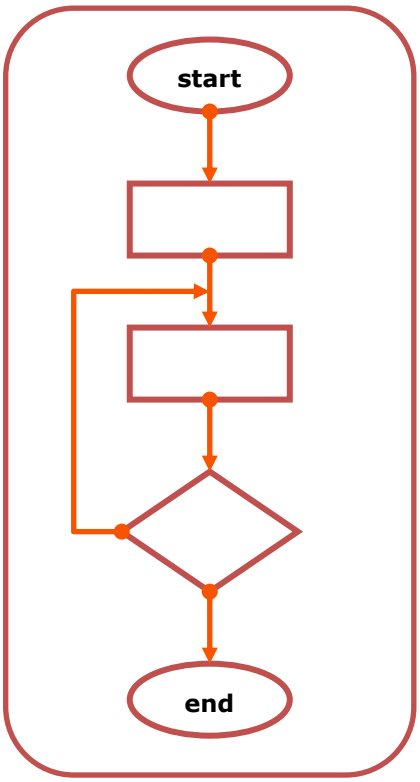
STARFORCE C++ OBFUSCATOR

Service providers report annually about the multi-million dollar losses resulting from fraud and piracy. To avoid it companies pay more attention to software protection. And one of the methods is the deliberate complication of source code to prevent its analysis and modification.

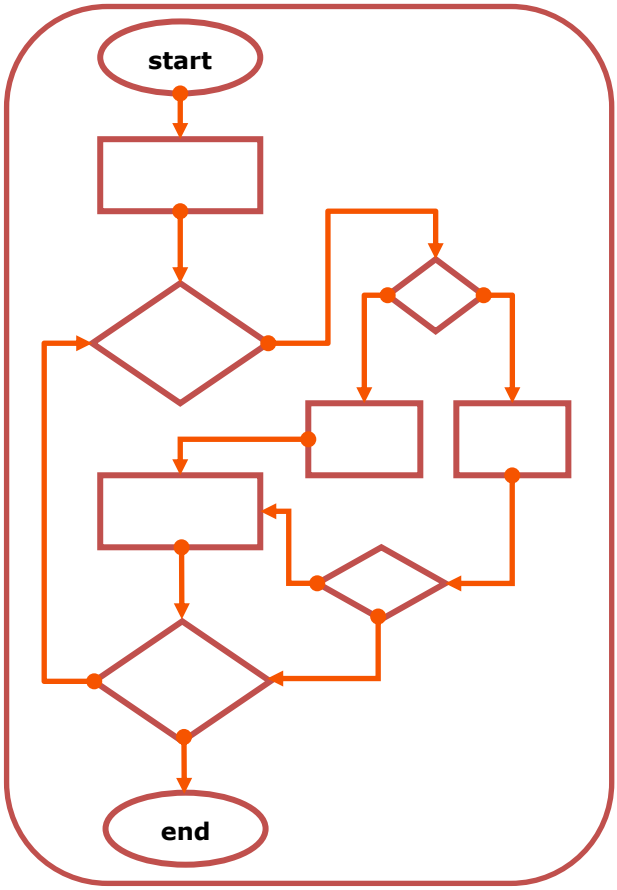
StarForce C++ Obfuscator is efficient solution is designed to obfuscate (transform) the source code of the C/C++programs (text files) to protect them from reverse engineering. After obfuscation the code is reliably protected from analyses that can be performed by a man or by a machine.

HOW IT WORKS

Control flow graph



Obfuscated control flow graph



STARFORCE C++ OBFUSCATOR BENEFITS

- Versatility: it can be applied to protect programs designed for any operating systems (Windows, Mac, Linux, iOS, Android) and any processors, including protection of firmware.
- The high level of protection is provided by obfuscation both the source code and the binary code.
- The obfuscator supports more than 30 obfuscation methods:
  - Conversion of C++ code into virtual machine code
  - Encryption of strings and arrays
  - Code conversion into a state machine
  - Insertion of dummy links
  - Code section merging, etc.
- StarForce C++ Obfuscator is a stand-alone application that is installed at the customer's site.
- Flexible licensing policy (with or without time limitation).

USE

Protection from the loss of standard DRM keys

Situation
Standard DRM systems such as AACS and HDCP are used all over the world to control access to audio and video streams. They provide private keys for each model of the user devices (TV sets, players, consoles, set top boxes). A device key leak leads to pirate content distribution and as a result the discredited key is revoked by the DRM. After that the device stops working and a producer needs to get a new key and pays a lot of money for it. Sometimes, a producer may have a penalty for a device key leak, or its license may be suspended.

Solution
StarForce C++ Obfuscator provides protection for the software that is responsible for video/audio streams delivery and use. After obfuscation the code that is working with DRM keys is reliably protected against analysis and reverse engineering.

Protection of measuring equipment (meters) against analysis and reverse engineering

Situation

The prices for the resources are constantly increasing all over the world. Consequently, consumers do not want to pay the total cost for the resources and more often cheat the service providers. Modern Internet technologies provide readings the data from the meters remotely and sending them to the data processing centers over the network. Dishonest consumers crack the measuring equipment and substitute the consumption data. The losses of the service providers are estimated at billions of US dollars.

Solution

StarForce C++ Obfuscator allows significantly increase the complexity of reverse engineering for measuring equipment software, and thereby reduce the likelihood of tampering.

Protection from cracking of Conditional Access System for satellite and cable TV

Situation

Conditional Access System (CAS) is used in telecommunication industry to limit access to TV channels. It works the following way: TV channels are delivered encrypted and only subscribers who paid the service have a key to decrypt content. Decryption is performed on a user device – set top box (STB). If a hacker could analyze the software and using reverse engineering extract the key he is able to emulate a card (if it is a card CAS) or make a custom firmware to get illegal access to the content. And what is worse he may publish emulated software on the Internet for mass distribution.

