STAR FORCE®
Your reliable Technological Partner

FrontLine MMOG FAQ

A guide for MMOG developers who are interested in the first version's features

1. Protection against bots and cheating software

1.1 How does the protection against bots and cheating software work? Protection mechanisms

Protection against bots and cheating software is performed indirectly, with the help of the following FL MMOG features:

1. Checking the integrity of the application during its start.
2. Checking the integrity of code and data in memory.
3. Protecting certain game variables against unauthorized access/modification.
4. Protecting against direct recording to RAM, protection of the game processes in RAM against penetration, termination, etc. (see section 1.4).
5. Checking the parent process: it is possible to block the start of the protected game from an unauthorized process.
6. Protecting traffic between client and server.

The technology does not work at such a high level and cannot detect bots and cheating software by itself. StarForce offers just a set of tools (SDK) that can be used by the MMOG developers. These tools include the check of the parent process, the check of the integrity of a storage area, protection against direct recording to RAM, dynamic encryption of data in memory, the check of the server certificate, network traffic encoding, and so on. All these tools operate at a software developer's level.

1.2 How is cheating software detected?

If a call of a check function from the previous section (for example, a function that checks the integrity of a storage area or a variable decoding) results in "check failed", a program error is displayed. A developer can capture the error in his/her code in some cases; in other cases he/she can disable displaying the message during the protection.

1.3 How does the Safe'n'Sec® behavioral analyzer technology work? How does it help protect against bots and cheating software?

StarForce licensed Safe'n'Sec® technology provides developers with a powerful tool to protect RAM against malware intrusion. The technology is based on the limitation of some process privileges that allow changing other processes at an OS level. The technology allows protecting a process in memory against:

- writing to the process address space,
- creating or terminating the process threads,
- forced process termination,

and others.

A developer can work out a system of rules that the behavioral analyzer will follow when blocking access to the game processes in RAM. Besides, there is a possibility to make a White List and a Black List for different programs installed on a user's computer.

Please keep in mind that the Safe'n'Sec® behavioral analyzer uses mini filters to operate. Mini filters are a documented API provided by Microsoft. Mini filters are embedded into the OS driver stack and filter all the system calls.

Safe'n'Sec® SDK is provided to implement user-developed functionality.

The technology supports Windows XP 32/64 bit, Vista 32/64 bit, Windows 7 32/64 bit.

1.4 Does a publisher have to release a new build if a new bot or cheating software appears?

As mentioned in 1.2, the protection works with indirect indicators of a malware presence. Therefore, the update of the threat detection tools is only required when new threats with a completely different character and behavior appear.

With the Safe'n'Sec® behavioral analyzer technology, the developer can make the sets of rules, White Lists and Black Lists and use them to detect and block penetration of malware into the game process memory. Such lists should be updated as new threats appear, while the rules can be corrected.

2. Traffic protection

2.1 How is traffic protection performed?

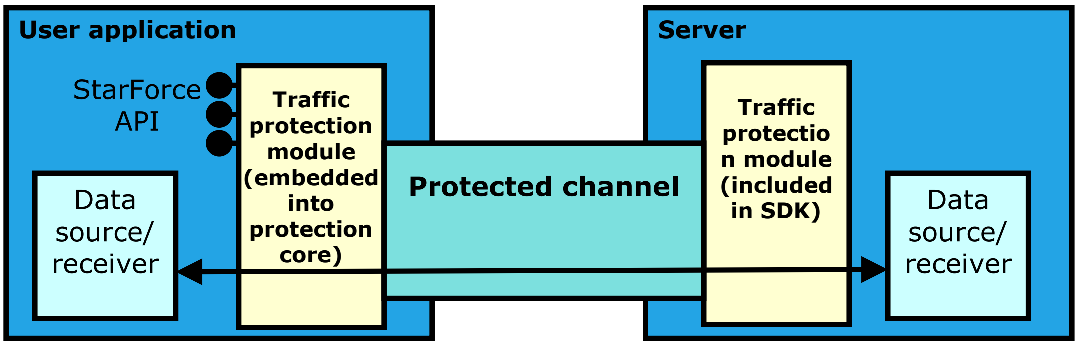
Traffic between the game client and server is encrypted. The main point of protection is that the communication channel between client and server is cryptographically protected.

To transfer data from PS to protection server, a bundle of standard encryption algorithms is used:

- RSA – public key encryption algorithm
- RC4 – symmetric stream cipher algorithm.

This bundle is considered the most secure for the information transfer.

