

AhnLab HackShield

Programming Guide

Version 5.18

AhnLab HackShield for Online Game

Preface

Copyright (C) AhnLab, Inc. 2002-2008. All rights reserved.

Contents of this document and the related software programs are protected by the Copyrights Act and the Computer Program Protection Act.

Document Overview

This programming guide describes the overall structure and API features of AhnLab HackShield for Online Game and AhnLab HackShield Pro for Online Game (hereinafter to be referred to as HackShield).

Customer Support

AhnLab Customer Center	
Address	AhnLab, 673, Sampyeong-dong, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Homepage	http://www.ahnlab.com
Email	hs@ahnlab.com
Telephone	031-772-8250 (Hotline for corporate clients)
Fax	031-722-8901 (Representative)

Table of Contents

1. Introduction	9
1.1. Functions.....	9
1.2. System Environment	10
2. Basic Features	11
2.1. Overview	11
<i>Functions</i>	11
<i>Features</i>	12
<i>System Architecture</i>	13
2.2. Application Programming	15
<i>Programming Procedure</i>	15
2.3. Preparation.....	18
2.3.1. HackShield Files	18
2.3.2. Application	19
<i>Write HackShield update function</i>	19
<i>Write HackShield monitoring function</i>	20
<i>Issuing License Key</i>	20
<i>Write HackShield initialization function</i>	21
<i>Write HackShield service start function</i>	23
<i>Write HackShield callback function</i>	23
<i>Write HackShield service stop function</i>	25
<i>Write HackShield service complete function</i>	25
<i>Write HackShield logging function</i>	27
2.4. Application Programming Interface	29
_AhnHS_Initialize	29
_AhnHS_Callback	41
_AhnHS_StartService	47
_AhnHS_StopService	51
_AhnHS_Uninitialize	53
_AhnHS_PauseService.....	55
_AhnHS_ResumeService	57
_AhnHS_CheckHackShieldRunningStatus	59
_AhnHS_SendHsLog	61
_AhnHS_VerifyProtectedFunction	62
_AhnHS_QueryPerformanceCounter	64
_AhnHS_QueryPerformanceFrequency	66
_AhnHS_GetTickCount.....	68
3. HackShield Update	70
3.1. Overview	70
<i>Functions</i>	70
<i>Features</i>	70
<i>System Architecture</i>	71
3.2. Application Programming	72
<i>Programming Application</i>	72
3.3. Application Programming Interface	81
_AhnHS_HSUpdateEx	81
_AhnHS_HSUpdate	86
4. Extended Server-side Detection	90

4.1. Overview	90
<i>Functions</i>	90
<i>Features</i>	91
<i>Extended Server-side Detection Files</i>	91
<i>System Architecture</i>	92
4.2. Application Programming	96
<i>Programming Application</i>	96
4.3. Application Programming Interface	99
_AhnHS_CreateServerObject	99
_AhnHS_CloseServerHandle	101
_AhnHS_CreateClientObject	102
_AhnHS_CloseClientHandle	104
_AhnHS_MakeRequest	105
_AhnHS_VerifyResponseEx	108
_AhnHS_VerifyResponseEx_WithInfo	110
_AhnHS_VerifyResponse	113
_AhnHS_MakeResponse	119
5. Monitoring Service	123
5.1. Overview	123
<i>Functions</i>	123
<i>Features</i>	123
<i>System Architecture</i>	124
5.2. Application Programming	125
<i>Programming Application</i>	125
5.3. Application Programming Interface	128
_AhnHS_StartMonitor	128
_AhnHS_SetUserId	130
_AhnHS_SetUserCustomInfo	131
_AhnHS_SendUserCustomInfo	133
6. LMP	136
6.1. Overview	136
<i>Functions</i>	136
<i>Features</i>	137
<i>System Architecture</i>	137
6.2. Application Programming	139
<i>Programming Application</i>	139
_AhnHS_IsModuleSecure	141
_AhnHS_IsModuleSecure	142
7. Other features	145
7.1. Data file/message encryption	145
7.1.1. Overview	145
<i>Functions</i>	145
<i>Features</i>	145
<i>System Architecture</i>	145
7.1.2. Application Programming	146
<i>Programming Procedure</i>	146
<i>Preparation</i>	147
<i>HsCryptLib File</i>	147
_HsCrypt_InitCrypt	148
_HsCrypt_GetEncMsg	148
_HsCrypt_GetDecMsg	149
HsCrypt_FRead	149

7.1.3. Application Programming Interface.....	150
_HsCrypt_InitCrypt.....	150
_HsCrypt_GetEncMsg	152
_HsCrypt_GetDecMsg	153
HsCrypt_FRead	154
7.2. User Rights Support.....	157
7.2.1. Overview	157
<i>Functions</i>	157
<i>Features</i>	157
<i>System Architecture</i>	157
7.2.2. Application Programming.....	158
<i>Programming Procedure</i>	158
<i>Preparation</i>	160
_AhnHsUserUtil_CreateUser	160
_AhnHsUserUtil_SetFolderPermission	161
_AhnHsUserUtil_DeleteUser.....	161
_AhnHsUserUtil_IsEnableHSAdminRights	162
_AhnHsUserUtil_CheckHSShadowAccount.....	162
_AhnHSUserUtil_IsAdmin	163
7.2.3. Application Programming Interface.....	163
_AhnHsUserUtil_CreateUser	163
_AhnHsUserUtil_SetFolderPermission	166
_AhnHsUserUtil_DeleteUser	167
_AhnHsUserUtil_IsEnableHSAdminRights	168
_AhnHsUserUtil_CheckHSShadowAccount	169
_AhnHSUserUtil_IsAdmin	171
8. Tools.....	172
8.1. HSBGen Tool (For HackShield 4.2 or later)	172
<i>Functions</i>	172
<i>System Environment</i>	172
Using HSBGen Tool	173
<i>Creating HSB Information File Based on Manual UI Setting</i>	173
<i>Automatically Creating HSB Information File</i>	175
<i>Check LMP Data</i>	178
<i>HSBGen.ini description</i>	178
8.2. HSUpSetEnv Tool (For HackShield 5.1 or later)	183
<i>Functions</i>	183
Using HSUpSetEnv Tool	184
<i>Creating HSUpdate.env File</i>	184
8.3. CSInspector Tool	187
<i>Functions</i>	187
<i>Setting the File Subject to Protection</i>	187
8.4. SetServerList Tool (For HackShield 5.1 or later).....	188
<i>Functions</i>	188
8.5. Using SetServerList Tool.....	189
<i>afs.dat File Creation</i>	189
<i>afs.dat File Distribution</i>	190
8.6. Using HSBHelper Tool.....	192
<i>Functions</i>	192
<i>Run Command Line</i>	193
9. Appendix.....	194
9.1. FAQ	194
9.2. Index.....	198

9.3. Revisions.....	200
---------------------	-----

List of Tables

Table 2-1 HackShield File.....	18
Table 2-2 Files that need to be installed on game update server	19
Table 2-3 [Callback code when applying AHNHS_CHKOPT_SELF_DESTRUCTION]	33
Table 3-1 update-related File.....	73
Table 3-2 Files that need to be installed on update server	73
Table 3-3 Automatically created files after HackShield update	74
Table 3-4 HackShield update server specification	75
Table 4-1 Server interaction version management	93
Table 4-2 AntiCpXSvr-related File.....	96
Table 5-1 Monitoring-related Files.....	125
Table 6-1 LMP-related Files	139
Table 6-2 Packets that LMP Supports.....	143
Table 7-1 HsCryptLib File	147
Table 7-2 HsUserUtil File.....	160

List of Figures

Fig. 2-1 HackShield's System Architecture	13
Fig. 2-2 Application Programming Sequence	16
Fig. 3-1 HackShield update	72
Fig 3-2 Default HackShield Update Image	80
Fig. 4-1 AntiCpX operation	94
Fig. 5-1 Monitoring service	124
Fig. 6-1 Local Memory Protection.....	138
Fig. 7-1 HsCryptLib structure and operation.....	146
Fig. 7-2 General Architecture and Operating Principles of HsUserUtil	158
Fig. 7-3 HsUserUtil programming order.....	159
Fig. 8-1 HSB File Generator	174
Fig. 8-2 Command-line HSBGen.exe ([Packing:1],[Execution:1], [EXE:1], [LMP:1])	176
Fig. 8-3 Command-line HSBGen.exe ([Packing:1],[Execution:1], [EXE:1], [LMP:0])	177
Fig. 8-4 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:1], [LMP:1])	177
Fig. 8-5 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:1], [LMP:0])	177
Fig. 8-6 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:1], [LMP:1])	177
Fig. 8-7 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:1], [LMP:0])	177
Fig. 8-8 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:0], [LMP:1])	178
Fig. 8-9 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:0], [LMP:1])	178
Fig. 8-10 HSUpSetEnv.exe Basic Information tab	184
Fig. 8-11 HSUpSetEnv.exe Extension Information tab	185
Fig. 8-12 CSInspector.exe	187
Fig. 8-13 SetServerList.exe	189
Fig. 8-14 SetServerList tool's address field	190
Fig. 8-15 Tool.....	192

1. Introduction

HackShield is AhnLab's security solution designed to detect hacking tools, block hacking attacks, prevent cracks, protect executable files in real time, and encrypt data.

1.1. Functions

Hack prevention

Detects hacking tools based on signatures and memory heuristics, and blocks memory hack, speed hack, debugging, message hooking, file manipulation, and auto-mouse clicking.

