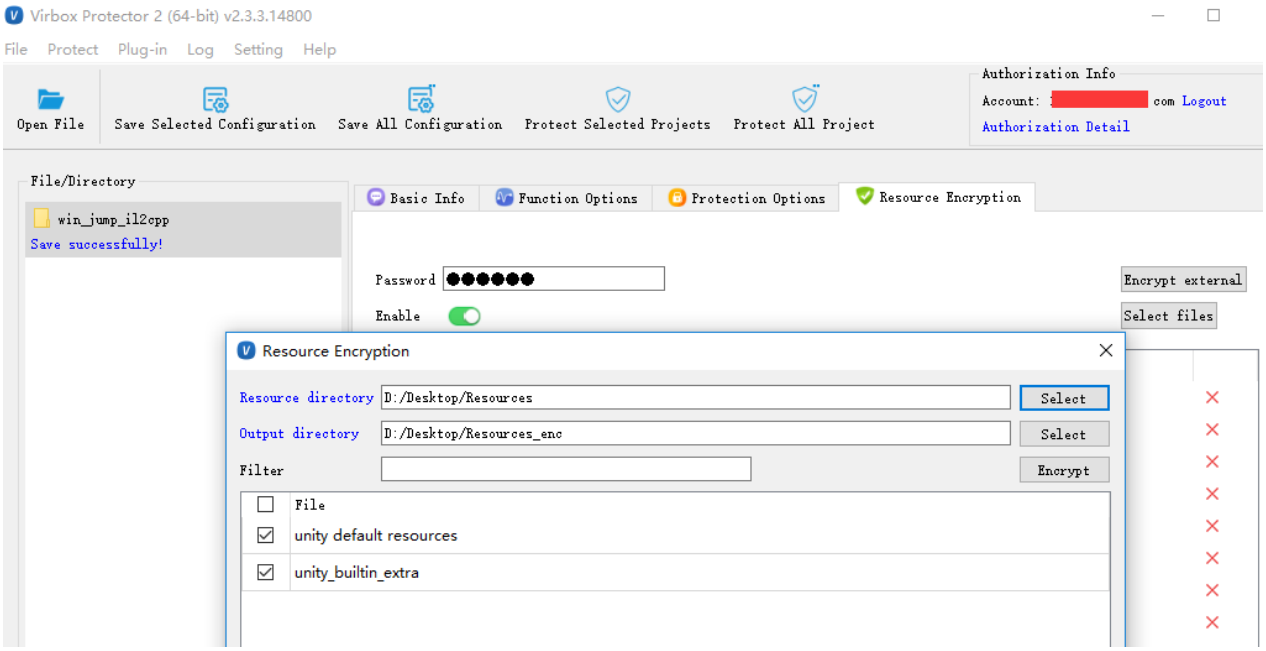


Figure 5-65

5.4.3.3 Using Command Line to protect the Unity3D program

5.4.3.3.1 Protect the Unity directory

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

- Open a console terminal, enter the path where "***virboxprotector_con***" is located, and enter "*virboxprotector_con*" to run Virbox Protector. Help information can be viewed.
- ***Command*** to protect U3D:

  virboxprotector_con <The Unity3D directory which need to be protected> -o <The directory output>

  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.

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

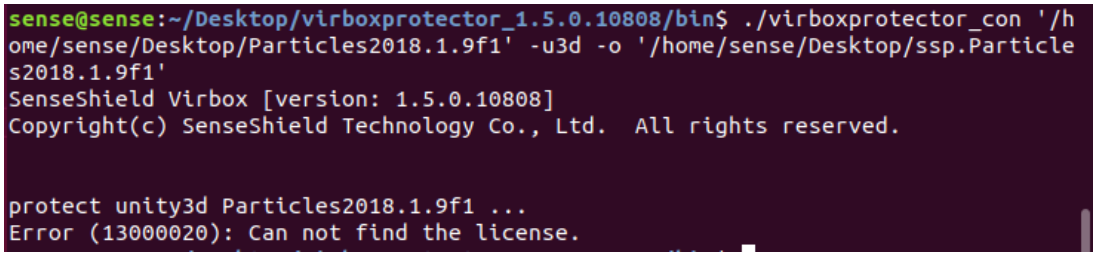


Figure 5-66

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

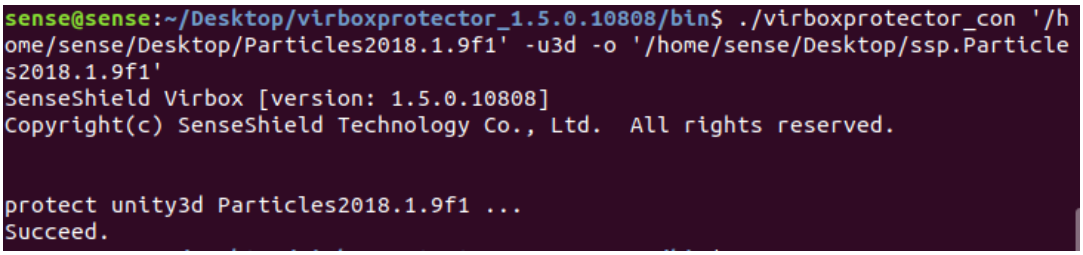


Figure 5-67

5.4.3.3.2 Protect and update the Unity3D Resource Directory

Following example is based on Linux Unity3D,

- Use the Virbox Protector GUI tool to generate configuration files
- 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.

Command:

virboxprotector_con -u3dres <The directory which need to be update Resource -x <configuration files> -o <The Directory which Resource output>

Note:

For Virbox Protector 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.

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

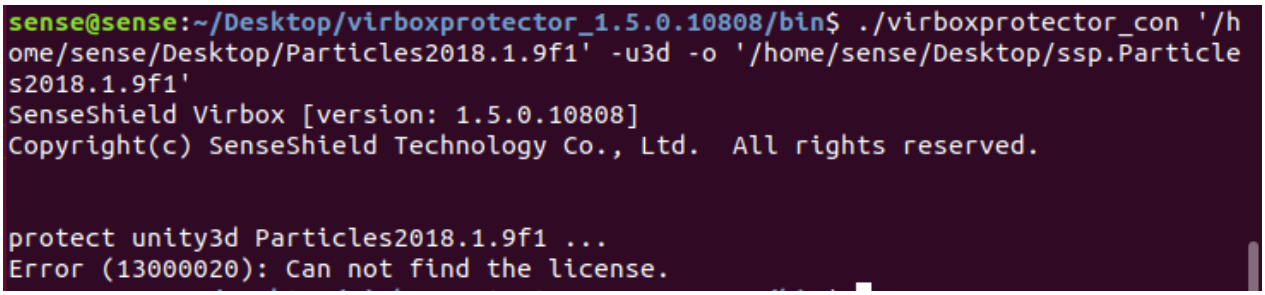


Figure 5-68

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

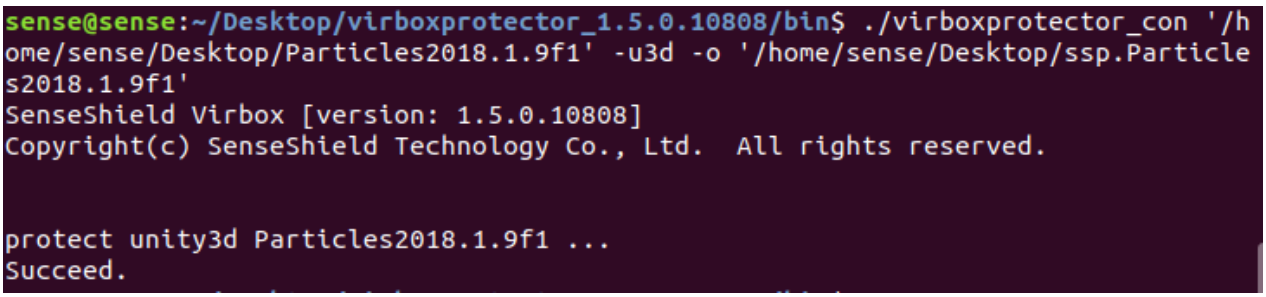


Figure 5-69

5.4.3.4 Protection comparison:

Without Protection:

Assembly-CSharp*.dll

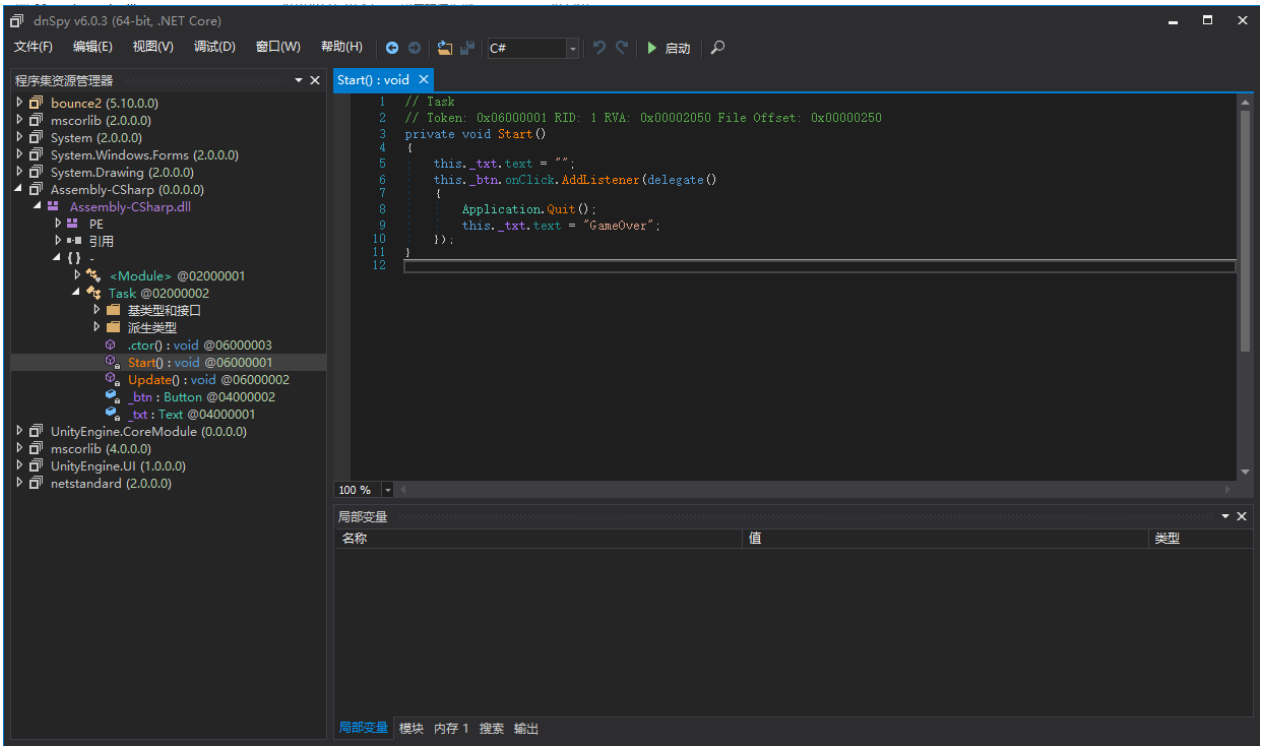


Figure 5-70

With Protection:

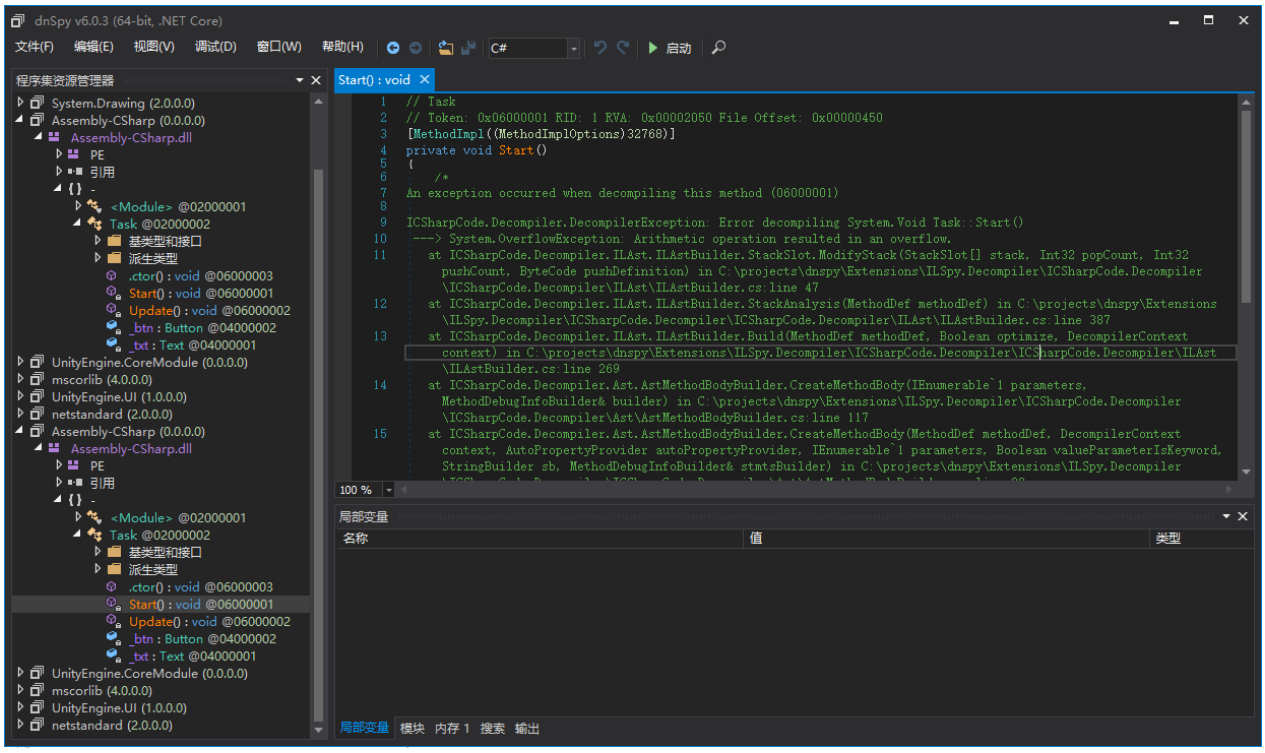


Figure 5-71

Comparison of protection to Unity3D resource file

The Decompiling to Resource file of Unity3D:

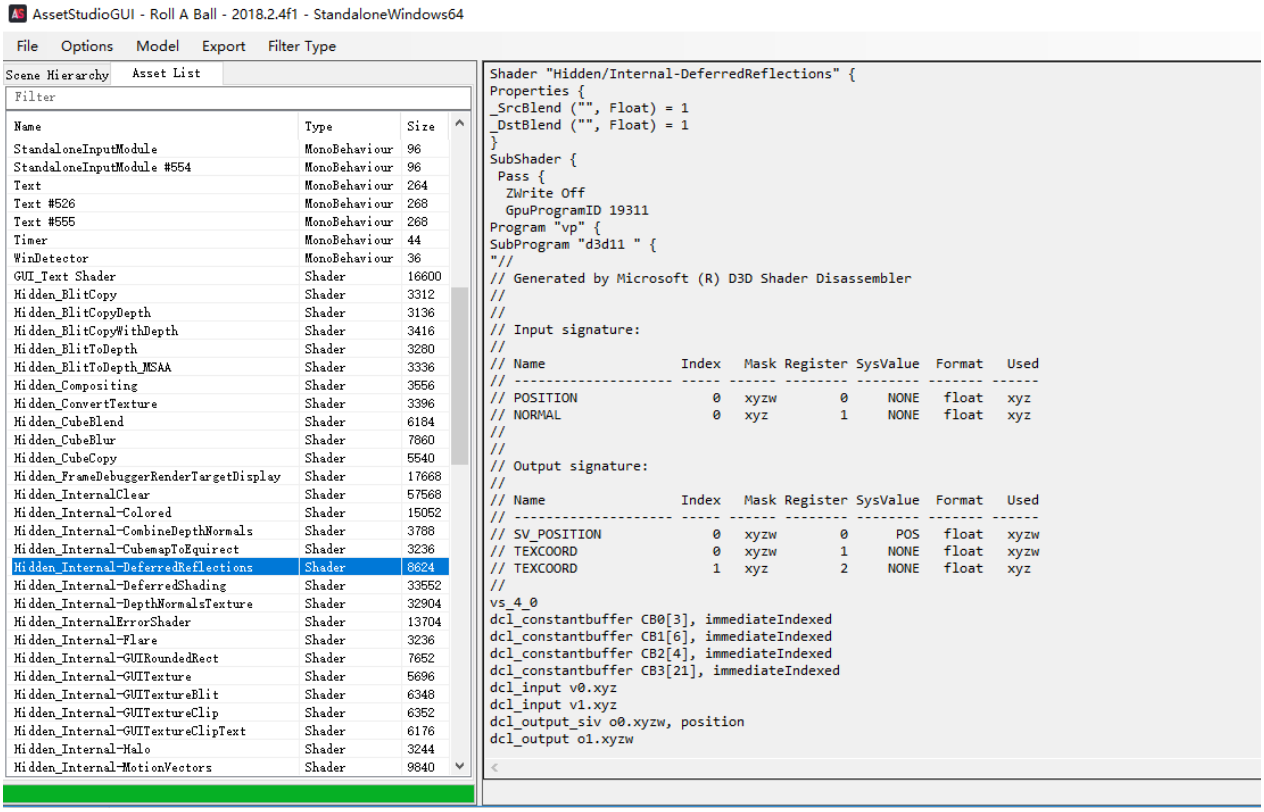


Figure 5-72

The Decompiling to Protected resource file of Unity3D:

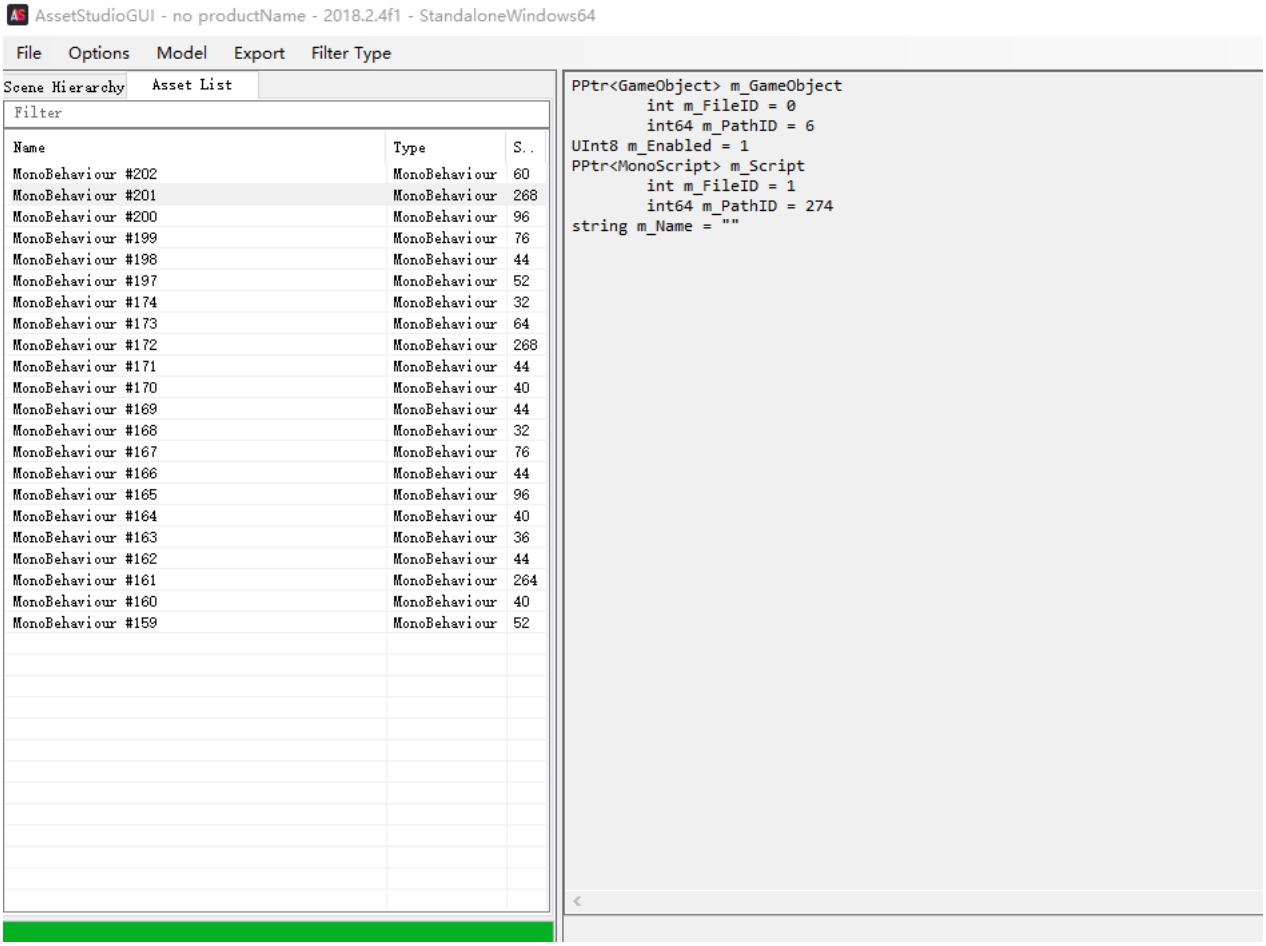


Figure 5-73

5.4.4 Protect Unity3D android application (Apk)

5.4.4.1 Protect android application with Virbox Protector GUI

Virbox Protector support to protect for both mono and IL2CPP based mobile Unity3D applications in Android systems, Developer who want to protect to Mobile Unity3D apk files, you just need to drag your Android apk into the Virbox Protector, then the Virbox Protector will recognized the Unity3D Apk based on mono frame or IL2CPP frame, then it will shown the relevant "protection option" in the "***Protection Option*** tab respectively.

Note: it will no difference for APK and AAB format in protection mode, Virbox Protector take same protection mode to protect the APK and AAB format.

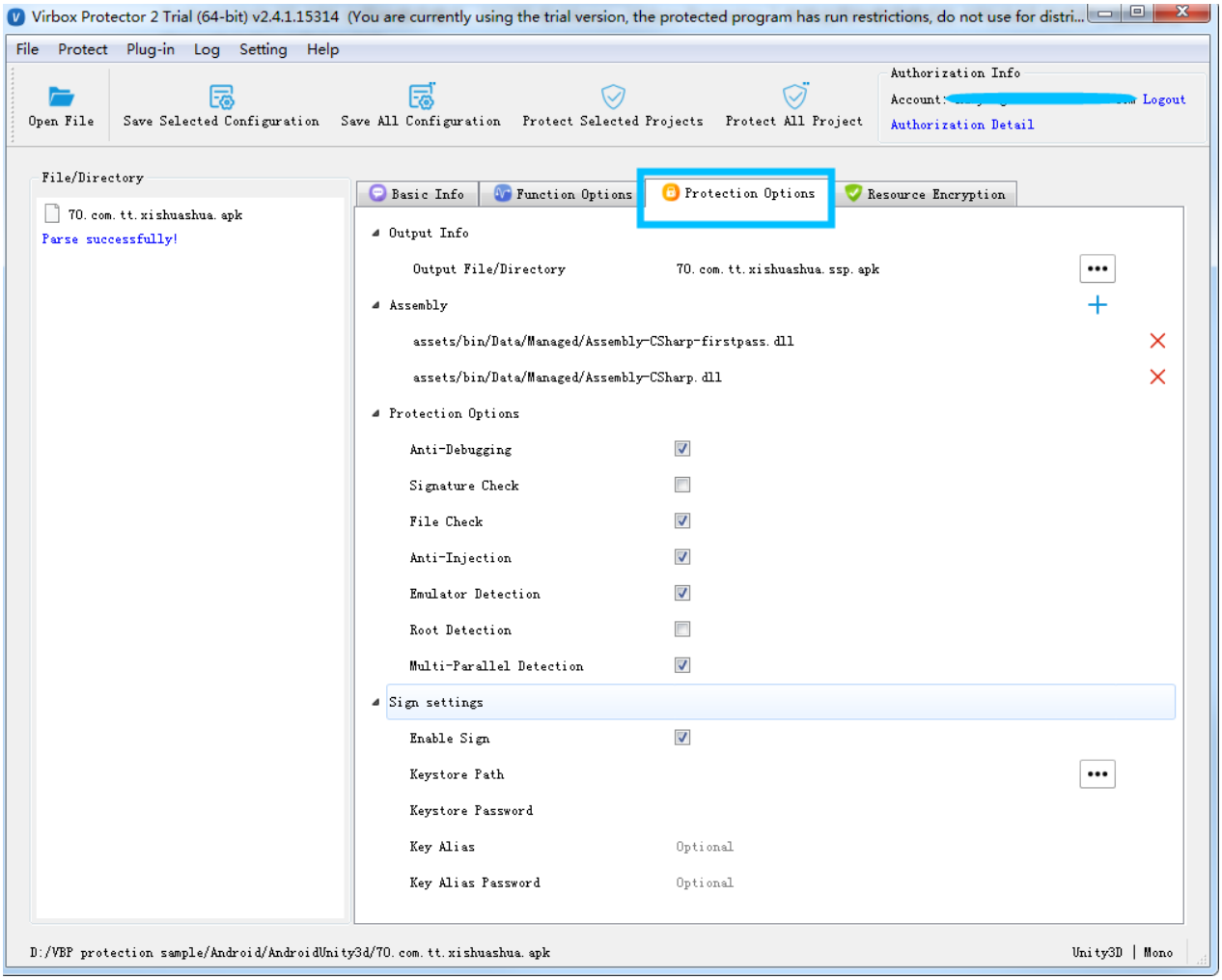


Figure 5-74

5.4.4.2 Protection Process

When you drag your Unity3D android apk into the Virbox protector, you can set and select following "Protection option" in the "***Protection Option***" panel:

Protection Option Setting:

Click and Select the "MEMO CHECK" button, to verify memo and check memo integrity(effective to IL2CPP project);

Click and Select the "META DATA ENCRYPTION" button, to encrypt meta data (effective to IL2CPP project);

Click and Select the "NAME OBFUSCATION OF META " button, to obfuscate the name of method metadata (effective to IL2CPP project);

Click and Select the "ANTI-DEBUGGING" button, then when the debugger use the IDA or other third party debug tools to debug the protected Android Apk, the protected APK will exit directly;

Click and select the ANTI-INJECTION, to prevent the other session to add debug or injection.

Click and select "EMULATOR DETECTION", to prevent the App running in the emulator environment'

Click and Select "ROOT DETECTION" to prevent the App running in the rooted device;

Click and Select "MULTI-PARALLEL DETECTION" to prevent app running in multiple account;

Signature setting:

Click and select Enable Sign, to enable the signature verification, then you need to set the keystore file path and password, the signature will be automatically signed after encryption, if not enable sign, the encrypted APK need to sign separately;

Resource Encryption

Now go to the "***Resource Encryption***" tab to protect the resource file in the Unity3D apk;

Switch on the "ENABLE" button in the "***Resource Encryption***", then you can encrypt the resource file of the Unity3D program

if you set the password to protected resource file, then it will be used to be the seed to encrypt the resource file every times, otherwise, it will use random key to encrypt;

If you use same password to encrypt the updated resource file, then the encrypted resource file (under \assets sub directory) will be used and replace previous encrypt file directly.

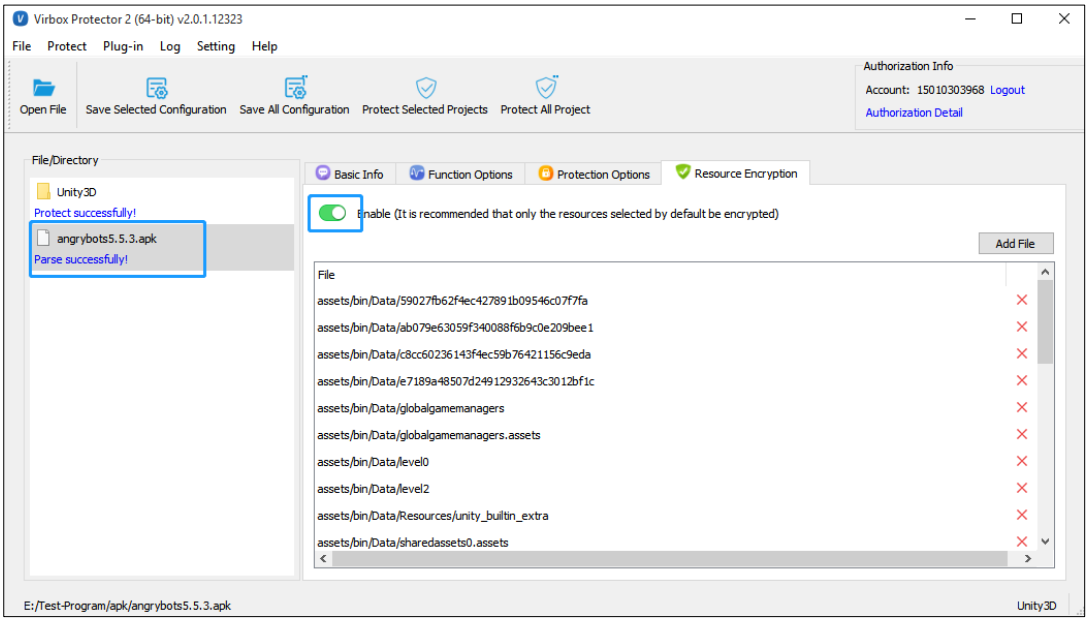


Figure 5-75

After complete above setting, click "Protect Selected Project" to complete the protection process.

Virbox Protector will protect the "**libmono.so**" or **"libil2cpp.so"** and "**Assembly-CSharp.dll**".