The distribution kit includes a special module that is installed on the client and encrypts the outgoing traffic. Additional decrypting Java module is installed on the server.


User application
StarForce API
Data source/ receiver
Traffic protection module (embedded into protection core)
Protected channel
Server
Traffic protectio n module (included in SDK)
Data source/ receiver

2.2 What is the capacity of the decrypting Java module on the server?

The encrypting and decrypting capacity is not less than 50 Mbit/s for a computer with Core 2 Duo (Processor x86 Family 6 Model 15 Stepping 2 Genuine Intel ~2000 Mhz).

2.3 What are the technical requirements for the Java version on the server and for the server OS?

RSA and RC4 algorithms from javax.crypto should be supported. Testing has been performed on the Java SE 6 platform. There are no special requirements for the server OS; only the proper Java platform is required.

3. RAM protection

3.1. How is the memory control performed, and what parameters are used?

The integrity of read-only code and data in memory is checked. This method is designed to check a protected executable file during the game operation. The integrity of read-only elements of the protected file is checked, i.e. game variables are not checked. The protected game calls the corresponding SF API function for this check.

3.2. What memory modifications do we protect against?

Protection controls the protected variables, as well as read-only code and data.

3.3. How do we resist memory modification?

The technology includes API functions that control the integrity of the marked memory area. That is, the memory areas are compared at different times. If a function returns a value indicating that a memory area has been modified, the integrity check fails. In this case the behavior of the protected program is specified by the developer.

3.4. How do we resist the creation of a new process in memory?

Counteracting the creation of a new memory process, as well as many other protection features are available with the Safe'n'Sec® behavioral analyzer technology (see section 1.4).

3.5. Is there protection against direct recording to memory?

Yes, with the help of the Safe'n'Sec® behavioral analyzer technology (see section 1.4).

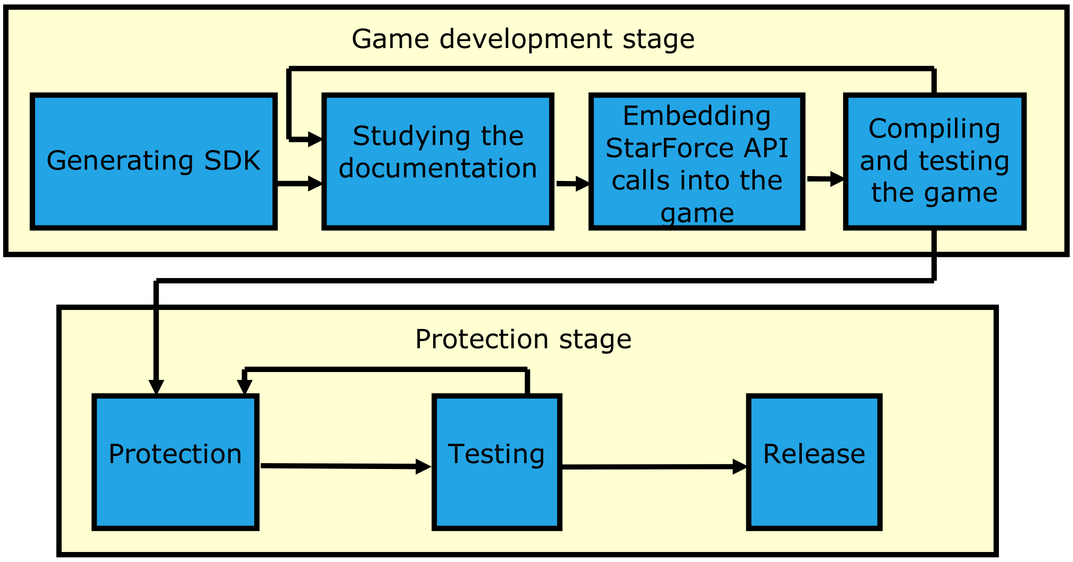
4. The protection process

4.1. How is a game protected?

The protection of files and game data is performed via the Internet with the help of Protection Studio (available for free).