Server-side detection

Detects executable file manipulation and memory execution, and checks the operational status of HackShield by interfacing with the servers.

Data file/message encryption

Encrypts important data files and messages exchanged between the server and the client in order to protect them from unauthorized users.

User authority execution

Allows users to execute the game using user authority or guest authority in NT series.

1.2. System Environment

The following system environment is required for installation and operation of HackShield.

Client Side

Category	Recommended specifications	Minimum requirements
Operating system	Windows 98/ME/2000 Professional/XP Home /XP Professional/Server 2003/Vista/Windows7	Windows 98 or higher
CPU	Intel Pentium 500Mhz or higher	Intel Pentium 133MHz or higher IBM-PC compatible
RAM	128MB or more	32MB
HDD	2MB or more	2MB

Server Side (Extended server-side detection)

OS	Platform
Windows 2K, 2K3, XP, VISTA, 2008	x86, x64
Solaris 8, 10 (32bit)	x86
Fedora Linux 7.1(x86), 11(x64), redhat enterprise 4, cent OS 5.3, FreeBSD 7.x(x86)	x86, x64

Caution

When operating HackShield in Windows NT series server such as a web server or a DB server, unexpected performance compromise may occur.

Note

Only Intel x86 and x64 series (including x86 compatible AMD) CPUs are supported. Alpha chip machine running on the Windows NT and NEC pc 8xxx machine OS running on the Windows are not supported.

2. Basic Features

This chapter describes the system structure and programming method required for implementing hacking prevention and hacking tool detection features.

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

2.1. Overview

Functions

Engine-based (signature and heuristic) detection of hacking tools

AhnLab signatures that are registered in the engine detect hacking tools. If the hacking tool is registered in the engine, the corresponding process will be forcibly terminated (depending on the option) and the corresponding file name will be notified to the game client by the callback function. If a hacking tool that is not registered in the engine is detected, the signature will be updated in order to block the hacking tool.

Memory-access block

Blocks memory access through Windows API (OpenProcess, Read/WriteProcessMemory and etc.). HackShield directly traces the memory at the kernel level and blocks the hacking attack which may manipulate the result data.

Caution

If a program, not a hacking tool, directly accesses the memory, the program may not operate properly.

Speed-hack block

Speed hack usually manipulates the timer installed in the system (hardware) or

time-related APIs in the OS (software). In order to prevent such speed hack attacks, HackShield regularly compares the system time in the microprocessor level and the logical time data created by the OS. If there is difference between the system time and OS time, the callback function will notify it to the game client.

Depending on the user's system, OS, and characteristics of the game, the level of speed hack detection may differ. You can set the speed hack detection level in five stages using parameters.

Note

Speed hack refers to a program that manipulates the timer or time-related features provided by the Windows system.

Debugging block

HackShield analyzes the game program using a debugger and blocks all debugger tracing in order to prevent hacking attacks. When a debugger tracing attempt is detected, the callback function will notify it to the game client. When HackShield is initialized, it will first check whether a debugging program such as SoftICE is running. If a debugging program is running, an error will be returned.

Auto mouse block

HackShield blocks auto mouse in order to prevent manipulation of the game program and overload of the game server.

Features

Interface Function (API)

HackShield provides interface DLL for you to use the HackShield features and view the result data. You can select only the hack prevention features you need, using the provided interface DLL.

Sample programs

HackShield provides a test game client program implemented by using the API provided by HackShield. You can check the features and performance of HackShield and easily develop client programs by referring to the sample program.

HackShield updates

HackShield update provides hack prevention features and hacking tool detection engine. The game client can download the latest hack prevention features and hacking tool detection engine through the update server. The update server supports both FTP and HTTP. The FTP supports anonymous login and user login. When update is made, you can check the update progress by setting options.

System Architecture

HackShield is provided in the form of a library that uses SDK. It is not provided in the form of an executable file. The general architecture and operating principles of HackShield are as follows:

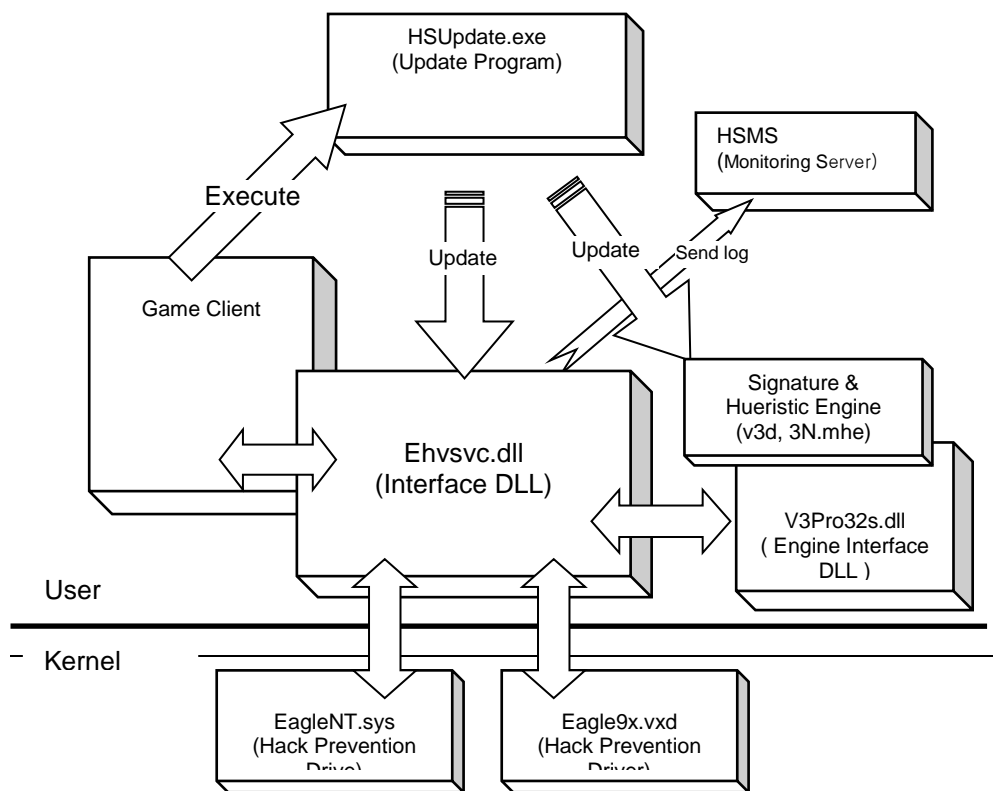


Fig. 2-1 HackShield's System Architecture

HSUpdate.exe(Update Program)

HackShield update program which receives the hacking prevention module and the hacking tool detection module from the update server.

EHSvc.dll(Interface DLL)

Interface file that operates the hacking prevention module and the hacking tool

detection engine. Results on detected hacking tools and blocked hacking attempts are notified to the game client through the callback function.

V3Pro32s.dll(Engine Interface DLL)

Starts the hacking tool detection engine with AhnLab's anti-virus technologies. When a hacking tool registered in the engine signature file runs, this will immediately detected and notified to the game process. If a hacking tool not registered in the engine signature file is detected, the hacking tool will be immediately registered in the engine signature file and the engine signature file will be updated.

EagleX9x.vxd, EagleNT.sys or EagleX64.sys(Anti-Hacking Driver)

Driver file operating on the kernel mode that blocks hacking attempts. It is included in EHSvc.dll (interface DLL) file. EagleX9x.vxd, EagleNT.sys, or EagleX64.sys is loaded to the system depending on the OS.

2.2. Application Programming

This chapter describes how to implement the hack prevention and hacking tool detection features using the APIs provided by HackShield.

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

Programming Procedure

The game client developer can implement the hack prevention and hacking tool detection features in the following sequence:

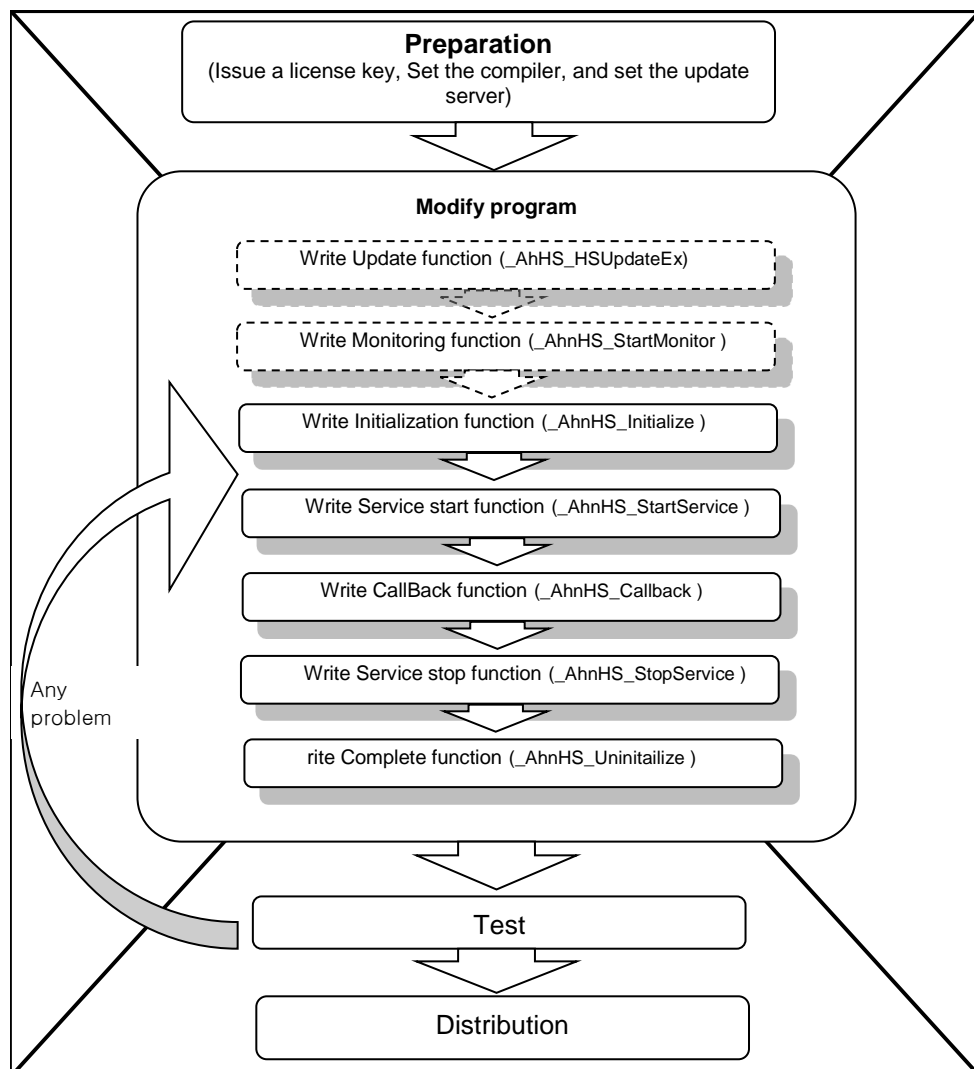


Fig. 2-2 Application Programming Sequence

1. **Preparation:** Check the list of the provided HackShield files and copy the necessary files. In order to start programming, you need a license key.
2. **Write HackShield update function (AhHS_HSUpdateEx):** If you are using HackShield Update, write the HackShield update function to automatically update the HackShield module.

Note

When updating the engine using the HackShield update program, call the update function using the provided update library (HSUpChk.lib).

3. **Write HackShield monitoring function (AhnHS_StartMonitor):** If HackShield Monitoring Server (HSMS) is installed, write the function to send hack and error information to the server.
4. **Write HackShield initialization function (AhnHS_Initialize):** Write a code to

call the HackShield initialization function to check the file manipulation status and initialize the data.

5. **Write HackShield service start function (AhnHS_StartService):** Write a code to call the service start function to block hacking attacks and detect hacking tools.
6. **Write HackShield callback function (AhnHS_Callback):** Write a callback function to block hacking attacks and detect hacking tools.
7. **Write HackShield service stop function (AhnHS_StopService):** Add stop function call in part to terminate the program to stop the hack prevention and hacking tool detection features.
8. **Write HackShield complete function (AhnHS_Uninitialize):** Write a code to call the service stopping function and complete function.

Caution

Even if a game ends abnormally, the HackShield termination routine must be executed.

For abnormal termination, add the following code at the beginning of the game to execute the HackShield termination routine when the game is terminated.

```
void Game_UnhandledExceptionHandler ()
{
    _AhnHS_StopService();
    _AhnHS_Uninitialize();
}

::SetUnhandledExceptionFilter ( Game_UnhandledExceptionHandler );
```

9. Test whether the written source code normally operates.
10. Distribute the game client to the users.

2.3. Preparation

This chapter provides the list of HackShield files to be checked under the installation directory before the programming process, and describes how to set the compiler, use the update module, set the update server, and issue license keys.

2.3.1. HackShield Files

Installation Directory

Create HackShield folder under the game client folder and store HackShield files under the HackShield folder.

<Game Directory>\HShield\

HackShield Files

Table 2-1 HackShield File

File name	Installation folder	Description
3n.mhe	[HackShield Folder]	Heuristic engine file
EhSvc.dll	[HackShield Folder]	Interface DLL
hshield.dat	[HackShield Folder]	HackShield dat file
v3pro32s.dll	[HackShield Folder]	Hacking tool detection engine interface DLL
asc_com.dll	[asc folder]	Hacking tool detection engine interface DLL
asc_dh.dll	[asc folder]	Hacking tool detection engine interface DLL
asc_fse.dll	[asc folder]	Hacking tool detection engine interface DLL
asc_intg.dll	[asc folder]	Hacking tool detection engine interface DLL
asc_mmgr.dll	[asc folder]	Hacking tool detection engine interface DLL
asc_unp.dll	[asc folder]	Hacking tool detection engine interface DLL
fse_base.dll	[asc folder]	Hacking tool detection engine interface DLL
fse_fact.dll	[asc folder]	Hacking tool detection engine interface DLL
fse_pe.dll	[asc folder]	Hacking tool detection engine interface DLL

gfs_base.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_fact.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_file.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_mem.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_os.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_proc.dll	[asc folder]	Hacking tool detection engine interface DLL
gfs_util.dll	[asc folder]	Hacking tool detection engine interface DLL
0asc.scd	[asc folder]	Hack prevention engine pattern file
0scure.scd	[asc folder]	Hack prevention engine pattern file
0sgame.scd	[asc folder]	Hack prevention engine pattern file
0spe3f.scd	[asc folder]	Hack prevention engine pattern file
moduler.scd	[asc folder]	Hack prevention engine pattern file
option.scd	[asc folder]	Hack prevention engine pattern file
AhnRpt.exe	[HackShield Folder]	HackShield logging file
HsLogMgr.exe	[HackShield Folder]	HackShield logging selection file
AhnRpt.ini	[HackShield Folder]	HackShield logging information file
HShield.h	[Game source folder]	Header file
HShield.lib	[Game source folder]	Library file

Compiler Setting

The project file of the game client program where HackShield is installed must include HShield.lib file in the library or the source code list.

2.3.2. Application

Write HackShield update function

[Case1. Using HackShield update module]

Refer to [3. Update](#) to set the update server and apply the client.

When applying the client, you must call the update library function to update HackShield before loading the EhSvc.dll file, which is a HackShield interface DLL.

[Case2. Using own patch module]

When updating HackShield files using your own patch module, you do not need to distribute a separate update module. You only need to distribute the existing game patch module and HackShield files.

When using your own patch server, the following HackShield files must be installed:

Table 2-2 Files that need to be installed on game update server

File name	Description
3n.mhe	Heuristic engine file
Ehsvc.dll	HackShield interface DLL
hshield.dat	HackShield dat file
V3pro32s.dll	Hacking tool detection engine interface DLL
0asc.scd	Hacking tool pattern engine file
0scure.scd	Hacking tool pattern engine file
0sgame.scd	Hacking tool pattern engine file
0spe3f.scd	Hacking tool pattern engine file
asc_com.dll	Hacking tool detection engine interface DLL
asc_dh.dll	Hacking tool detection engine interface DLL
asc_fse.dll	Hacking tool detection engine interface DLL
asc_intg.dll	Hacking tool detection engine interface DLL
asc_mmgr.dll	Hacking tool detection engine interface DLL
asc_unp.dll	Hacking tool detection engine interface DLL
fse_base.dll	Hacking tool detection engine interface DLL
fse_fact.dll	Hacking tool detection engine interface DLL
fse_pe.dll	Hacking tool detection engine interface DLL
gfs_base.dll	Hacking tool detection engine interface DLL
gfs_fact.dll	Hacking tool detection engine interface DLL
gfs_file.dll	Hacking tool detection engine interface DLL
gfs_mem.dll	Hacking tool detection engine interface DLL
gfs_os.dll	Hacking tool detection engine interface DLL
gfs_proc.dll	Hacking tool detection engine interface DLL
gfs_util.dll	Hacking tool detection engine interface DLL
moduler.scd	Hacking tool pattern engine file
option.scd	Hacking tool pattern engine file

Write HackShield monitoring function

The monitoring server must be installed and operated through the AhnLab_HSMS_Install_Guide and AhnLab_HSMS_Operator_Guide.

Refer to [4. Monitoring Service](#) to apply the monitoring feature.

Issuing License Key

You need a license issued from AhnLab to apply HackShield.

You can get a license key issued as follows:

1. Send the name of the executable file that uses EhSvc.dll, the publisher's name (region), game developer's name, and game program name to AhnLab and request a license key.
2. A unique 4-digit game code and 24-digit character string license key will be issued.
3. Send the issued game code and the license key as the parameters of the Initialization function, `_AhnHS_Initialize`.

Caution

If you send the wrong value as the parameter of the initialization function, `_AhnHS_Initialize`, the initialization function will not be called properly, and the error (`HS_ERR_INVALID_LICENSE`) will be returned.

Write HackShield initialization function

After the preparation is complete for programming, call the initialization function, `_AhnHS_Initialize`. The hack prevention and hacking tool detection features can be executed only when the initialization function is successfully called.

The routine that initializes the game client program instance or the main window calls the initialization function.

An example of initialization function is as below. In the following example, the last parameter of the initialization function is `AHNHS_CHKOPT_ALL` so as to use all options .

(There is a core option1 which is automatically applied even when not selected.)
The core option is automatically applied even when it is not selected by the user.)