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

Before Protection:

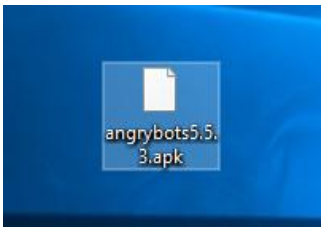


Figure 5-76

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

After Protection:

Figure 5-77

With Protection,

“**angrybots5.5.3.ssp.apk**” is the file With 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.

5.4.4.3 Use command line to protect Unity3D android Apk.

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

1. Use the Virbox Protector GUI to generate “.ssp” configuration file (Optional)
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” “the directory of the android apk to be protected” -u3d –o “the directory of the output path of the Android apk”

If the license to unity3D program can not be found, it shows as the snapshot below:

```
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 5-78

If the license of Virbox protector has been verified, then:

```
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.
```

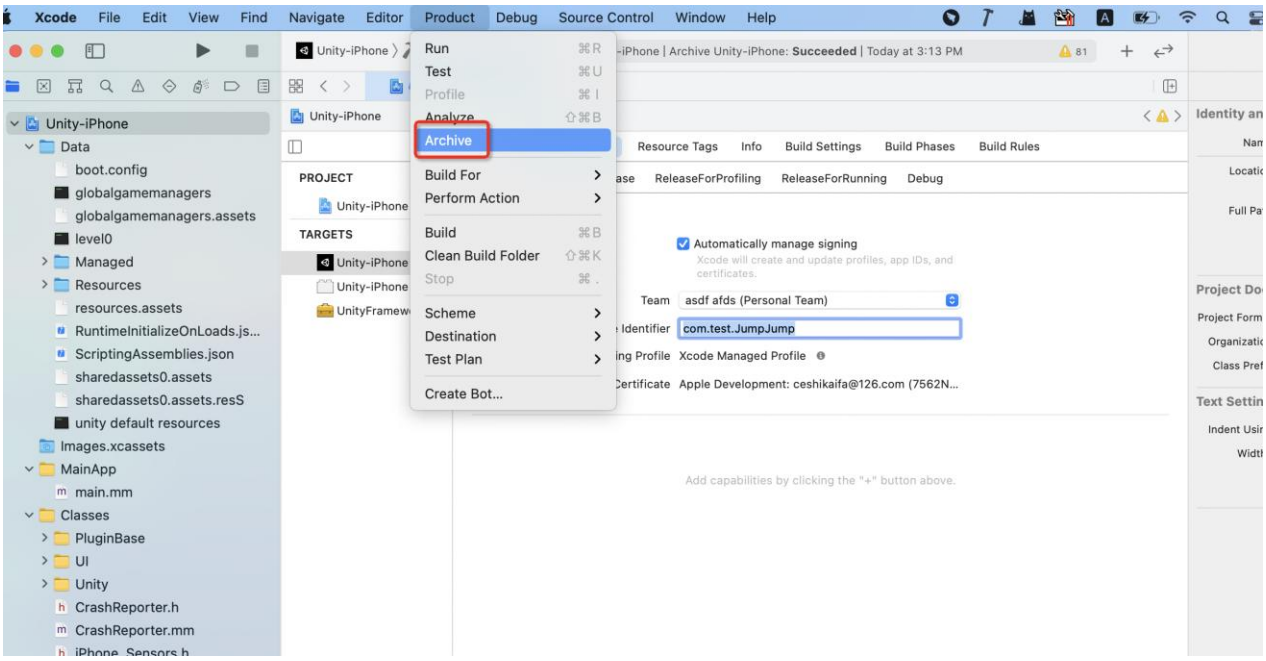
Figure 5-79

5.4.5 Protect Unity3D mobile application in iOS platform

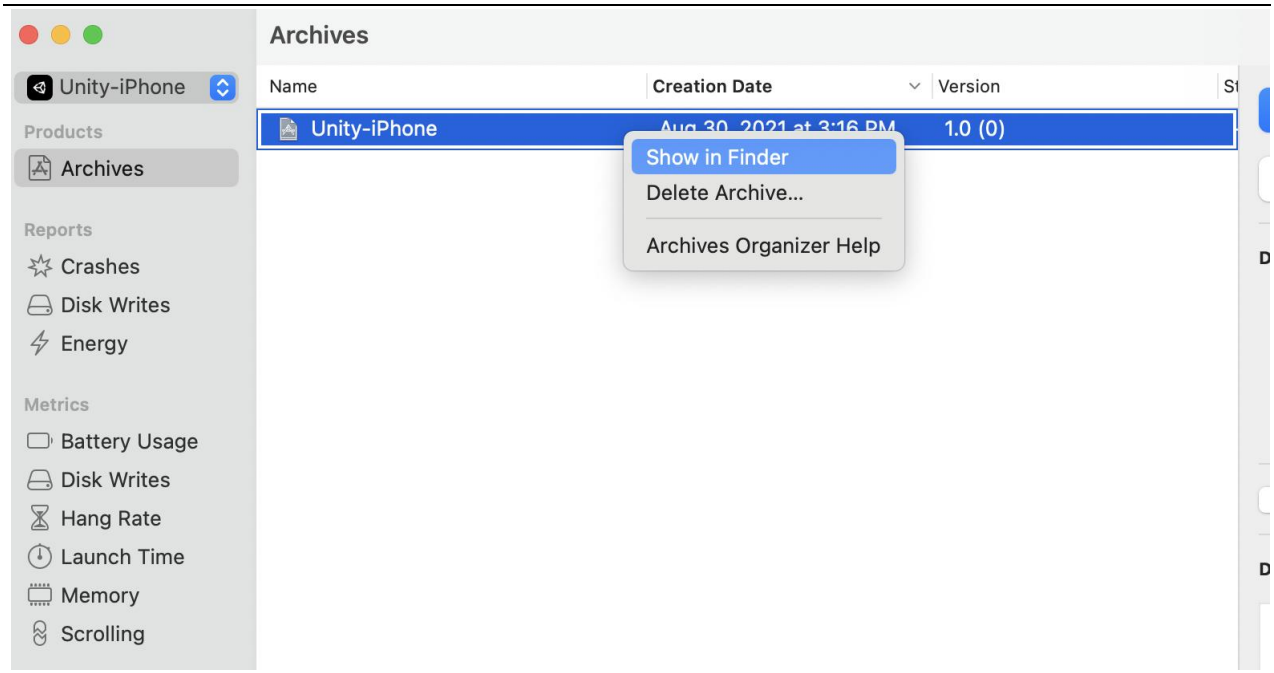
5.4.5.1 Protection Process by Using *Virbox Protector GUI* tools

For the Unity3D application for iOS platform, it is necessary to generate *xcarchive* first:

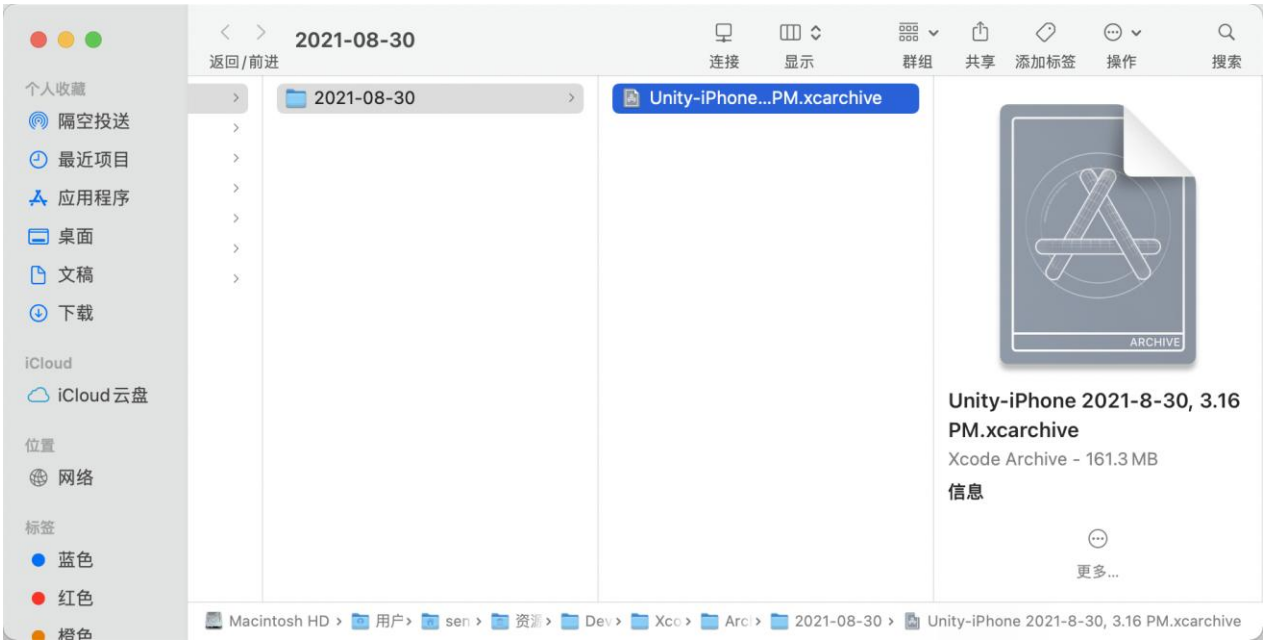
Open the Xcode, click "Product"-> Archive:



Go to "Archive" right click, find the file location in the "Finder"



Find the xcarchive



then you can drag the app which located in *"xcarchive\Products\Applications"* into the Virbox Protector;

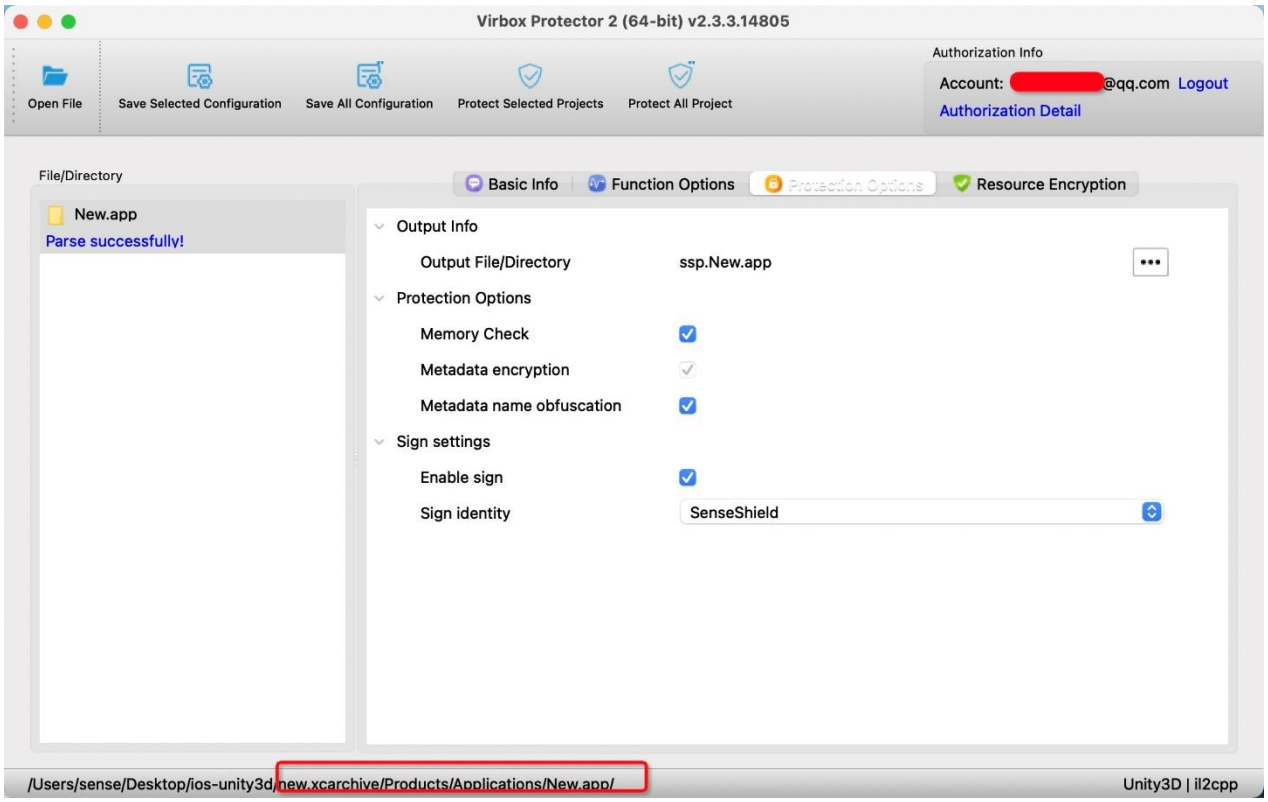


Figure 5-80

Click and select the "sign", this option will enable to sign a signature to the protected unity3D application automatically;

Click and “Enable” the "Enable" button, to protect the resource file;

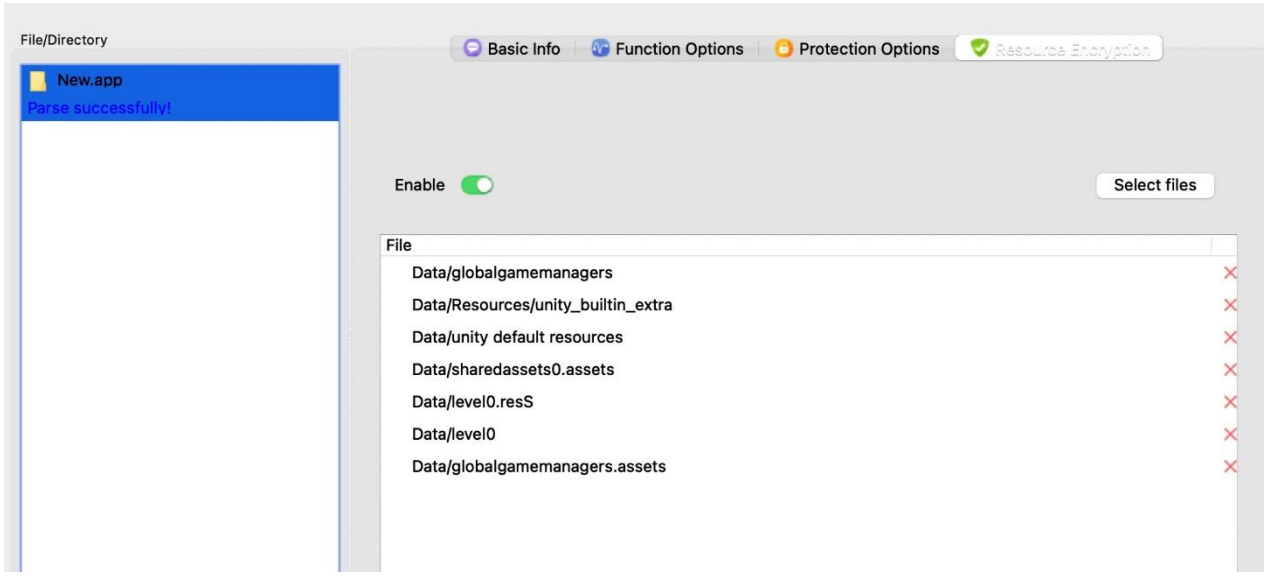


Figure 5-81

Click "Protect the selected project" or "Protect all Projects" to protect your unity3D application;

Note:

you can protect the xcarchive directly;

if you copy the unity3D app to other directory, you need to put the *dSYM* file and *Unity3D* apps in same directory before protect it.

5.4.5.2 Protect Unity3D apps in iOS by using *Virbox Protector CLI* tools

1. Use the Virbox Protector GUI generate ".ssp" configuration file (if no configuration file has been generated, then no signature will be signed to the protected application on default)
2. Open the Command line terminal in window, get into the "*Virboxprotector_con.exe*" directory, and input "*virboxprotector_com.exe*" , you can see the help info;
3. Command to Protect your app: *"Virboxprotector_con" "the app to be protected" –o " the output of the protected app"*