Protection implementation

Client’s actions

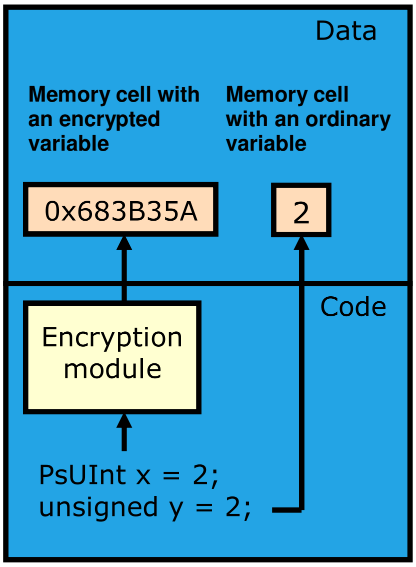

Game development stage
Generating SDK
Studying the documentation
Embedding StarForce API calls into the game
Compiling and testing the game
Protection stage
Protection
Testing
Release

Copyright © Protection Technology, 2008

www.star-force.com

Embedding StarForce API calls into the game at the source code level is mandatory.

4.2. How is modifiable data protected?


Data
Memory cell with an encrypted variable
Memory cell with an ordinary variable
0x683B35A
2
Code
Encryption module
PsUInt x = 2;
unsigned y = 2;

- Protection is performed by encrypting the variables
- 32-bit integer variables are supported
- C++ and C# are supported
- Using SDK is necessary.

4.3. How long does it take to protect a game several gigabytes in size?

It depends on the number of the modules (executable files and libraries) and the application functions to be protected. In whole, the total size of the distribution kit does not influence the protection time in any way.

The protection process on server usually lasts from 20 min to 1 hour. Besides, a user who implements protection for the first time needs some time to read the documentation and understand the process.

4.4. How is the code to be placed to VM marked?

A function code is declared in a certain way, and then the function is selected in Protection Studio. For details see FL MMOG User Manual.

The function is not moved to the virtual machine completely during protection; rather its part – basic blocks – is moved.

Declaring exported functions (example):

To declare a function as exported, special code should be added to the source project.

- When programming on C/C++, the `__declspec(dllexport)` keyword is used.
  Example:

`__declspec(dllexport)` **<function type> <function name>**

`__declspec(dllexport) void InitAVICodec(void);` – for a callback function;

`__declspec(dllexport) int RunMasterMenu(CMenu &, int );` – for a loopback function.

- When programming on PASCAL, the `exports` keyword is used.
  Example:

```
exports
    YourFirstExportedFunction name 'YourFirstExportedFunction';
    YourSecondExportedFunction name 'YourSecondExportedFunction';
    YourThirdExportedFunction name 'YourThirdExportedFunction';
```

4.5. How can we configure the VM options?

Client virtual machine is not configured. There is only one option for a function being protected: function execution speed.

The execution speed depends on the level of the function protection. For example, strong protection of functions that are called frequently is not recommended, since it would slow down the application.

The execution speed can be specified separately for each function.

5. Protection of game resources and data files

5.1. How are non-executable files (data files) of the game protected? How does it influence the performance?

The integrity of read-only files is checked. The protected game checks digital signatures of the data files. Thus, this complicates reverse engineering and modification of the protected game files.

It has no significant influence on the overall game performance.

Data files can be protected with the help of a ‘file container’. There are two ways to implement a container:

- Using driver.
  Advantages: There is no need to modify the application code; using Protection Studio is sufficient. It is not necessary to use StarForce API and modify the source code in this case.
  Disadvantages: protection driver is required.

- Using StarForce API.
  Advantages: Higher reliability, since it is more difficult to intercept readable data. (The protection is especially good when a function that reads data is also protected.) No driver is required.
  Disadvantages: Source code modification is necessary.

5.2. Does protection of the game data files complicate patch and update release?

When updates are released, they should be protected as usual. The size of the updates increases by several megabytes because of the protection library.

6. Protection driver

6.1. Is the protection driver used?

Protection driver is not used in the FL MMOG scheme by default. However, it can be used to access files in a file container when using file container with driver.

Besides, to implement reliable memory protection, mini filters from the Safe'n'Sec® behavioral analyzer are used. Mini filter is a documented API provided by Microsoft. A user should have administrator rights to install and work with mini filter.

6.2. If drivers are used, how many of them are installed?

One driver to access the protection containers and one mini filter when using the behaviour analyzer technology.

6.3. If the protection driver is used, can it be deleted without the game deletion?

- Driver can be easily deleted using the protection functionality.
- If the driver has been deleted, the protected application suggests driver installation at the first run.
- There is a driver auto deletion procedure. Special service checks the presence of the game. If the game is deleted, the service deletes driver and itself.

6.4. How does the protection driver influence the system when the protected game is not running?

One can say it has no influence. Driver is inactive until the game runs. After the game runs, driver is loaded to memory and uses only several kilobytes. After the game terminates, driver remains in the memory until the computer shuts down. Driver uses several megabytes on the hard disk.