Example

```
BOOL HS_Init()
{
    int          nRet = 0;
    TCHAR szFullPath[MAX_PATH];
    TCHAR szMsg[MAX_PATH];
    DWORD       dwOption = 0;

    // ① Define location of EhSvc.dll in HackShield folder.
    lstrcat ( szFullPath, _T( "\\HShield\\EhSvc.dll" ) );

    // ② Define the option flag to use to call _AhnHS_Initialize function
    dwOption = AHNHS_CHKOPT_ALL;

    // ③ Call _AhnHS_Initialize function to initialize HackShield service.

    nRet = _AhnHS_Initialize ( szFullPath,
                              HS_CallbackProc, // callback function
                              1000,             // game code
                              "B228F291B7D7FAD361D7A4B7", // License key
                              dwOption,        // option flag
                              AHNHS_SPEEDHACK_SENSING_RATIO_NORMAL );

    // ④ Check the return value of _AhnHS_Initialize function and create an error.
    if ( nRet != HS_ERR_OK )
```

```

    {
        switch ( nRet )
        {
            case HS_ERR_COMPATIBILITY_MODE_RUNNING:
            case HS_ERR_NEED_ADMIN_RIGHTS:
            case HS_ERR_INVALID_FILES:
            case HS_ERR_INIT_DRV_FAILED:
            case HS_ERR_DEBUGGER_DETECT:
            case HS_ERR_NOT_INITIALIZED:
            default:
                wsprintf( szMsg, "A problem occurred in the hack prevention
                             feature.(%x)", nRet );

                break;
            }
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            return FALSE;
        }
        return TRUE;
    }
}

```

Caution

AhnHS_Initialize function can only be called once for each process. Calling the initialization function more than once may cause an error.

Calling the initialization function when EhSvc.dll file is manipulated or the file version is not correct will result in an error (HS_ERR_INVALID_FILES).

Note

The part which calls the HackShield function from the game client program, not the HackShield interface DLL file, EhSvc.dll, could be manipulated. Therefore, it is recommended that the packer program encrypts, compresses and distributes the game client files, to prevent file manipulation in the game client program.

But, HackShield provides a feature that prevents games from being started by a cracked executable file using the server interface client crack.

Some users and hackers may attack the program by running a program in the lower compatibility mode provided on the Windows XP. If a program runs on the lower compatibility mode, the current system, Windows XP, will be considered as Windows 98, ME, or Windows 2000. This will cause unexpected results. If the initialization function of HackShield module running on the lower compatibility mode is called, an error (HS_ERR_COMPATIBILITY_MODE_RUNNING) will be returned.

When calling the initialization function, you can select additional options in addition to the default options.

Write HackShield service start function

To use the hack prevention and hacking tool detection features of HackShield, call the service start function, `_AhnHS_StartService`, as below:

```
Example
BOOL HS_StartService()
{
    int          nRet = 0;
    TCHAR szMsg[MAX_PATH];

    // ① Start the HackShield service by calling _AhnHS_StartService function.
    nRet = _AhnHS_StartService();

    // ② Check the return value of _AhnHS_StartService function and create an error.
    if ( nRet != HS_ERR_OK )
    {
        switch ( nRet )
        {
            case HS_ERR_START_ENGINE_FAILED:
            case HS_ERR_DRV_FILE_CREATE_FAILED:
            case HS_ERR_REG_DRV_FILE_FAILED:
            case HS_ERR_START_DRV_FAILED:
            default:
                wsprintf ( szMsg, "A problem occurred in the hack prevention
feature.(%x)", nRet );
                break;
        }

        MessageBox( NULL, szMsg, szTitle, MB_OK );
        return FALSE;
    }
    return TRUE;
}
```

Call the service start function, `_AhnHS_StartService`, after properly calling the initialization function, `_AhnHS_Initialize`.

Caution

For tighter security of the game client program, it is recommended to call the service start function as soon as calling the initialization function. The later you call the service start function, the more likely the hacking tool attacks the game client.

Write HackShield callback function

If you call the service start function, the current service status will be notified as an event. Add each event handling method in the callback function as shown in the following example:

Example

```
int __stdcall HS_CallbackProc ( long ICode, long IParamSize, void* pParam )
{
    TCHAR szMsg[MAX_PATH];

    // ① Display a proper error message for each case
    switch ( ICode )
    {
        // ② Engine Callback
        case AHNHS_ENGINE_DETECT_GAME_HACK:
            wsprintf( szMsg, "You cannot run the following program and game simultaneously. (%x) \n [%s]", ICode, (LPTSTR)pParam );
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            break;

        // ③ AutoMacro detection
        case AHNHS_ACTAPC_DETECT_AUTOMACRO:
            wsprintf(szMsg, _T("Suspicious macro operation has been detected. (Code = %x)", ICode);
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            break;

        // ④ Speed
        case AHNHS_ACTAPC_DETECT_SPEEDHACK:
            wsprintf( szMsg, "Suspicious speed hack operation has been detected. (%x)", ICode );
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            break;

        // ⑤ Debugging Prevention
        case AHNHS_ACTAPC_DETECT_KDTRACE:
        case AHNHS_ACTAPC_DETECT_KDTRACE_CHANGED:
            wsprintf( szMsg, "Attempt of game debugging has been detected. (%x)", ICode );
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            break;

        // ⑥ HackShield is running properly.
        case AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING:
            // HackShield is running properly.
            // Execute the logic defined by the game developer.
            DWORD *dwParam = (DWORD*)pParam;
            break;

        // ⑦ Other abnormal hack prevention features
        case AHNHS_ACTAPC_DETECT_DRIVERFAILED:
        case AHNHS_ACTAPC_DETECT_HOOKFUNCTION:
        case AHNHS_ACTAPC_DETECT_MODULE_CHANGE:
        case AHNHS_ACTAPC_DETECT_LMP_FAILED:
        case AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP:
        case AHNHS_AHNHS_ACTAPC_DETECT_ENGINEFAILED:
        case AHNHS_ACTAPC_DETECT_CODEMISMATCH:
        case AHNHS_ACTAPC_DETECT_ANTIFREESERVER:
        case AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS:
            wsprintf( szMsg, "There is a problem in the hack detection feature.
```



```
(%x)", ICode );
    MessageBox( NULL, szMsg, szTitle, MB_OK );
    break;

    return 1;
}
```

Caution

AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING callback is a callback that occurs when you call _AhnHS_CheckHackShieldRunningStatus.

If HackShield is running, this callback occurs regularly, and if callback does not occur regularly, there could be a problem in HackShield. The logic after callback occurrence is implemented according to the game developer's policy.

Write HackShield service stop function

Before terminating the game client program, call the _AhnHS_StopService function to stop the hack prevention and hacking tool detection features.

When stopping the HackShield service for a while only without terminating the game client program, call the this _AhnHS_StopService function and then call the _AhnHS_StartService again to restart the service.

```
Example
BOOL HS_StopService()
{
    int nRet = 0;

    // ① Stop the HackShield service by calling _AhnHS_StopService function.
    nRet = _AhnHS_StopService();

    if ( nRet != HS_ERR_OK )
    {
        return FALSE;
    }
    return TRUE;
}
```

Write HackShield service complete function

In order to completely terminate the game client program, call the _AhnHS_Uninitialize function as shown in the following example and deallocate the memory allocated for the program.

```
Example
BOOL HS_UnInit()
{
    int nRet = 0;

    // ① Terminate the HackShield service by calling _AhnHS_Uninitialize function.
    nRet = _AhnHS_Uninitialize();

    if ( nRet != HS_ERR_OK )
    {
        return FALSE;
    }
    return TRUE;
}
```

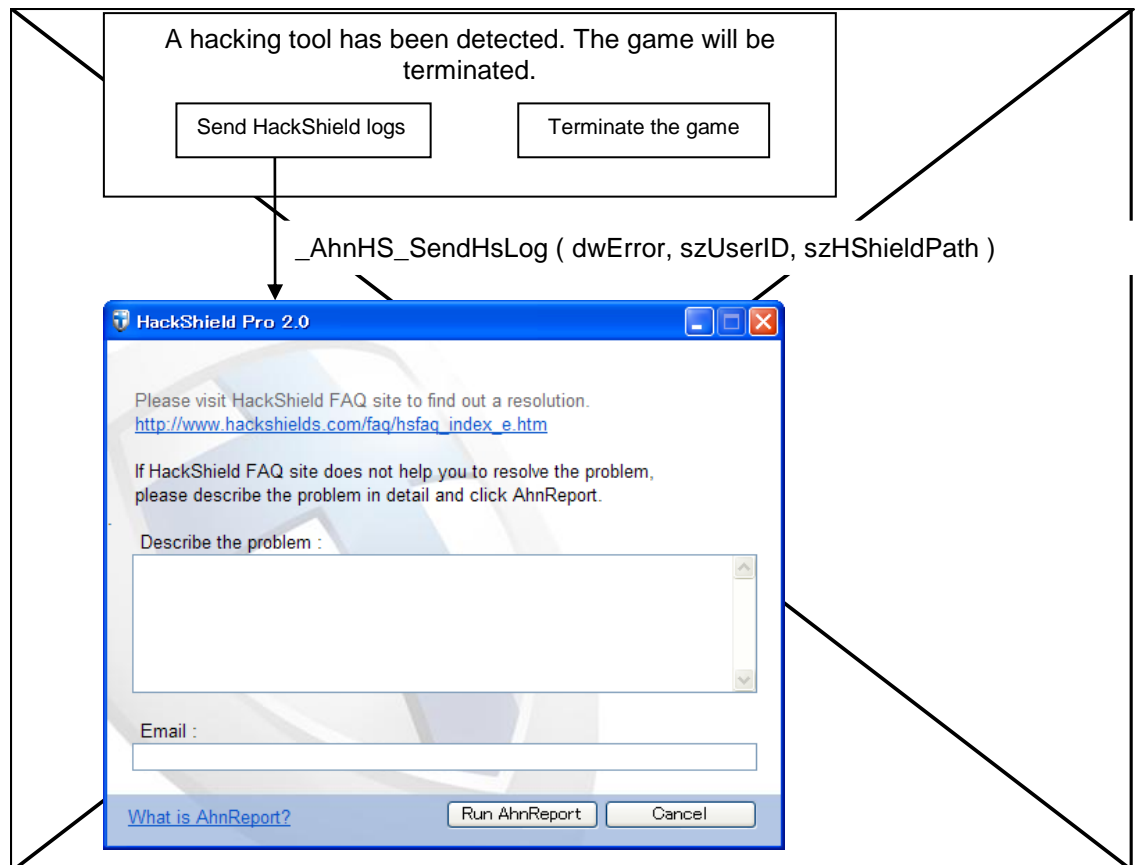
Once `_AhnHS_Uninitialize` is called, the game client can no longer be protected by HackShield. Therefore, call this function at the last point when the game client is terminated.