5.4.6 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:
   - Usually, the **mono.dll** located in the location: **.\Editor\Data\Mono\EmbedRuntime** directory

Figure 5-82

   - **and, mono-2.0-bdwgc.dll** located in the location:

     .\Editor\Data\MonoBleedingEdge\EmbedRuntime

Figure 5-83

2. You can create a "Dummy" Unity3D directory, the directory which need to be same with the Unity3D program directory. This is to help Virbox Protector can recognize and protect 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***

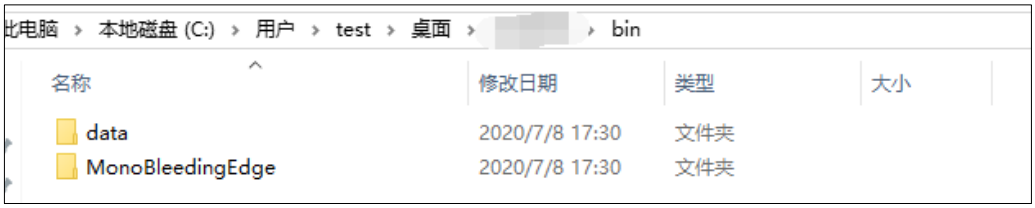


Figure 5-84

- Put the dynamic link library to the location: ***bin\data\Managed, here we take (demo.dll***) to be example, see snapshot shown below:

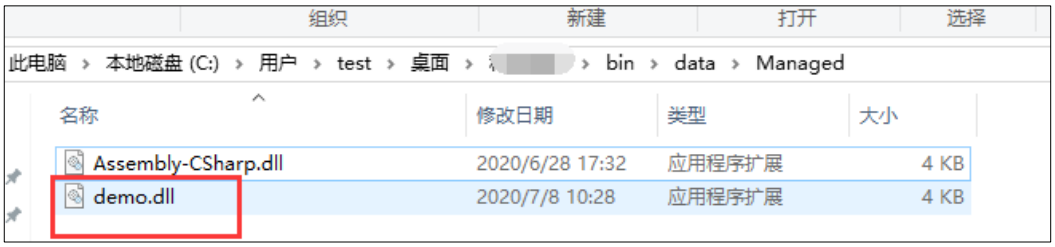


Figure 5-85

3. Drag the ***bin*** directory to the Virbox Protector:

- login Virbox Protector License

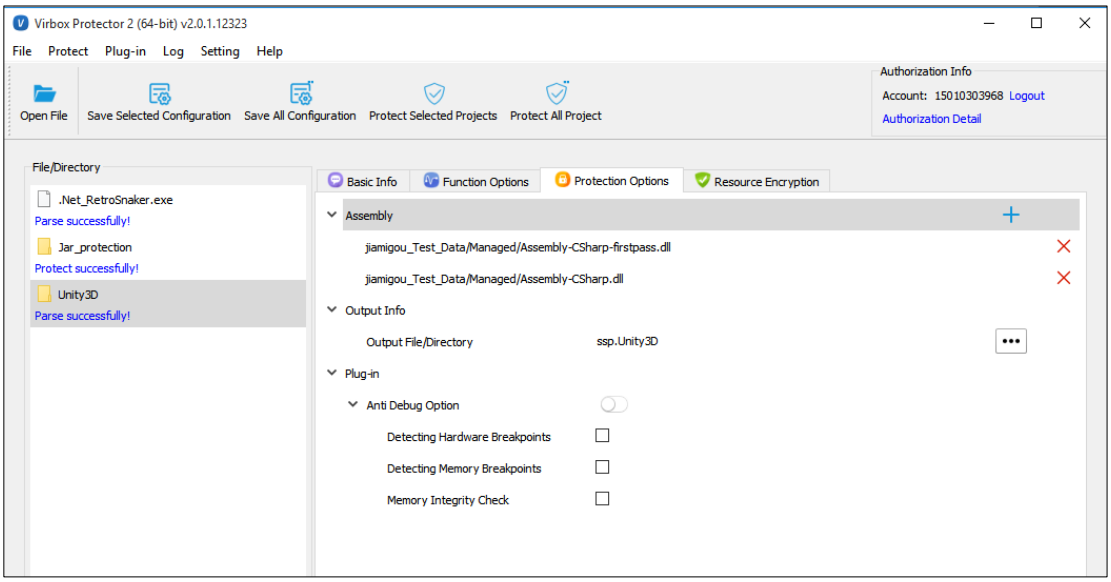


Figure 5-86

- Click to Add the *assembly* set in the "***Protection option***":

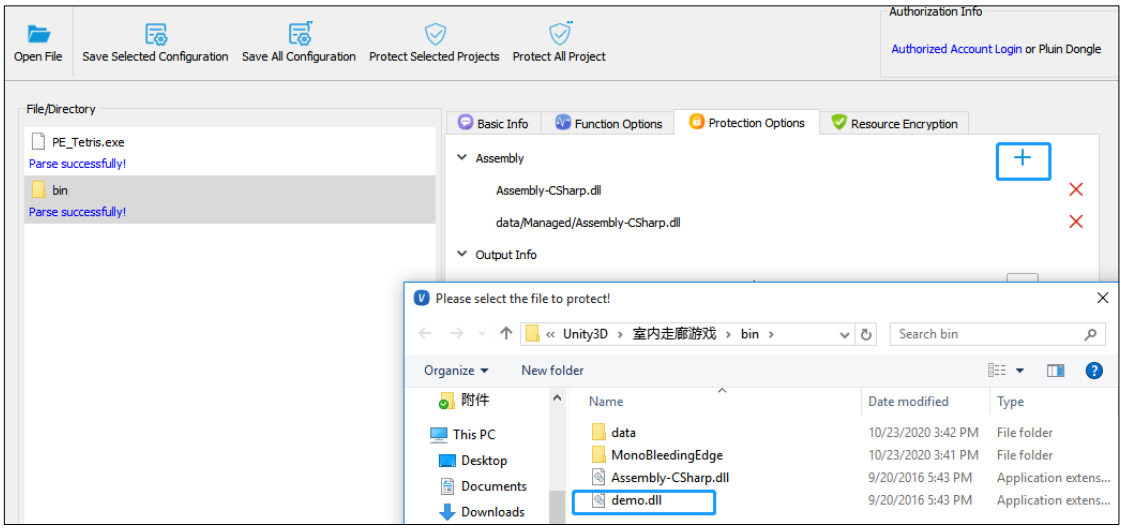


Figure 5-87

- After the assembly set is added successfully, delete the non-necessary program set and keep your own plugin dll.

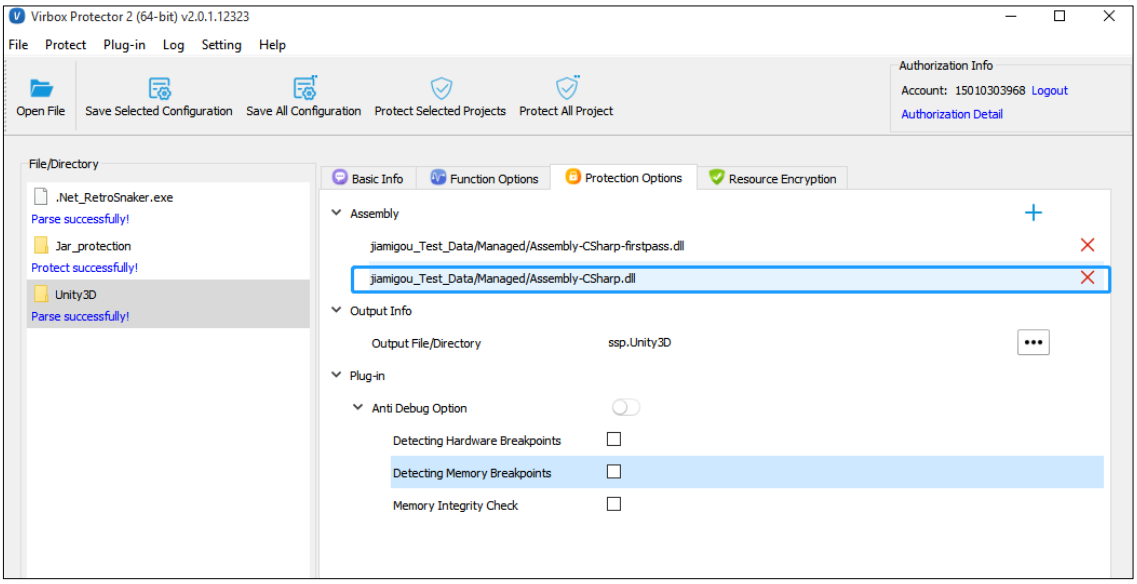


Figure 5-88

- Click "***Protect the selected project***" to protect the program, the ***ssp.bin*** will be generated. This is the program With Protection shown in below:

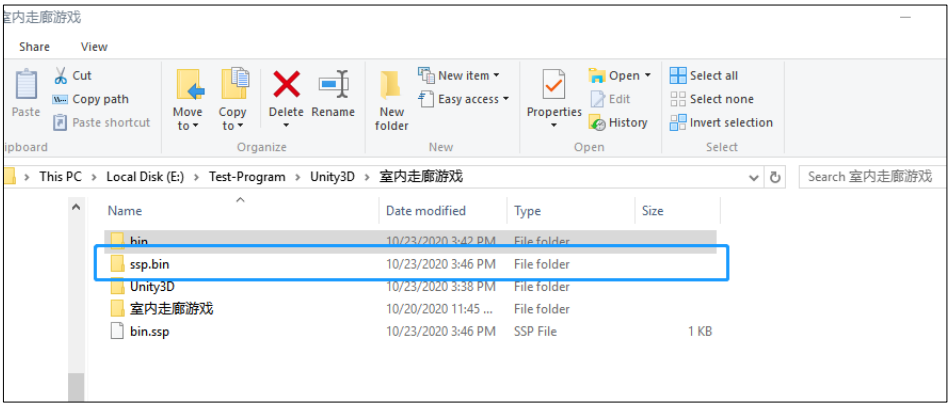


Figure 5-89

4. Go into the ***ssp.bin*** directory, copy the

   "mono-2.0-bdwgc.dll"

   which located in the ***ssp.bin\MonoBleedingEdge\EmbedRuntime*** directory

   To "***Editor\Data\MonoBleedingEdge\EmbedRuntime***" (pls backup the original file in advance), then put the "***demo.dll***" in the directory "***ssp.bin\data\Managed***" into project.

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

5.4.7 Protection Comparison

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

Without Protection/encryption:

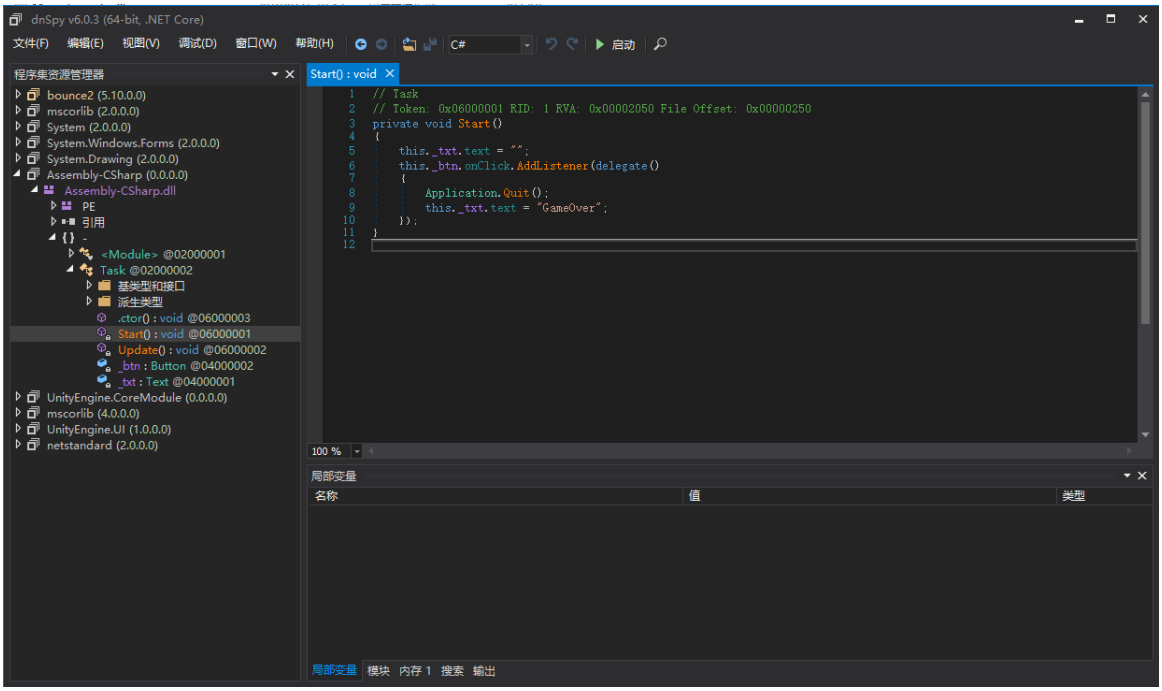


Figure 5-90

With Protection/encryption:

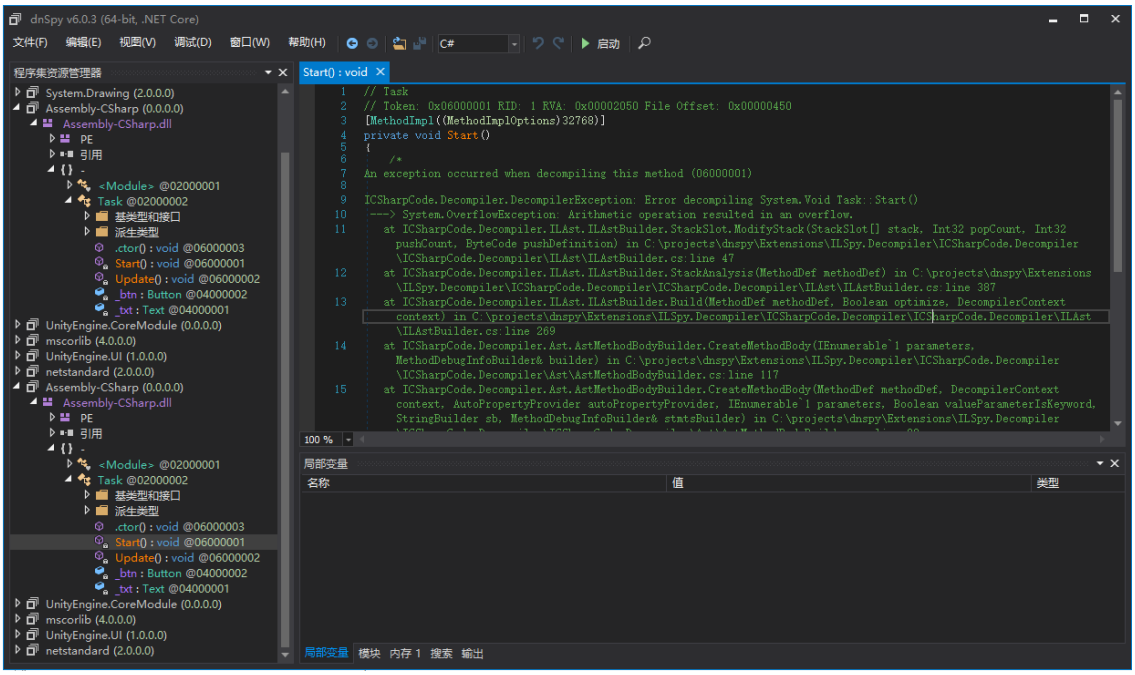


Figure 5-91

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

Without Protection:

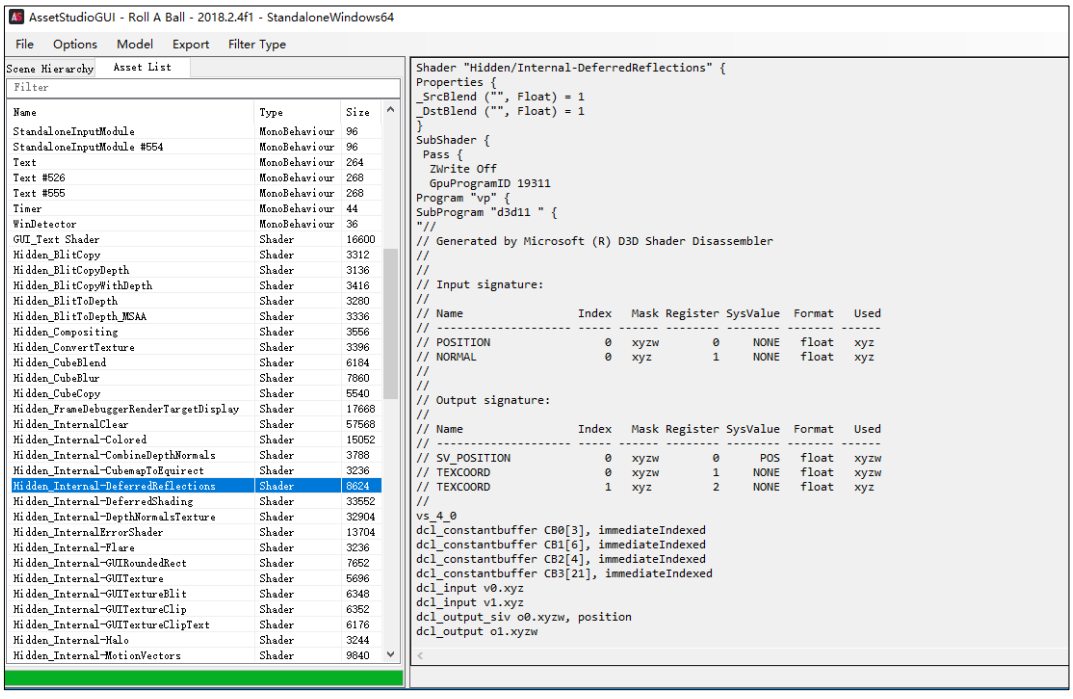


Figure 5-92

With Protection:

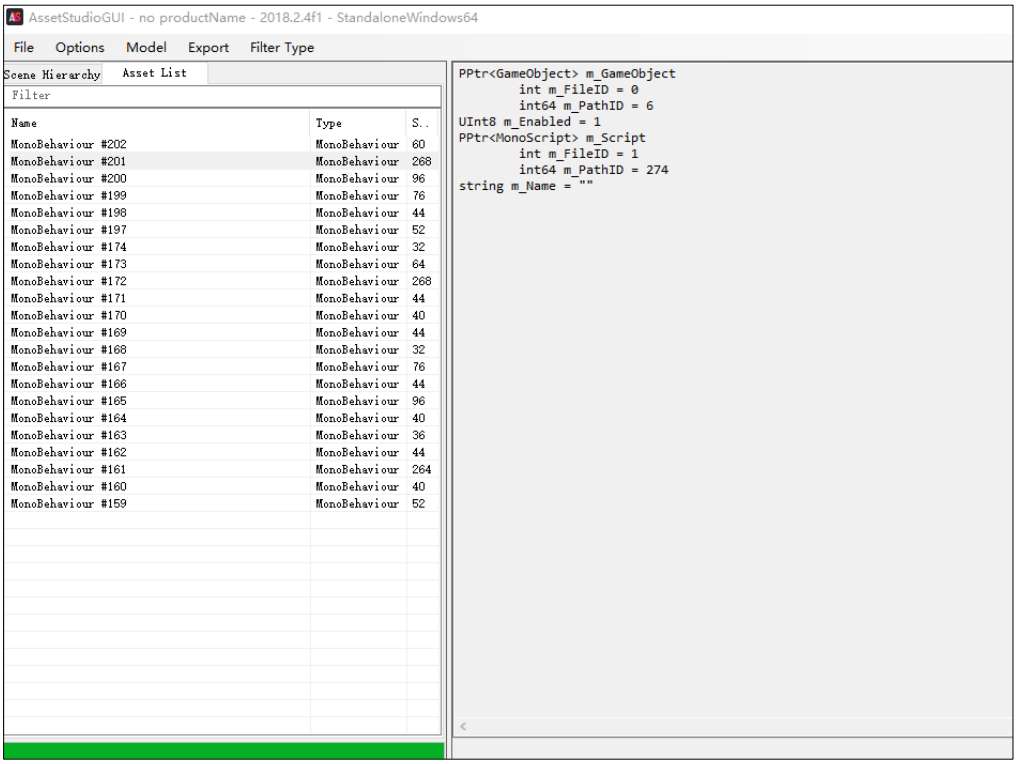


Figure 5-93

5.5 Protect Android application

Virbox Protector supports developer to protect Android Applications in different kind of format, includes .so libs, APK, AAR and latest AAB format.

5.5.1 Protect the .so libs

For normal apk application, the application need to be unzipped and then to protect the .so library which located in lib directory by use of Virbox Protector, see snapshot shown as below:

Figure 5-94

To protect the .so libs, following options you may select in the "***Protection Option***": ***Compression***, ***Memory Check***, ***Anti-Debugging***.

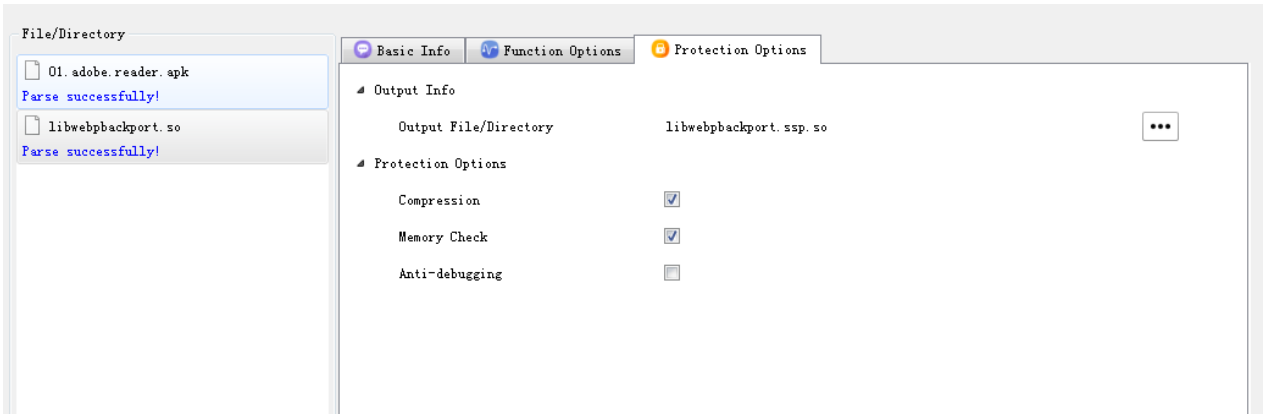


Figure 5-95

Also, you can select the concrete protection mode to specified "function" in the "***Functions Option***".

When complete protection to .so libs, put the libs into the apk accordingly.

5.5.2 Protect the APK/AAB application

For APK/AAB application, Developer may select following "protection mode" in the "Protection Option" to protect the Android APK file:

Anti-Debugging, This option will effectively defend the cracker using the debug tool to debug the apk and prevent cracker to parse and get the source code by use of IDA, or other reverse-engineering tool

Signature check: Select this option and setting the signature by input your keystore file and password (Key alias and password could be optional) to prevent APK repackaging and protect the critical coding, core algorithm/logic;

Anti-Injection: Click and Select this option, it will prevent the other process to add debugger or injection to your APK Process.

The other protection feature you may select includes: Emulator detection, Root detection and Multi-Open detection, you can select these feature accordingly.

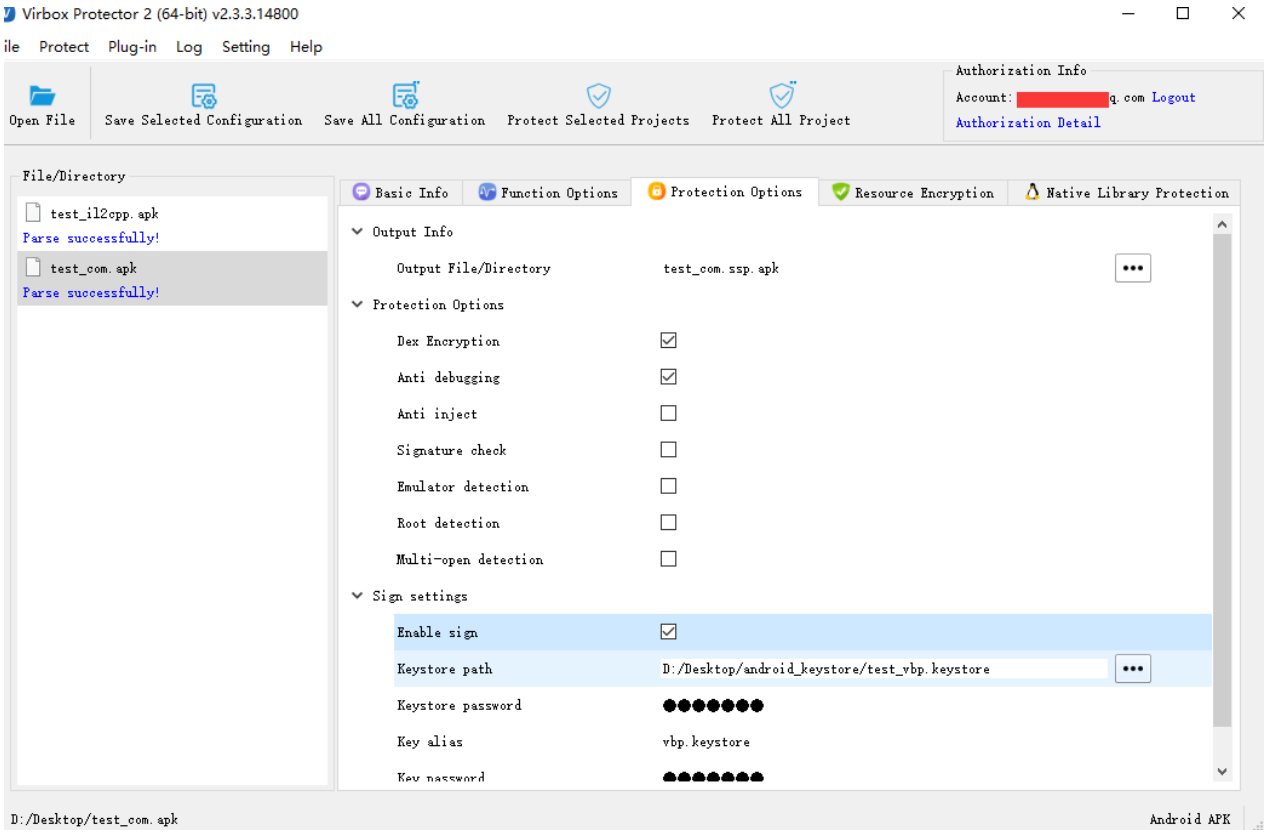


Figure 5-96

Protection Process:

Drag your Android APK into the Virbox Protector;

Protection Option tab:

You may click and select the protection option in the "***Protection Option***" panel:

Select the "Anti-Debug" option, the protected apk will exited when cracker use the IDA, or other debug tool to debug the apk;

Select the "Signature check" and set up the signature, input your "keyStore" file and password

Select the "Anti-Inject", to prevent the other "Process" to add the debugging or injection to your APK process.

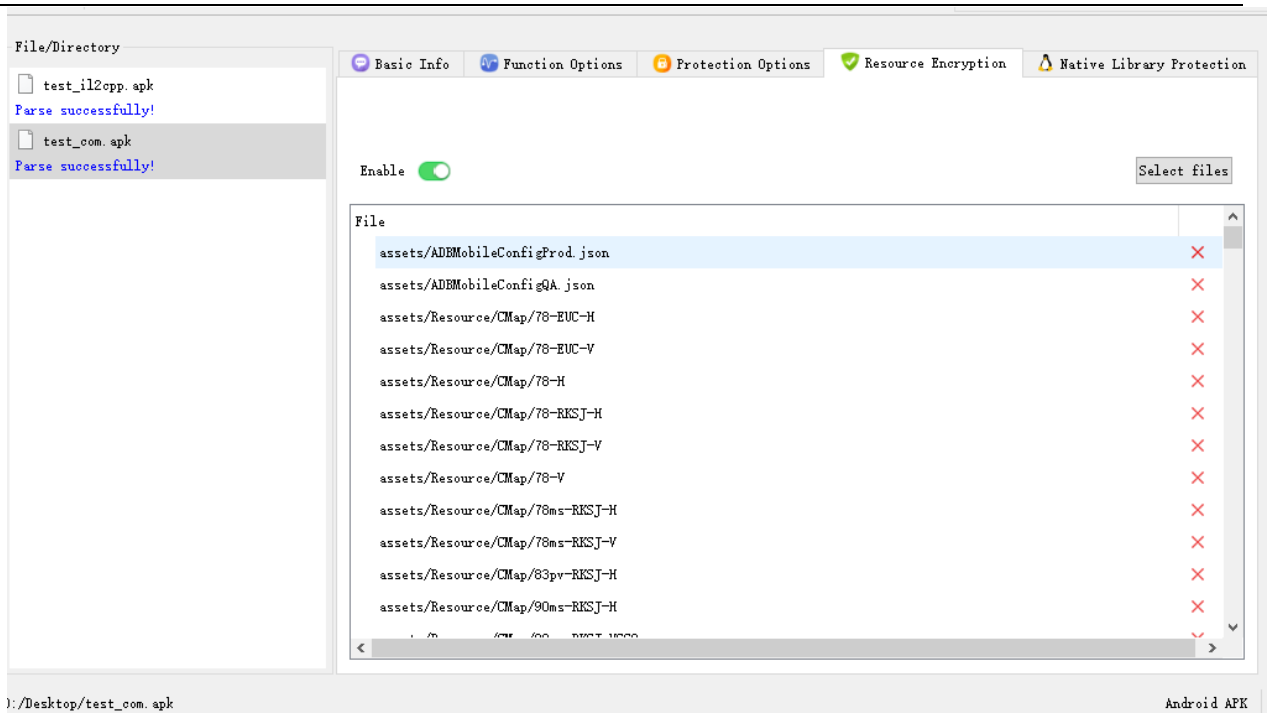


Figure 5-97

Resource Encryption tab: To Encrypt the resource file under /assets directory of your APK, support to encrypt the picture, configuration, and script files;

if you set the password to protected resource file, then it will be used to be the seed to encrypt the resource file every times, otherwise, it will use random key to encrypt;

If you use same password to encrypt the updated resource file, then the encrypted resource file (under \assets sub directory) will be used and replace previous encrypt file directly.

Native lib Protection tab: click "***Native Lib Protection***" Option and click to add the .so libs which need to be protected;