Solution

StarForce C++ Obfuscator provides reliable protection against analyses and reverse engineering for firmware that is used in STBs. It helps to decrease the number of pirate connections to satellite and cable service providers.

Protection of a standard DRM client side

Situation

The front end of any DRM system, for example OMA DRM, is vulnerable to the attacks that include reverse engineering, modification and protection disabling. Depending on the DRM, an attack can result in compromising either a certain protected object or all of the protected objects of the user, or all of the protected objects in the system.

Solution

StarForce C++ Obfuscator allows reducing the risk of DRM cracking due to significant increase of the reverse engineering complexity. How the obfuscator should be used (i.e. what should be obfuscated and how) depends on the system's parameters. To get the optimal solution, please, consult StarForce technical team.

Securing the implementation of custom DRM from reverse engineering

Situation

The development of custom DRM is a costly task. If the developer decides to do it, he should in any case solve the problem of securing the DRM code that works on the end user side, from reverse engineering. The developer faces two problems in this case:

1. Man-hours for creating an efficient protection against reverse engineering are significant.
2. There is a risk that protection against reverse engineering may not be efficient enough.

Solution

StarForce C++ Obfuscator enables the developers to save time for the development of a system that protects DRM against reverse engineering. Using StarForce C++ Obfuscator provides a high efficiency level that is known in advance. The developer estimates the level of efficiency when he decides to buy the obfuscator on the basis of the analysis of the obfuscation results.

STARFORCE C++ OBFUSCATOR EASY TO START

StarForce C++ Obfuscator is a stand-alone application protected from copying with the help of StarForce ProActive for Business.

For obfuscation unprotected CPP files are used. In the files the methods with the code that should be protected against reverse engineering are marked with special attributes. The result of the obfuscation is also CPP files but the selected methods in them are obfuscated.

How to use StarForce C++ Obfuscator:

1. Install the obfuscator. No special operations are required. Just unpack the archive with the obfuscator to any folder on a developer's computer.
2. Activate the obfuscator. Run the omniform.exe executable file without any parameters, enter the serial number and activate the product.
3. Perform test obfuscation. To do that please:
    a. Create the test_input.cpp file with the following contents.

```
__attribute__((obfuscate(0))) int f1( int a, int b )
{
      if( a > b ) return a;
      else return b;
}
```

b. Run the obfuscator with the obfrun.exe test_input.cpp test_output.cpp command.
c. Make sure that the test_output.cpp file contains the obfuscated code.

4. Prepare the source files for the obfuscation by adding the obfuscate attribute before the required methods. The simplest way to prepare the files is described above; see user guide for details on how to prepare the files.
5. Customize the obfuscator to work with the required compiler by editing the configuration files.
6. Run the obfuscation using command obfrun.exe.
7. Compile the obfuscated files.
8. Test the compiled application and make sure its functionality is equivalent to the one from non-obfuscated files.

To make the obfuscator easier to use, it should be integrated into the application build process during the development.

FEATURES

The obfuscator can work on the following operating systems:

- Windows XP 32/64-bit.
- Windows Server 2003 32/64-bit.
- Windows Vista 32/64-bit.
- Windows Server 2008 32/64-bit.
- Windows 7 32/64-bit.
- Windows 8 32/64-bit.
- Windows Server 2012 32/64-bit
- Linux (option).

The obfuscator requires the following to operate:

- 350 Mb of disk space for installation.
- 4 GB of RAM.

The obfuscator has the following C++ compatibility parameters:

- Supported compilers:
  - Microsoft Visual C++ 6.0.

- GCC 3.
- GCC 4.
- Support of other compilers compatible with ANSI C++.

- Supported target platforms for C++:
  - Windows (32/64 bit x86).
  - Linux (32/64 bit x86).
  - Android 4.0-4.4 (32-bit ARM, 32-bit x86).
  - Mac.
  - iOS.
- Obfuscation of the following C++ language constructions are not supported:
  - Methods that contain Structured Exception Handling (SEH).
  - Constructors and destructors.
  - Function templates and methods of the template classes.
  - Some other structures that are used less often than those mentioned above. The full list of the restrictions is in the user guide.

ABOUT STARFORCE

StarForce Technologies (www.star-force.com) is a leading vendor of information protection, copy protection and code obfuscation solutions for software, electronic content and audio/video files. Since 2000, StarForce has been successfully developing and implementing its state-of-the-art security solutions, providing copyright and intellectual property protection worldwide. One of these solutions was transformed into a StarForce cloud service (www.sfcontent.com) to protect e-Documents against illegal copying and distribution.

StarForce is a reliable and responsible Technological Partner for enterprises potentially incurring losses due to cyber-gangs, hackers, software piracy, unauthorized data access and information leaks. StarForce's customers are Russian Railways, Corel, 1C, Mail.ru, Aeroflot, SUN InBev Russia, AMD Labs, ATC International, MediaHouse, Russobit M, New Disc, Buka, Snowball, 2Play, GFI, CENEGA, Akella, etc.

For more information please contact us:

StarForce Moscow HQ
Altufevskoe shosse, 5/2
127106 Moscow, Russia
Phone: +7 (495) 9671451
Fax: +7 (495) 9671452
E-mail: sales@star-force.com
www.star-force.com

StarForce France
20, rue Malar
F-75007 PARIS
Phone: +33 (0)1.44.18.37.05
Fax: +33 (0)9.56.72.07.47
E-mail: olivier.duran@star-force.com
www.star-force.com