Caution

Abnormal termination of the game client program may prevent HackShield hack prevention driver from unloading. In this case, call the initialization function `_AhnHS_Initialize`; the error (`HS_ERR_INIT_DRV_FAILED`) is returned, since the HackShield hack prevention driver has been already loaded in the user's system. Reboot the system and restart the game client program.

Write HackShield logging function

This function is provided to directly and quickly solve problems related to HackShield without the game developer's notices. To send a log on hack attacks to AhnLab, implement the log transmission button in the message box or game termination dialogue box that processes the callback code, and call the `_AhnHS_SendHsLog` function.



If you enter the event code (error code), the game user ID, and the HackShield path that has been received from the HackShield callback code to the `_AhnHS_SendHsLog` and call `_AhnHS_SendHsLog` function, the AhnReport window will appear. Enter the detailed error information and his/her email address on the AhnReport window and sends them, it will be delivered to the AhnLab, along with HackShield log.

Caution

Program AhnReport window to open only when the game is terminated by a HackShield error. For the other types of errors, display a dialogue box or message box which cannot run AhnReport.

Caution

The szHShieldPath, the third argument of _AhnHS_SendHsLog is the path where ehsvc.dll resides. Write the folder path except Ehsvc.dll.
Ex) %GameRoot%\HShield

```
TCHAR szPath[MAX_PATH] = { NULL, }  
TCHAR szHShieldPath[MAX_PATH] = { NULL, };  
::GetModuleFileName(NULL, szPath, sizeof(szPath)/sizeof(*szPath));  
TCHAR* szPos = _tcsrchr(szPath, _T("\\")) + sizeof(_T("\\"));  
*szPos = _T("\\0");  
_stprintf(szHShieldPath, _T("%s\\HShield"), szPath );  
_AhnHS_SendHsLog(dwError, "UserID", szHShieldPath);
```

Note

The description of the errors and email on the AhnReport window is not mandatory; you do not have to write. Also, AhnLab will not send a reply to all customers as the sent HackShield log is only for reference on errors and wrong diagnosis.

Note

AhnReport window is supported by Windows 2000, Windows XP, Windows 2003, Windows Vista and Windows 7 (Win9x is not supported), and supports Korean, English, Chinese (traditional and simplified), and Japanese.

2.4. Application Programming Interface

AhnHS_Initialize

DESCRIPTION

Initializes HackShield and sets options. Can be called only once when the program is initialized. An error may be returned when another game program uses HackShield or the service has been abnormally terminated.

SYNTAX

```
EHSVC_API
int __stdcall
_AhnHS_Initialize(
    const char* szFileName
    PFN_AhnHS_Callback pfn_Callback,
    int nGameCode,
    const char* szLicenseKey,
    DWORD dwOption,
    UINT unSHackSensingRatio
);
```

PARAMETERS

Parameter	Value	Description
szFileName		Full path of EHSvc.dll
Pfn_Callback		Callback function pointer
nGameCode	4-digit numeric code	Unique code of each game
szLicenseKe	24-digit character string	License key for each game
dwOption		Initialization option setting
unSHackSensingRatio		Speed hack detection level

OPTIONS

Caution

The following shows the core options which are automatically applied.

AHNHS_CHKOPT_SPEEDHACK

```
AHNHS_CHKOPT_READWRITEPROCESSMEMORY
AHNHS_CHKOPT_KDTRACER , AHNHS_CHKOPT_OPENPROCESS
AHNHS_CHKOPT_AUTOMOUSE , AHNHS_CHKOPT_PROCESSSCAN
```

AHNHS_CHKOPT_SPEEDHACK (Core Option)

Uses the speed hack prevention feature. Notifies the user of the speed hack status through the callback function when hardware control or software-type speed hack is detected. The detection sensitivity depends on the detection level. Refer to the speed hack sensing ratio.

AHNHS_CHKOPT_READWRITEPROCESSMEMORY (Core Option)

Protects the memory of the current process that is using HackShield DLL from being used by other processes.

AHNHS_CHKOPT_KDTRACER (Core Option)

Detects kernel mode debugger, and notifies it to the game process.

AHNHS_CHKOPT_OPENPROCESS (Core Option)

Blocks calling of OpenProcess API function which provides information on the current process that is using HackShield DLL.

AHNHS_CHKOPT_AUTOMOUSE (Core Option)

Prevents auto mouse program from affecting currently running processes.

AHNHS_CHKOPT_PROCESSSCAN (Core Option)

Regularly checks the process list and check if hacking tools have been executed.

AHNHS_CHKOPT_ALL

Includes all options described above.

AHNHS_CHKOPT_MESSAGEHOOK

Blocks message hooking.

AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION

Protects the memory of the file specified to protect.

AHNHS_CHKOPT_ANTIFREESERVER

Prevents connection to an address that is not the game server's.

AHNHS_CHKOPT_SEND_MONITOR_DELAY

Waits until the user enters his/her ID when a hacking tool is detected before the user ID is entered, and then sends the log to the monitoring server.

Note

It is to send the hacking tool detection log including user information to the HackShield monitoring server.

AHNHS_USE_LOG_FILE

Stores HackShield operation logs. Log files are created as hshield.log under the directory where EhSvc.dll is stored. Log files are used for error analysis. Because log files are encrypted, the contents can be viewed only by a separate program.

AHNHS_ALLOW_SVCHOST_OPENPROCESS

Opens the game process in Svchost, a service program which runs on the NT-series OS. The newly added Windows Audio Service of Windows XP generates the sound. As the Windows Audio Service opens the client process, blocking this service will prevent sound generation. Add this option when DirectMusic or DirectSound does not produce the sound properly on Windows XP.

AHNHS_ALLOW_LSASS_OPENPROCESS

Allows LSASS.exe, a service program running on the NT-series OS, to open the game process. If the game client includes credit card payment control, LSASS.exe will directly access the game process. The game developer can open the game process by using this option.

AHNHS_ALLOW_CSRSS_OPENPROCESS

Allows CSRSS.exe, a service program running on the NT-series OS, to open the game process. If the game client includes credit card payment control, CSRSS.exe will directly access the game process. The game developer can open the game process by using this option.

AHNHS_DONOT_TERMINATE_PROCESS

By default, the hacking tool process is terminated forcibly when hacking tools are

detected by inspecting the process list. However, this option can be used additionally not to terminate the hacking tool process forcibly. It is recommended to use this option with `AHNHS_CHKOPT_PROCESSCAN` option.

AHNHS_DISPLAY_HACKSHIELD_LOGO

Displays the HackShield logo during initialization. It operates as a separate thread, and the logo will automatically disappear after two seconds. The logo shows that HackShield is running.

Note

When using the HackShield logo image provided by the game developer, change the image filename to `hslogo.jpg`, and copy it to `[Game Directory]\HShield` folder. The logo will appear when initializing HackShield.

AHNHS_CHKOPT_PROTECTSCREEN

Set this option to prevent hacking attacks more effectively which steals screen information from the game.

Caution

This option may cause the UI corruption of other programs and abnormal operation of the screen capture tool. Before using this option, contact AhnLab Technical Support .

AHNHS_ALLOW_SWITCH_WINDOW

Enables users to switch from game screen to other application windows while playing games that run on the full screen, as well as using the hacking tool detection feature that forcibly runs games on window mode. It is only applied to games that cannot run on window mode, so this cannot be used in games that support both full screen mode and window mode.

AHNHS_CHKOPT_STANDALONE

This is HackShield-standalone option. Generally, HackShield supports multi loading. But if you use this option, HackShield returns the error code to disable duplicated execution of HackShield for the games that use the same game code.

AHNHS_CHKOPT_PROTECT_D3DX

Activates the feature that blocks hacking tools that hook the VTable of DirectX module. This option is only available in Windows NT and later versions.

Note

For FPS games, it is recommended to use the option, as DirectX hacking tool takes up more than 90%.

AHNHS_CHKOPT_SELF_DESTRUCTION

If the game process is not terminated even after HackShield has detected a hacking attack and called the callback function, HackShield terminates the process after a specific period of time (1 minute).

This is used to detect hacking tools that modify the callback function to disable the HackShield hack detection feature.