HIDE SYMBOL Option, click and select this Option, to hide the "export function" of the .so libs (this option will be available to "Select all .so libs only)

Note: the .so libs selected with this option, can be supported for "Compression" only, if you want to protect to specified "Functions" of .so libs, pls protect the .so libs separately.

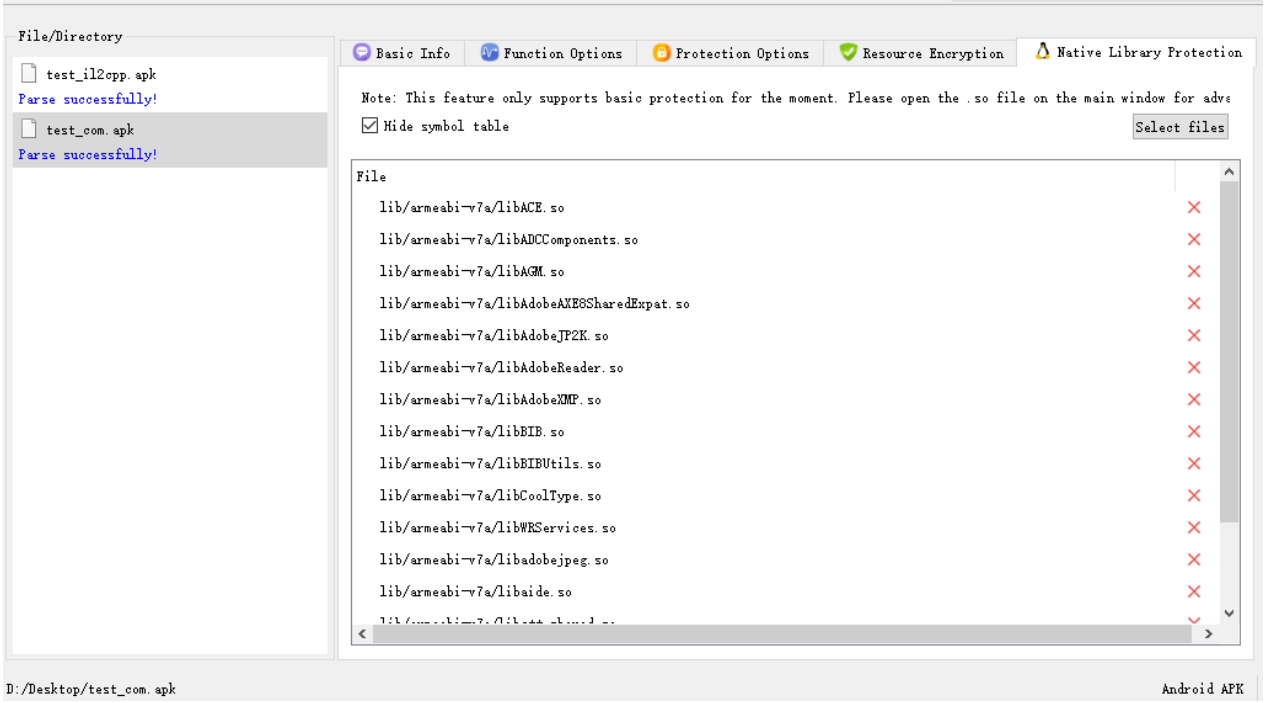


Figure 5-98

5.5.3 Protect the AAR files

Virbox Protector support to protect/harden the AAR format files, to protect the method of JAR archive by use of Virtualization and other comprehensive technology to defend the reverse engineering tool to retrieve the source code.

Protection Process

Drag the AAR archive into the Virbox Protector GUI tools directly;

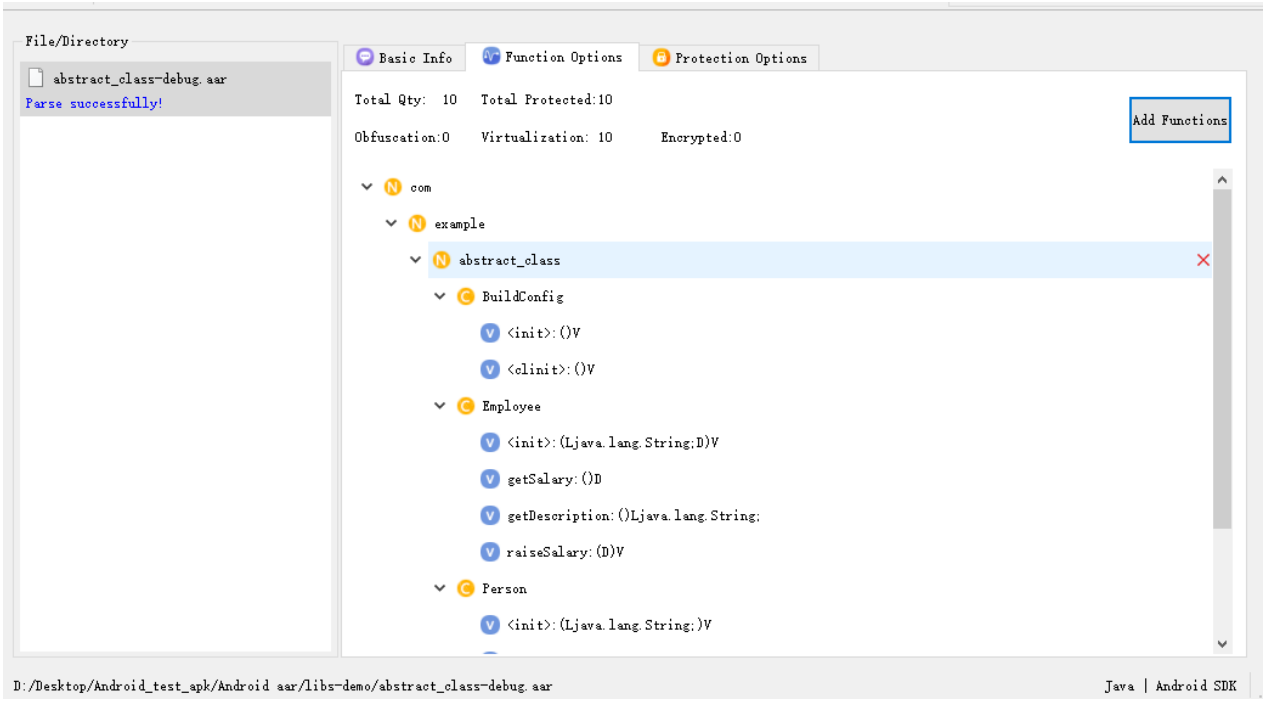


Figure 5-99

in "***Functions Protection***" tab, you can find the functions list in this tab, you can select the "Virtualization" to specified functions which you want to protect.

Click "***Protect Selected Project***", to compete the AAR file protection, then you can distribute the AAR package to your user.

Using the Command line to protect AAR

1. Using Virbox Protector GUI tool to generate the .ssp, the configuration file;
2. Open a Terminal in windows, go to the directory of *virboxprotector_con*, input "*virboxprotector_con*" to view the help info;
3. Use Following command to complete the protection:

   virboxprotector_con <The so/apk/aar which need to be protected> -o <The Output so/apk/aar>

5.5.4 Protect the AAB files

pls refer the User Manual of Protect AAB separately

5.6 Protect the iOS application

5.6.1 Protection Process with Virbox Protector GUI tools

1. Go to the directory which iOS App archive located and drag the iOS App archive into the Virbox Protector

GUI tools, Virbox Protector support to protect 2 kinds of file format: iOS App archive file format and executive file of iOS App, developer may select either of them and drag it into Virbox Protector GUI

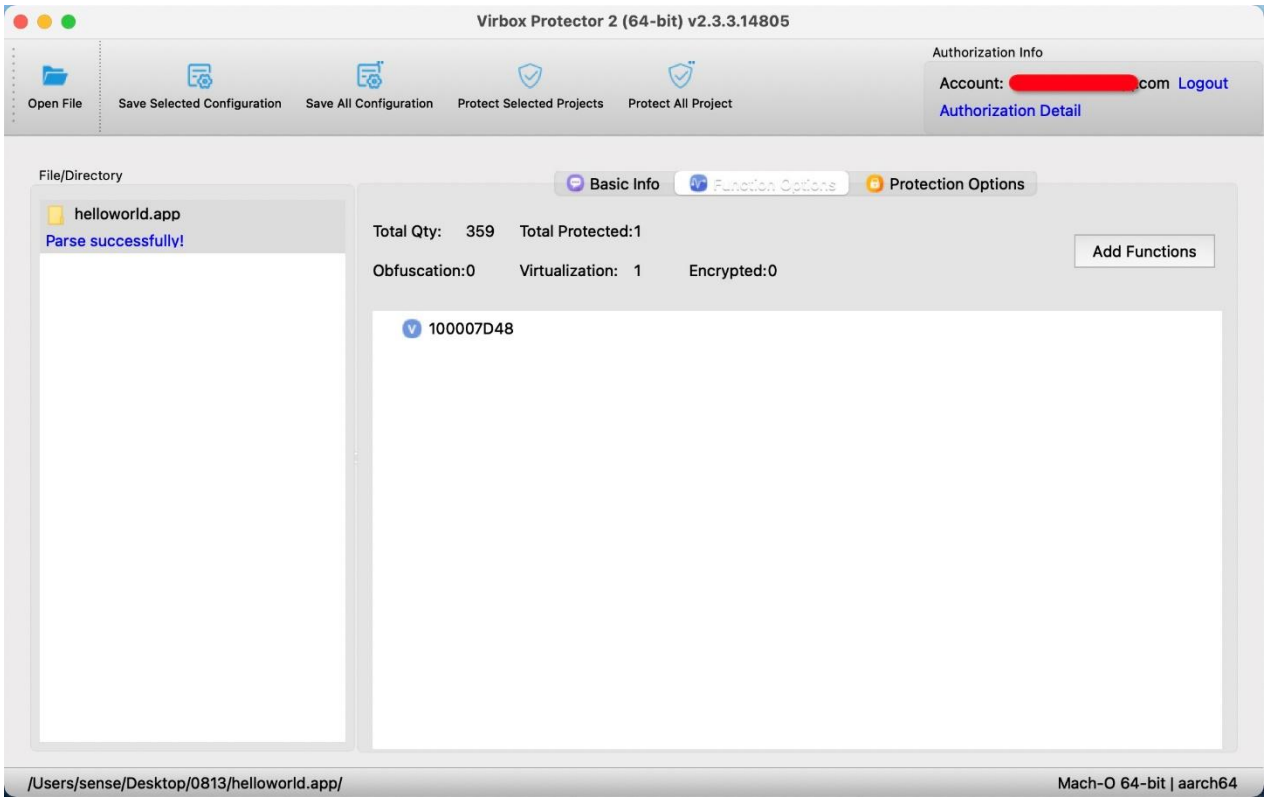


Figure 5-100

2. In the "***Function Protection***" pane, Click "Add Functions" to select the "Function" which you want to protect;

3. To each "function" which you want to protect, you can select following option to each "function" be protected: Virtualization, No Protected and Obfuscation;

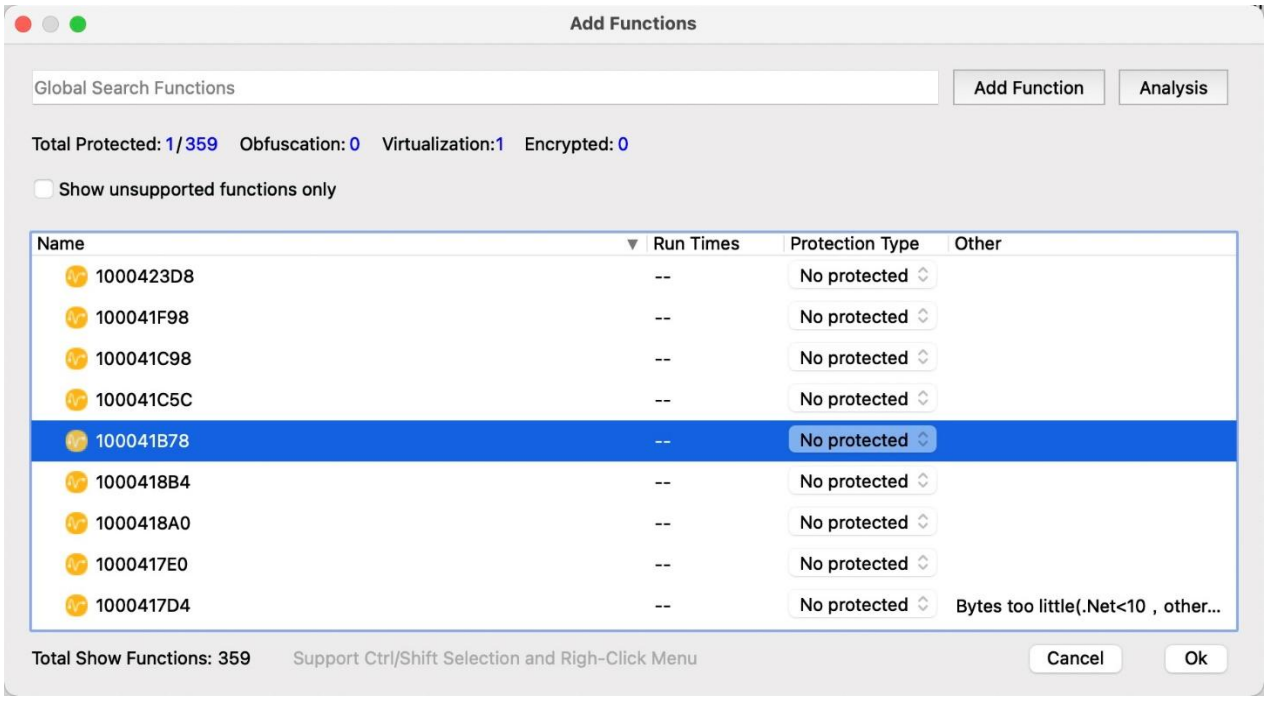


Figure 5-101

Note:

pls put the "dSYM" file into the directory of iOS App archive located, then Virbox protector can parse and get the function's name, otherwise only the address will be shown;

The "dSYM" file location, as shown as belown:

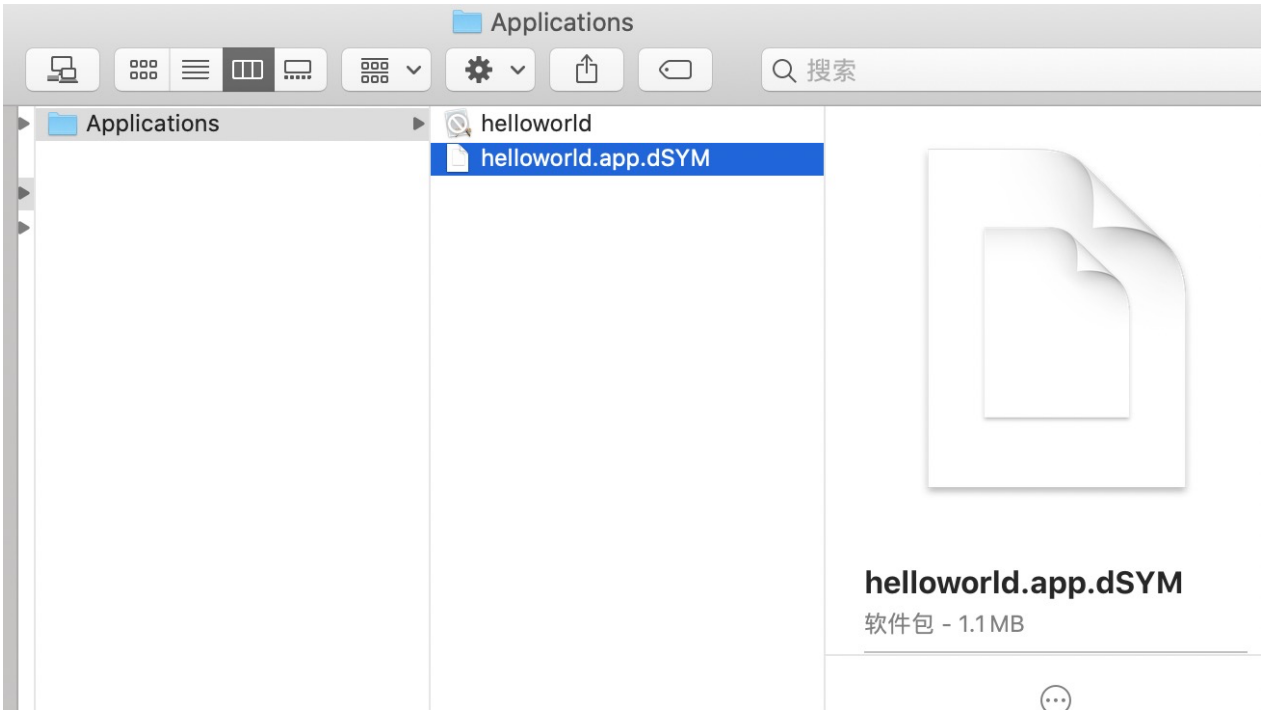


Figure 5-102

4. In the "***Protection Option***" , you will have following option can be select/set:

Memory Check, to prevent App tampering;

Anti-Debugging, detect the debug tools, to prevent the dynamic debugging;

and Objective-C name Obfuscation, sign setting etc.

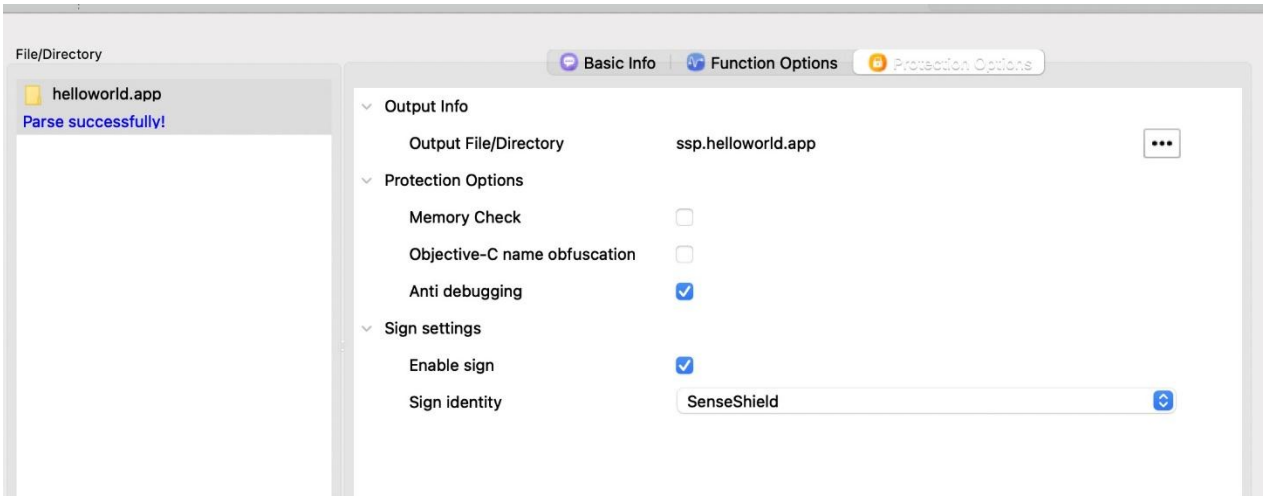


Figure 5-103

5. Select the Output Folder;

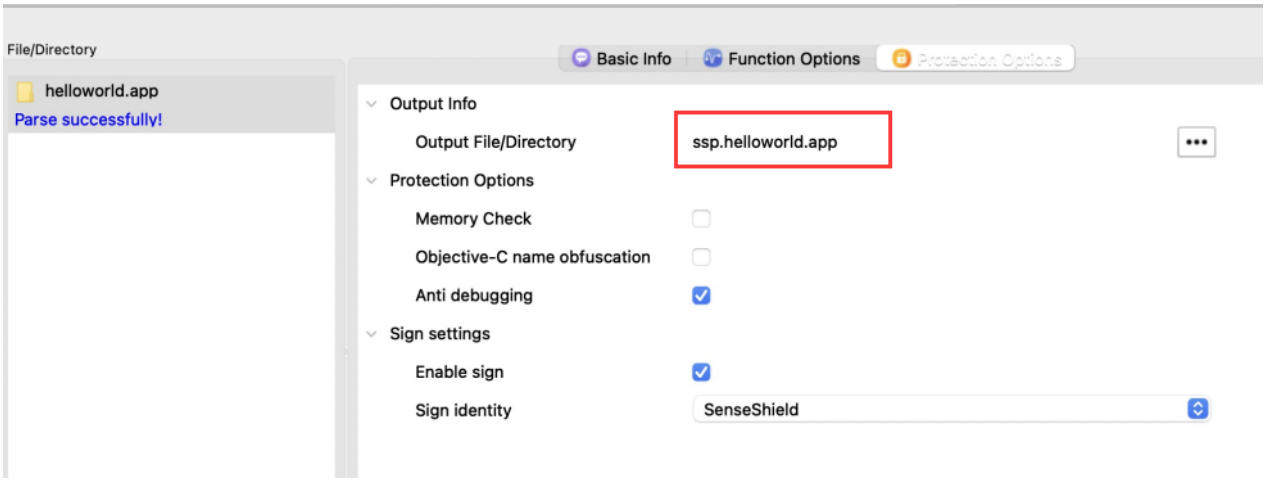


Figure 5-104

6. Click the "***Protect selected Project***", to complete protection.

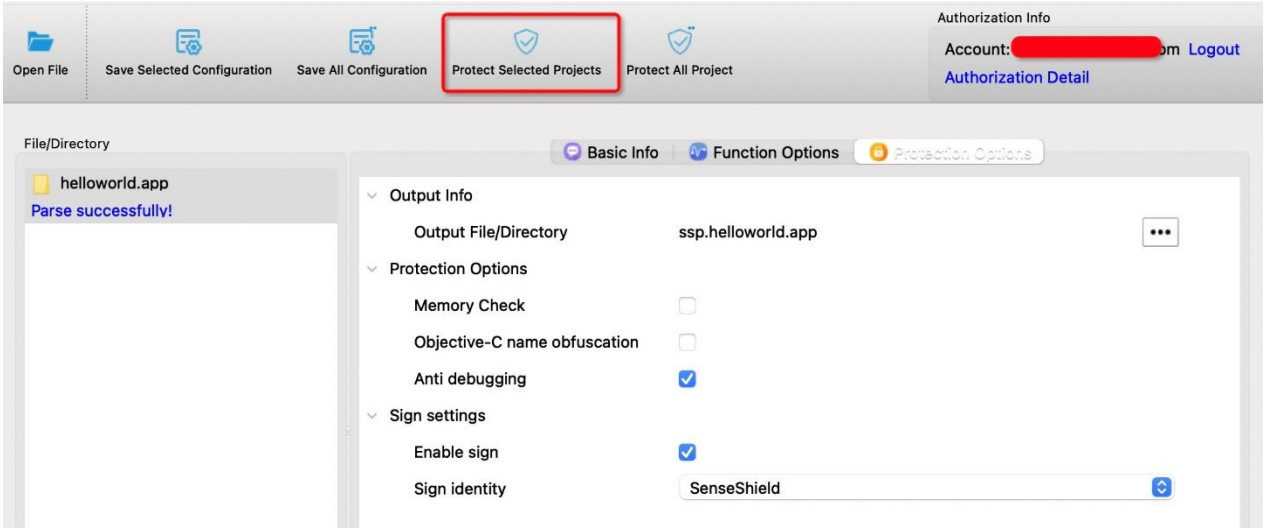


Figure 5-105

5.6.2 Using The Command line tool to protect

1. Use the Virbox Protector GUI tools to Generate the configuration file, if no configuration file has been generated, then the protected App (by Virbox Protector CLI tools) will not signed on default;
2. Open a Terminal in windows, go to the directory of *virboxprotector_con*, input "*virboxprotector_con*" to view the help info;
3. Use Following command to complete the protection:

virboxprotector_con <The iOS App which need to be protected> -o <The Output App>

5.6.3 Note

1. Virbox Protector not support bitcode currently, pls disable the *ENABLE BITCODE* option in "Build Options" process. pls refer the snapshot as shown in below:

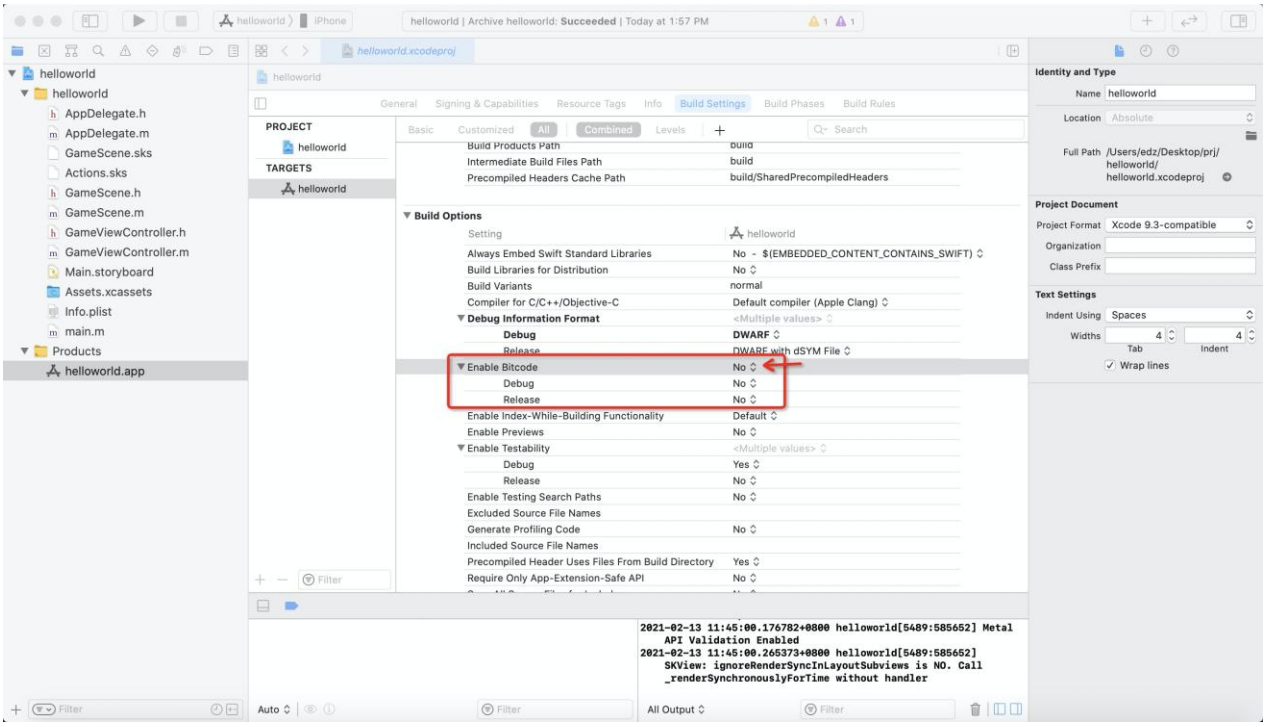


Figure 5-106

2. Currently, Virbox Protector doesn't support the FAT format yet, pls disable this mode in " Build Options " ; and iOS version must set to higher than Release 11.0 in "iOS Deployment Target";

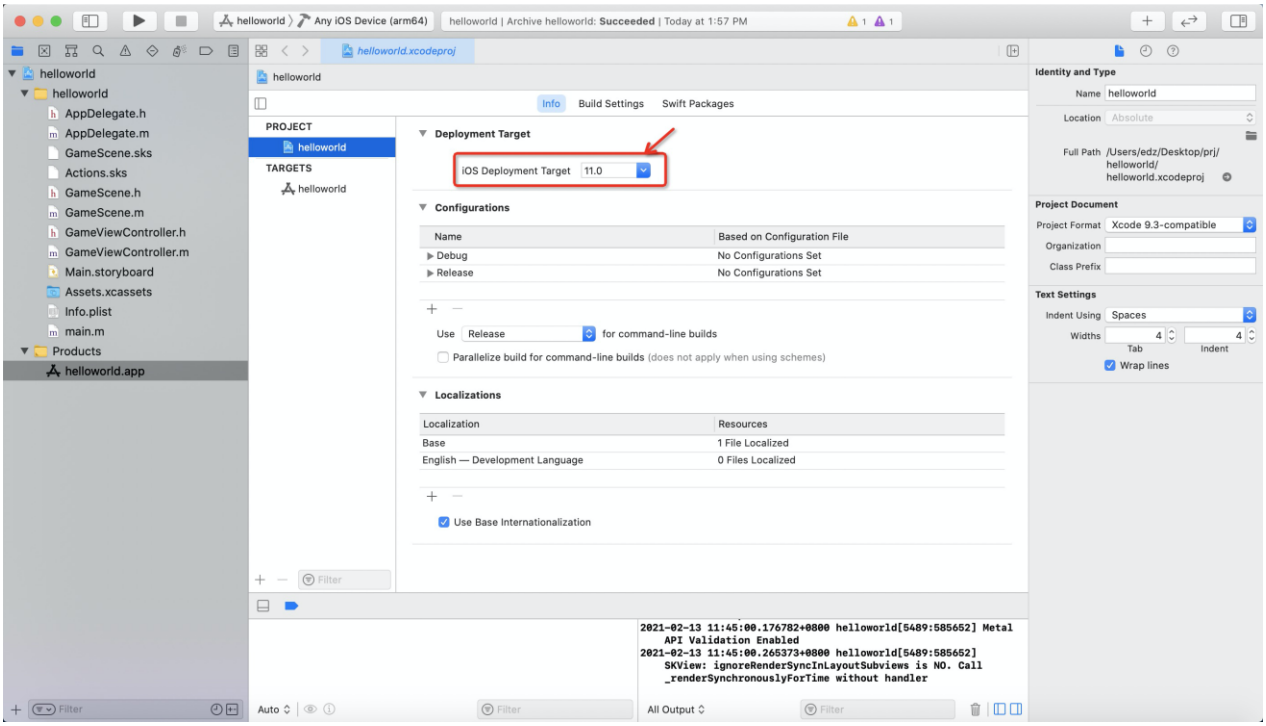


Figure 5-107

3. In "Build Option", if you select the "DWARF with dSYM File" in the "Debug Information Format", pls put the dSYM file into the same directory which your iOS App located. For set the Debug Information Format, pls

refer to following snapshot:

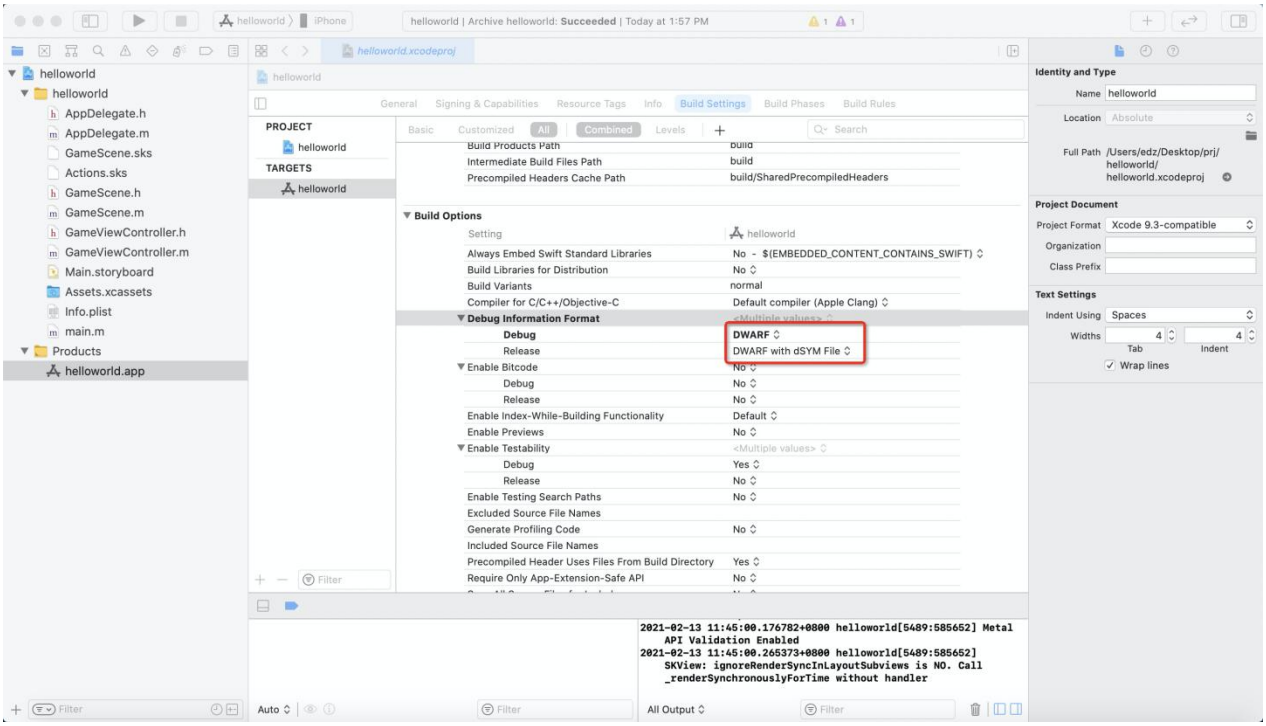


Figure 5-108

5.7 Protect the Python or other Script language based application

There are 2 steps to protect Python or other script language based application:

Use ***Virbox Protector*** to Protect "Interpreter" of Script Language; The protection process is same as normal .exe program; you can protect the python.exe with the protection available in "***Protection Option***" and "***Function Option***" pane;

Use ***DS Protector*** to Protect the Script program itself;

(The User Manual of ***DSProtector*** can be found in the /help sub directory.

5.7.1 Interpreter 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:

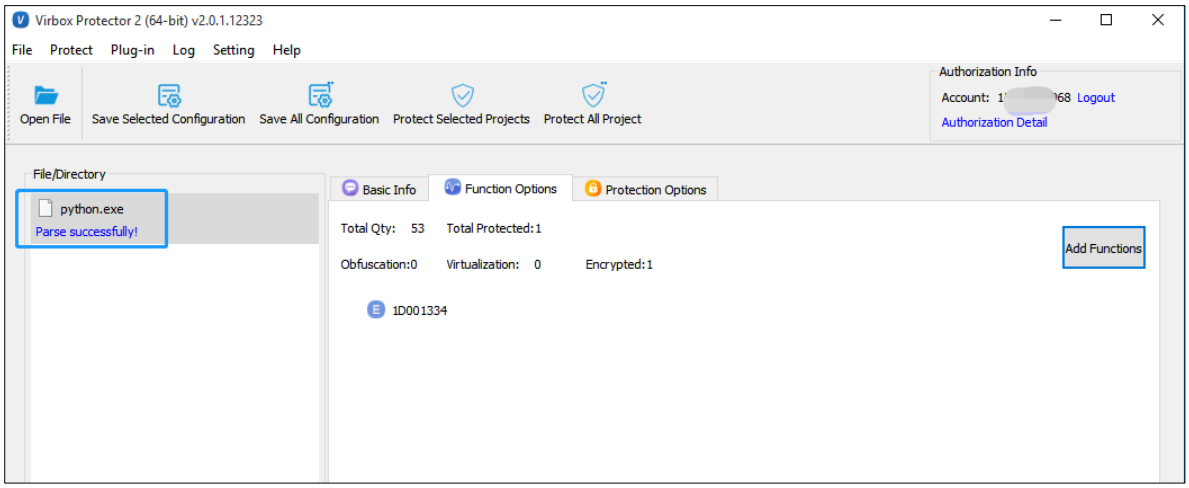


Figure 5-109

Protection Options Setting:

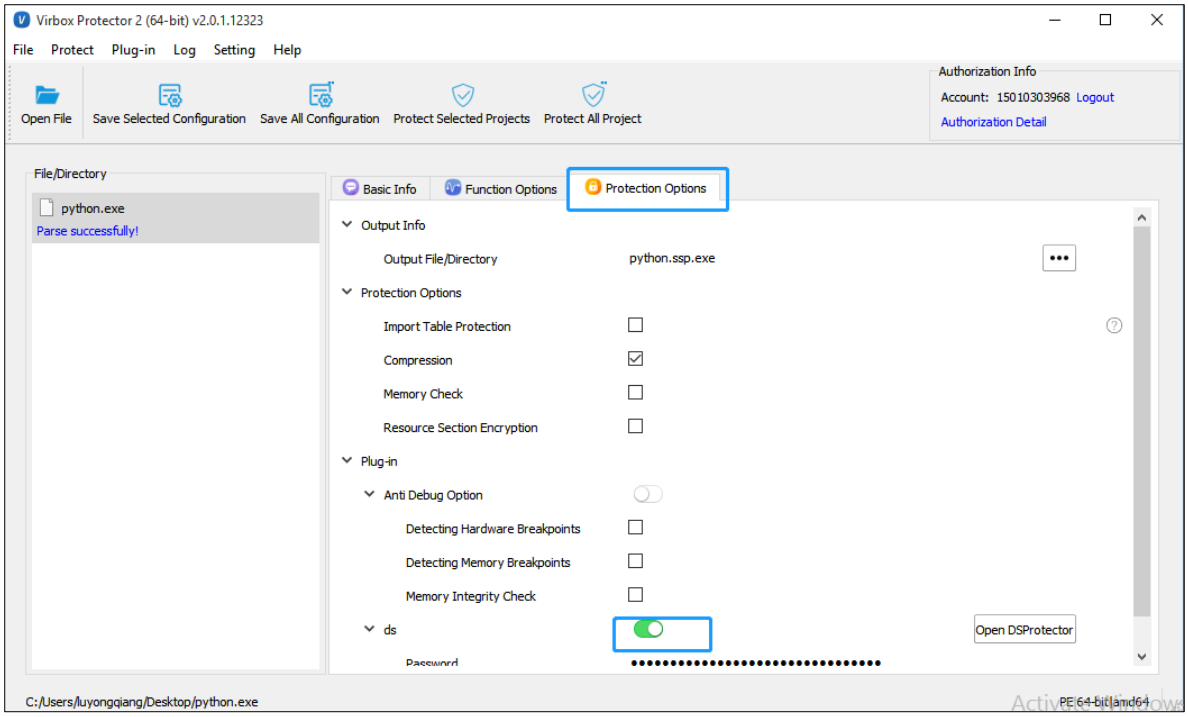


Figure 5-110

Resource Protection

- For resources protection, protect the python.exe with Virbox Protector first, and then protect the **.pyc, pyd** and **.py** file with DSProtector. (for how to use DS Protector, pls refer the User manual of DS Protector)