When using this feature, you must terminate the game process properly within 1 minute of the HackShield error returning to the callback function, as HackShield could abnormally terminate the process.

When using this option, refer to the table below. With the table, define the callback code in the callback function and terminate the process. If none of the following callback code is defined in the callback function, HackShield may terminate the game abnormally.

Table 2-3 [Callback code when applying AHNHS_CHKOPT_SELF_DESTRUCTION]

Options	Callback code to be defined in the callback function
Required callback codes	AHNHS_ACTAPC_DETECT_AUTOMACRO
	AHNHS_ACTAPC_DETECT_SPEEDHACK
	AHNHS_ENGINE_DETECT_GAME_HACK
	AHNHS_ACTAPC_DETECT_ABNORMAL_MEMORY_ACCESS
	AHNHS_ACTAPC_DETECT_ENGINEFAILED
	AHNHS_ACTAPC_DETECT_HOOKFUNCTION
	AHNHS_ACTAPC_DETECT_KDTRACE
	AHNHS_ACTAPC_DETECT_CODEMISMATCH
	AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS
	AHNHS_ACTAPC_DETECT_DRIVERFAILED
AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION	AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP
	AHNHS_ACTAPC_DETECT_LMP_FAILED
AHNHS_CHKOPT_ANTIFREESEVER	AHNHS_ACTAPC_DETECT_ANTIFREESERVER
AHNHS_CHKOPT_PROTECTSCREENEX	AHNHS_ACTAPC_DETECT_PROTECTSCREENFAILED
AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL_V2	AHNHS_ACTAPC_DETECT_ABNORMAL_FUNCTION_CALL

Example

```
int __stdcall AhnHS_Callback(long ICode, long IParamSize, void* pParam)
{
```

```

switch (Icode)
{
    //Engine Callback
    case AHNHS_ENGINE_DETECT_GAME_HACK:
    case AHNHS_ACTAPC_DETECT_SPEEDHACK:
    case AHNHS_ACTAPC_DETECT_HOOKFUNCTION:
    case AHNHS_ACTAPC_DETECT_KDTRACE:
    case AHNHS_ACTAPC_DETECT_ABNORMAL_MEMORY_ACCESS:
    case AHNHS_ACTAPC_DETECT_ENGINEFAILED:
    case AHNHS_ACTAPC_DETECT_AUTOMACRO:
    case AHNHS_ACTAPC_DETECT_CODEMISMATCH:
    case AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP:
    case AHNHS_ACTAPC_DETECT_ANTIFREESERVER:
    case AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS:
    case AHNHS_ACTAPC_DETECT_ABNORMAL_FUNCTION_CALL:
        MessageBox ( );
        ExitGame ( );
        break;

    {

    return 1;
}

```

AHNHS_DISPLAY_HACKSHIELD_TRAYICON

When HackShield runs, the HackShield tray icon will be displayed in the system tray.

AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE

Prevents HackShield from running on a virtual machine or emulator.

If you apply this option, you can prevent games from running on a defined virtual machine or emulator.

When the virtual machine feature is changed, the detection logic could be added or changed.

Note

The AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE option will detect the following virtual machine programs.

Virtual PC, VMWare Workstation, Virtual Box, Parallels Workstation
(The option does not detect 9X series guest machine for Virtual Box and Parallels Workstation.)

AHNHS_CHKOPT_UPDATED_FILE_CHECK

After completing HackShield update,
Check if the HackShield runs with normally updated files.

(Caution)

Following constraints exists to use this feature:

- V3Hunt.dll file, the module related to HackShield update, must exist in the location where the HackShield runs.
- HSUpdate.env file, the module related to HackShield update, must exist in the location where the HackShield runs.
- The address of update server written in the HSUpdate.env file supports 'HTTP' only.
(If any address which supports FTP is written in the HSUpdate.env file, an error may occur as the feature is performed and the callback is sent to the game client.)

Following features are recommended for this feature:

- Perform Update feature by using `_AhnHS_HSUpdateEx()`
(“Game code” must be sent as an argument while the function is called).
 - Save “Game code” in the HSUpdate.env file through HSUpSetEnv.exe tool and distribute the file.
-

AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL_V2

Use this option to detect any hacking tools that call the protected functions in the game executable file and module.

Note

[For further details, please refer to `_AhnHS_VerifyProtectedFunction` section.](#)

AHNHS_CHKOPT_SEND_MONITOR_ONCE

Applies the two features as below:

Duplicate logs are sent to the monitoring server only once. (For instance, even if HackShield detects Hack.exe hacking tool several times, the log will be sent only once to the monitoring server.)
If the user ID is not entered, the log will not be sent to the monitoring server until the game is terminated. (However, if the user ID is not entered until the game is terminated, the log will be sent to the monitoring server without user information.)

(Caution)

HackShield determines whether the user ID has been entered based on whether the `_AhnHS_SetUserId` function has been called.

Table 2-4 [Callback function and return value of each HackShield option]

Options	Callback code and return value
AHNHS_CHKOPT_SPEEDHACK	AHNHS_ACTAPC_DETECT_SPEEDHACK
AHNHS_CHKOPT_KDTRACER	AHNHS_ACTAPC_DETECT_KDTRACE
AHNHS_CHKOPT_AUTOMOUSE	AHNHS_ACTAPC_DETECT_AUTOMACRO
AHNHS_CHKOPT_MESSAGEHOOK	AHNHS_ACTAPC_DETECT_MESSAGEHOOK
AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION	AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP
AHNHS_CHKOPT_ANTIFREESERVER	AHNHS_ACTAPC_DETECT_ANTIFREESERVER
AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL_V2	AHNHS_ACTAPC_DETECT_ABNORMAL_FUNCTION_CALL
AHNHS_CHKOPT_READWRITEPROCESSMEMORY	AHNHS_ACTAPC_DETECT_ABNORMAL_MEMORY_ACCESS
AHNHS_CHKOPT_PROCESSCAN	AHNHS_ENGINE_DETECT_GAME_HACK AHNHS_ENGINE_DETECT_WINDOWED_HACK
AHNHS_CHKOPT_UPDATED_FILE_CHECK	AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS
AHNHS_CHKOPT_STANDALONE	AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS Error code return (HS_ERR_ALREADY_GAME_STARTED)
AHNHS_CHKOPT_PROTECT_D3DX	AHNHS_ACTAPC_DETECT_HOOKFUNCTION
AHNHS_CHKOPT_SELF_DESTRUCTION	AHNHS_ACTAPC_DETECT_SELF_DESTRUCTION
AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE	Error code return (HS_ERR_VIRTUAL_MACHINE_DETECT)

Note

Most of the options in the table above are used for “detection”. Other protection features are processed internally, so some of the callback codes or return values are not delivered.

SPEEDHACK SENSING RATIO

AHNHS_SPEEDHACK_SENSING_RATIO_HIGHEST

Detects hacking tools that affect gaming by speeding up the game.

Being the most sensitive level, the reference data are 36.5% and -3.0%. When the speed is faster than the 1.0718 times or lower than 0.8409 times of A Speeder Speed Hack program, it will be considered as a speed hack.
(However, the slowdown due to the hardware is not detected.)

AHNHS_SPEEDHACK_SENSING_RATIO_HIGH

Detects hacking tools that affect gaming by speeding up the game.

Being the second most sensitive level, the reference data are 30.5% and -9.5%. When the speed is faster than the 1.1487 times or lower than 0.8123 times of A Speeder Speed Hack program, it will be considered as a speed hack.

AHNHS_SPEEDHACK_SENSING_RATIO_NORMAL

Being a normal sensitive level, the reference data are 26.0% and -12.5%. When the speed is faster than the 1.1892 times or lower than 0.7846 times of A Speeder Speed Hack program, it will be considered as a speed hack.

AHNHS_SPEEDHACK_SENSING_RATIO_LOW

Detects hacking tools that affect gaming by slowing down the game.

Being the second least sensitive level, the reference data are 22.5% and -15.5%. When the speed is faster than the 1.2311 times or lower than 0.7579 times of A Speeder Speed Hack program, it will be considered as a speed hack.

AHNHS_SPEEDHACK_SENSING_RATIO_LOWEST

Detects hacking tools that affect gaming by slowing down the game.

Being the least sensitive level, the reference data are 17.5% and -18.5%. When the speed is faster than the 1.2746 times or lower than 0.7320 times of A Speeder Speed Hack program, it will be considered as a speed hack.

RETURN VALUE**HS_ERR_OK (Value = 0x0000)**

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_INVALID_PARAM(Value = 0x002)

- Description: Incorrect parameter.
- Cause: Incorrect callback function pointer, or null license key.
- Workarounds: Only occurs during the development process. No special workaround is required.

HS_ERR_NOT_INITIALIZED(Value = 0x003)

- Description: Initialization failed.
- Cause: Occurs when the system library used in HackShield operation is not loaded properly.
- Workarounds: If this problem persists, contact AhnLab, Inc.

HS_ERR_COMPATIBILITY_MODE_RUNNING (Value = 0x004)

- Description: The game was executed in the compatibility mode.
- Cause: Occurs when the game client runs on the compatibility mode of the Windows XP series.
- Workarounds: Users running the game on the compatibility mode will be considered malicious users. Forcibly terminate the program and restart it.

HS_ERR_EXCEPTION_RAISED (Value = 0x007)

- Description: An exception occurred.
- Cause: Returned when any exception occurred in HackShield operation.
- Workarounds: Send HShield.log file and AhnReport to AhnLab, Inc.

HS_ERR_INVALID_LICENSE (Value = 0x100)

- Description: Incorrect license.
- Cause: The game code set as the parameter for the initialization function and the license key do not match with the actual value.
- Workarounds: Check the license key.

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: Incorrect HackShield file.
- Cause: Occurs when an interface DLL file (EhSvc.dll) is forged and the version is different. A forged interface DLL file may prevent hack prevention.
- Workarounds: Check if the HackShield file is a previous version or is in the HackShield folder.

HS_ERR_INIT_DRV_FAILED (Value = 0x102)

- Description: HackShield driver was not started.
- Cause: Occurs when the hack prevention driver is not initialized. In this case, hack attacks will not be properly blocked. You must restart the program.
- Workarounds:
 - ① Check if the permission for the System folder so that EagleXNT.sys or EagleX64.sys file is not accessible. Check if EagleXNT.sys or EagleX64.sys file is running.

HS_ERR_ALREADY_INITIALIZED (Value = 0x104)

- Description: HackShield has been already initialized.
- Cause: Occurs when the system has been already initialized by calling `_AhnHS_Initialize`.
- Workarounds: `_AhnHS_Initialize` can be called only once when initialize the program. This function cannot be called more than once.
Check if the above function has been called several times.

HS_ERR_DEBUGGER_DETECT (Value = 0x105)

- Description: A debugger was detected.
- Cause: Occurs when a debugger is running in the system.
- Workarounds:
 - ① Perform debugging using the Dev version, not a release version, during the development process. (Refer to Remark)
 - ② If this error occurs while users play games, there may be a hacking attack. Terminate the game and stop the debugger. Then, restart the game.

If the error occurs when there is no debugger, contact AhnLab, Inc.

HS_ERR_NEED_ADMIN_RIGHTS (Value = 0x107)

- Description: Admin account is required.
- Cause: Occurs when the game client was executed with a common user account permission, not the administrator account, on the Windows NT-series system.
- Workarounds: If the game client was executed by a common user account, check if a shadow account has been created. In order to use the HackShield features with a common user account permission, a HackShield shadow account must be created first. For more information, refer to '7.2. User Rights **Support**'.

HS_ERR_MODULE_INIT_FAILED (Value = 0x108)

- Description: HackShield Module Initialization failed.
- Cause: Occurs when a problem occurs while initializing to run HackShield.
- Workarounds: Send HShield.log file and AhnReport to AhnLab, Inc.

HS_ERR_UNKNOWN (Value = 0x001)

- Description: Unknown error occurred.
- Cause: There could be an exception in the function, or a problem in the function structure.
- Workarounds: Send HShield.log file and AhnReport to AhnLab, Inc.

REMARKS

In order to debug the game client during development or test process, use HShield.lib, Ehsvc.dll, and hshield.dat designed for developers.

(Please note that there could be a problem in server interworking if you use Ehsvc.dll and hshield.dat files along with the release version.)

- Path: \Developer\

If a developer module is used, the following logo will be displayed upon execution of HackShield. (Click this logo; the logo will disappear.))



AhnHS_Callback

DESCRIPTION

Callback function, which sends the result of the hack prevention and hacking tool detection, sends the following events:

Hack prevention events

Hacking tool detection events

STRUCTURE

```
int __stdcall AhnHS_Callback(  
    long ICode,  
    long IParamSize,  
    void* pParam  
);
```

PARAMETERS

Parameter	Value	Description
ICode	Long	Event code
IParamSize	Long	Event parameter size
pParam	Void*	Event parameter

EVENTS

AHNHS_ENGINE_DETECT_GAME_HACK

When a game hacking tool is detected on the system, the event will be sent as an argument of the callback. The game developer can forcibly delete the executable file of the hacking tool or take other action based on the passed hacking tool information.

pParam: Name of the executable file of the detected game hacking tool (including the file path)

IParamSize: Length of the detected hacking tool executable file name

AHNHS_ENGINE_DETECT_WINDOWED_HACK

When a hacking tool, which forcibly switches games from full screen mode to window mode, is detected, the event will be sent as an argument of the callback. It only applies to games that cannot run on window mode, so this cannot be used in games that support both full screen mode and window mode.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_ALREADYHOOKED

When the API function to protect from hacking has been already hooked while the hack prevention feature is running, this event will be sent. Hooking may occur in a normal program, not a hacking program, so the game developer must decide the policy for detecting the hacking tool.

pParam: ACTAPCPARAM_DETECT_HOOKFUNCTION*

IParamSize: ACTAPCPARAM_DETECT_HOOKFUNCTIONStructure Length

AHNHS_ACTAPC_DETECT_HOOKFUNCTION

When Win32 function or protection function has been hooked while the hack prevention feature is running, this event will be sent. When this event occurs, it is likely that this is due to a hacking tool. The game developer must forcibly terminate the game program.

pParam: ACTAPCPARAM_DETECT_HOOKFUNCTION*

IParamSize: ACTAPCPARAM_DETECT_HOOKFUNCTIONStructure Length

```
struct _ACTAPCPARAM_DETECT_HOOKFUNCTION
{
    char szFunctionName[128];
    char szModuleName[128];
} ACTAPCPARAM_DETECT_HOOKFUNCTION,
*PACTAPCPARAM_DETECT_HOOKFUNCTION;
```

szFunctionName: Name of hooked function

szModuleName: Name of hooked module

AHNHS_ACTAPC_DETECT_AUTOMOUSE

This event occurs when data is automatically entered by manipulating the keyboard or mouse with the auto mouse program. The following structure pointer will be sent.

pParam: ACTAPCPARAM_DETECT_AUTOMOUSE

IParamSize: ACTAPCPARAM_DETECT_AUTOMOUSE Size

```
typedef struct
{
    BYTE  byDetectType;
    DWORD dwPID;
    CHAR  szProcessName[16+1];
    CHAR  szAPIName[128];
} ACTAPCPARAM_DETECT_AUTOMOUSE,
*PACTAPCPARAM_DETECT_AUTOMOUSE;
```

The value and description is described as follows. dwPID, szProcessName, and szAPIName are not currently in use.

EAGLE_AUTOMOUSE_APCTYPE_SHAREDMEMORY_ALTERATION (3):

Changes the internal data of the library which blocks the auto mouse feature. If a hacking program modifies the internal information used by HackShield to block hack attacks, the hack prevention feature will not properly function. In this case, the developer must forcibly terminate the game program.

EAGLE_AUTOMOUSE_APCTYPE_API_CALLED: Calls APIs related to keyboard and mouse.

EAGLE_AUTOMOUSE_APCTYPE_API_ALTERATION: API hooking manipulation

AHNHS_ACTAPC_DETECT_AUTOMACRO

This event occurs when data is automatically entered by manipulating the keyboard or mouse with the auto mouse program. As abnormal data is repeatedly entered, you must forcibly terminate the game program.

pParam: ACTAPCPARAM_DETECT_AUTOMACRO

IParamSize: ACTAPCPARAM_DETECT_AUTOMACRO Size

```
typedef struct
{
    BYTE byDetectType;
    CHAR szModuleName[128];
} ACTAPCPARAM_DETECT_AUTOMACRO,
*PACTAPCPARAM_DETECT_AUTOMACRO;
```

#define	EAGLE_AUTOMACRO_APCTYPE_KEYBOARD	1
#define	EAGLE_AUTOMACRO_APCTYPE_MOUSE	2

The value and description is described as follows. szModuleName is not currently in use.

EAGLE_AUTOMACRO_APCTYPE_KEYBOARD: Auto mouse (mouse-related)

EAGLE_AUTOMACRO_APCTYPE_MOUSE: Auto mouse (keyboard-related)

AHNHS_ACTAPC_DETECT_DRIVERFAILED

This event occurs if the hack prevention driver has not been loaded to the system. If the hack prevention driver has not been loaded to the system, the hack prevention feature will not operate properly. In this case, the game program must be immediately terminated. If the driver is abnormally removed, the system may become unstable. So, the system must be restarted.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_SPEEDHACK

This is an event that occurs when the time speed in the system is abnormal. It is very likely that a hardware or software-type speed hack is running. If the game program is sensitive to time, the game program must be forcibly terminated. Hardware-type speed hack directly manipulates hardware time affecting the entire system time. This kind of hardware-type speed hack is sometimes blocked at the OS level depending on the Windows version.

pParam: (double *) Time data for the last five seconds

IParamSize: Passed time data count as pParam

AHNHS_ACTAPC_DETECT_KDTRACE

This event occurs when a debugger trace is generated by the kernel-level or application-level debugger. When this event occurs, it is very likely that a malicious user like a hacker is debugging the game program. It is recommended to terminate the game program when this event occurs.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_ABNORMAL_MEMORY_ACCESS

This event occurs when an unauthorized process accesses the game process memory. The process can be identified through the path and the name of the process trying to access the memory.

pParam: Name of the executable file of the detected game hacking tool (including the file path)

IParamSize: Length of the detected hacking tool executable file name

AHNHS_ACTAPC_DETECT_KDTRACE_CHANGED

This event occurs when the debugger trace blocking routine is changed. This event occurs when the debugger program at the kernel level is activated while the debugger trace blocking routine for the game program is executed. The game program must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_MODULE_CHANGE

This event occurs when HackShield module manipulation is detected. When this event occurs, HackShield may not properly operate. The game program must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_ENGINEFAILED

This event occurs when the heuristic engine file (3n.mhe) of HackShield is deleted or not normally loaded. When this event occurs, ProcessScan feature of HackShield may not properly operate. The game program must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_CODEMISMATCH

This event occurs when the code area of Ehsvc.dll module, the interface module of HackShield, is manipulated. When the event occurs, HackShield features may not properly operate. The game program must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP

This event occurs when the memory area of the file to be protected has been manipulated. When this event occurs, the corresponding module may not properly operate. The game program must be forcibly terminated.

pParam: Name of the manipulated module (Starting address of manipulated page)

IParamSize: 0

AHNHS_ACTAPC_DETECT_LMP_FAILED

This event occurs when the LMP protection feature does not operate properly due to hack attack or problem in the system

. Restart the system and game.