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 With 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 Without Protection.

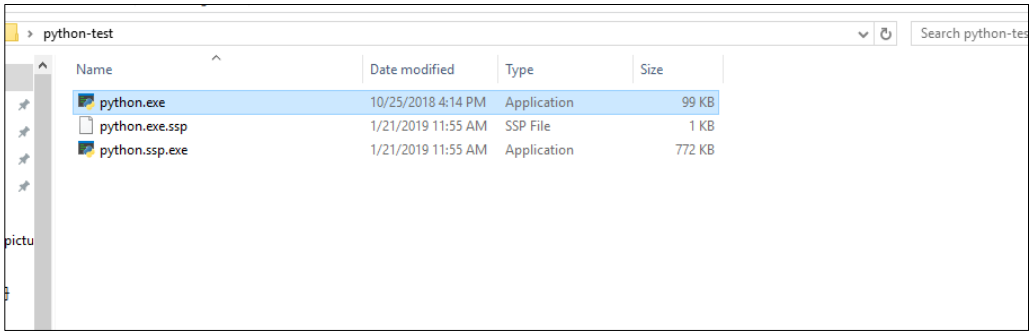


Figure 5-111

5.7.2 Use the DSProtector to protect .pyc and .py file,

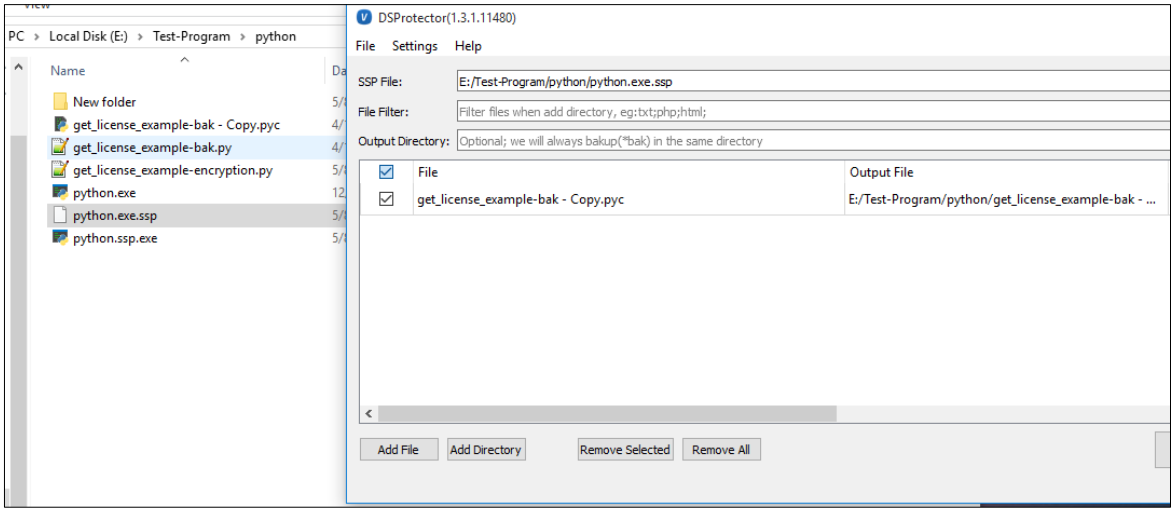


Figure 5-112

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 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 doesn't supported to be protected by Virbox Protector, only Microsoft 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.
		Using COMPRESSION will modify the program type from AnyCPU to be PE32. So when you select the COMPRESSION for .Net AnyCPU program. It would be no longer to be called by other program.
		For the dll of .NET Core 3 in Linux and macOS, it doesn't support COMPRESSION and JIT ENCRYPTION;
		For Virbox Protector LM, it doesn't support to protect the DLL of .NET Core3 in Linux and macOS system;
	PE	RESOURCES PROTECTION doesn't support to the executable file which converted from Powerpoint PPT files.
		RESOURCES PROTECTION doesn't supported for the program which developed based the VB6.0 language
		IMPORT TABLE：The symbol imported must be function and can’t be import variable, otherwise the program would crashed when start to execution.
		If the protected program run with LOAD MEMORY type, it can’t be started when compression selected.
	ELF	Additional data doesn't supported for Linux program currently.
		ELF program which is compiled by –static compile option doesn't supported to be protected by Virbox Protector.
		Map file analysis is not support for 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.
		It doesn't support to Compression for the Linux executable file which convert from the "pyinstall" file;

		Currently, Virbox Protector doesn't support to protect the ELF file which converted from the "GO" language with the "Static Compiling" Support to protect the ELF which compiling with dynamic;
	Unity3D	Virbox Protector, currently, doesn't support to protect the Unity3D program based IL2CPP and mono, which in macOS ARM64 platform;
Protection Option	Code Encryption	The function parsed by Virbox protector without function name may not 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.
		Code snippet protection mode is not available 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 Virbox Protector Command line tool 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 With 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 doesn't supported by Virbox Protector

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.