IParamSize: 0

AHNHS_ACTAPC_DETECT_ANTIFREESERVER

This event occurs when the game program attempts to access a server with an abnormal address. As this is an abnormal process, the game must be forcibly terminated. pParam is the address value. It is recommended to use this address value for internal use only, not showing it to the user.

pParam: IP address for connection

IParamSize: 0

AHNHS_ACTAPC_DETECT_ABNORMAL_HACKSHIELD_STATUS

This event occurs when HackShield does not operate properly. When this event occurs, HackShield may not properly operate. The game program must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_DETECT_PROTECTSCREENFAILED

This event occurs when the feature that protects the game with a virtual desktop does not operate after being attacked. If this event occurs, the virtual desktop feature will no longer operate, so the game program that uses the virtual desktop feature must be forcibly terminated.

pParam: NULL

IParamSize: 0

AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING

This is a status callback which occurs when HackShield operates properly. The feature is started by calling the `_AhnHS_CheckHackShieldRunningStatus` function. If HackShield operates properly, this event occurs every 25 seconds before `_AhnHS_StopService` and `_AhnHS_Uninitialize` functions are called.

If this event does not occur periodically, it means that HackShield is not operating properly.

pParam: (DWORD *) HackShield operation status value (one of the values defined in `enum HS_RUNNING_STATUS`)

IParamSize: pParam size

```
enum HS_RUNNING_STATUS {  
    HS_RUNNING_STATUS_CHECK_MONITORING_THREAD = 1,  
};
```

Caution

Now, `AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING` callback is called every 25 seconds internally. 25-second cycle may be used as the callback cycle according to the system.

[_AhnHS_StartService](#)

DESCRIPTION

Runs the hacking tool detection and hack prevention features. Shall be called after `_AhnHS_Initialize` function is called, and cannot be called more than once. If the service was stopped by `_AhnHS_StopService` function, the function may be called again to restart the service.

SYNTAX

```
EHSVC_API  
int __stdcall  
_AhnHS_StartService( );
```

PARAMETERS

None.

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x003)

- Description: HackShield has not been initialized.
- Cause: `_AhnHS_Initialize` function was not called or the function was called as HackShield had not been initialized.
- Workarounds: This error occurs only in the development process, so no other workaround is required.

HS_ERR_START_ENGINE_FAILED (Value = 0x200).

- Description: Engine loading failed.
- Cause: Occurs when initialization of the hacking tool pattern engine fails although the hacking tool process detection option (`AHNHS_CHKOPT_PROCESSSCAN`) was set as the initialization function was called. Also occurs when the hacking tool pattern engine-related files were not properly installed or the pattern engine-related files were not properly installed in HackShield folder.
- Workarounds: The program must be forcibly terminated and restarted or reinstalled.

HS_ERR_ALREADY_SERVICE_RUNNING (Value = 0x201)

- Description: HackShield has been already running.
- Cause: Occurs when `_AhnHS_StartService` function is called again while the function has been already called.

- Workarounds: In order to call this function again, the developer must stop the service first by calling `_AhnHS_StopService`. This error occurs only in the development process, so no other workaround is required.

HS_ERR_DRV_FILE_CREATE_FAILED (Value = 0x202)

- Description: Creation of HackShield driver file failed.
- Cause: Occurs when creation of a driver file for hack prevention failed. The HackShield program creates and loads the hack prevention driver in the game starting point. This error occurs when a driver file is not properly created upon starting the game.
- Workarounds:
 - ① Check whether the current session has write permission for the System folder.
 - ② Check if the HackShield driver (EagleXNT.sys or EagleX64.sys) in the System folder is read-only or inaccessible. In this case, delete the file and restart HackShield.

(If this problem persists, contact AhnLab, Inc.)

HS_ERR_REG_DRV_FILE_FAILED (Value = 0x203)

- Description: Registration of the HackShield drive file failed.
- Cause: Occurs when registration of the driver file for hack prevention failed.
- Workarounds: Once this error occurs, HackShield will not properly function. Forcibly terminate the program, and restart or reinstall the program.

HS_ERR_START_DRV_FAILED (Value = 0x204)

- Description: Running of the HackShield drive failed.
- Cause: Occurs when the game is restarted while the game client has been abnormally terminated due to various causes.
In this case, when the `_AhnHS_StopService` or `_AhnHS_Uninitialize` functions are not called, the driver which had been loaded previously is stopped or any driver which should not have been loaded had been loaded. When this error occurs, the system becomes unstable or may have any problem. Terminate and restart the program.
- Workarounds:
 - ① Check whether the current session has write permission for the System folder.
 - ② Check if the HackShield driver (EagleXNT.sys or EagleX64.sys)

in the System folder is read-only or inaccessible. In this case, delete the file and restart HackShield.

If this problem persists, contact AhnLab, Inc.)

HS_ERR_ALREADY_GAME_STARTED (Value = 0x206)

- Description: The game is already running.
- Cause: This error occurs when HackShield's standalone option (AHNHS_CHKOPT_STANDALONE) is applied when the initialization function is called. This occurs when the game is already running with the same game code.
- Workarounds: If HackShield's standalone option (AHNHS_CHKOPT_STANDALONE) is applied as the initialization function is called, you can run only one game process. If this error occurs, the program must be terminated.)

HS_ERR_VIRTUAL_MACHINE_DETECT (Value = 0x207)

- Description: The game is already running on the virtual machine or the emulator.
- Cause: This error occurs when the option (AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE), which prevents games from running on the virtual machine or the emulator, is applied as the initialization function is called.
- Workarounds: If the option (AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE) is applied, games will be protected from running on the virtual machine controlled by HackShield.

_AhnHS_StopService

DESCRIPTION

Stops hack prevention and hacking tool detection features.

SYNTAX

```
EHSVC_API  
int __stdcall  
_AhnHS_StopService( );
```

PARAMETERS

None.

RETURN VALUE

HS_ERR_OK (Value = 0x0000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x0003)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: This error occurs only in the development process, so no other workaround is required.

HS_ERR_SERVICE_NOT_RUNNING (Value = 0x301)

- Description: HackShield has not started.
- Cause: Occurs when _AhnHS_StartService function is called as HackShield has not been started. This error occurs only in the development process, so no other workaround is required.

- Workarounds: This error occurs only in the development process, so no other workaround is required.

_AhnHS_Uninitialize

DESCRIPTION

Deallocates the occupied memory in the system and initializes the parameters.

SYNTAX

```
EHSVC_API  
int __stdcall  
_AhnHS_Uninitialize ( );
```

PARAMETERS

None.

RETURN VALUE

HS_ERR_OK (Value = 0x0000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_SERVICE_STILL_RUNNING (Value = 0x302)

- Description: HackShield is running.
- Cause: Occurs when _AhnHS_StopService function is called as HackShield has not been terminated. This error occurs only in the development process, so no other workaround is required.
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x003)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.

- Workarounds: This error occurs only in the development process, so no other workaround is required.

_AhnHS_PauseService

DESCRIPTION

Pauses some HackShield features. Applicable only to the keyboard protection among message hooking prevention features. In other words, this is used to pause the keyboard protection, which is to block entry on Microsoft Internet Explorer web page for payment while the game is running. After pausing the keyboard protection, this function must be activated again with `_AhnHS_ResumeService`.

SYNTAX

```
EHSVC_API
int __stdcall
_AhnHS_PauseService (
    DWORD dwPauseOption
);
```

PARAMETERS

Parameter	Value	Description
dwPauseOption	DWORD	Currently, only <code>AHNHS_CHKOPT_MESSAGEHOOK</code> option is available. If other options are sent, <code>HS_ERR_INVALID_PARAM</code> error will be returned.

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x003)

- Description: HackShield has not been initialized.
- Cause: `_AhnHS_Initialize` function was not called or the function was

called as HackShield had not been initialized.

- Workarounds: This error occurs only in the development process, so no other workaround is required.

HS_ERR_SERVICE_NOT_RUNNING (Value = 0x301)

- Description: HackShield has not started.
- Cause: Occurs when `_AhnHS_StartService` function is called as HackShield has not been started. This error occurs only in the development process, so no other workaround is required.
- Workarounds: This error occurs only in the development process, so no other workaround is required.

HS_ERR_INVALID_PARAM (Value = 0x002)

- Description: Wrong parameters.
- Cause: Occurs when `dwPauseOption` is not `AHNHS_CHKOPT_MESSAGEHOOK`.
- Workarounds:

_AhnHS_ResumeService

DESCRIPTION

Calls AhnHs_Pause service and resumes HackShield features. Applicable only to the keyboard protection feature among message hooking prevention features. For more information about this feature, refer to _AhnHS_PauseService.

SYNTAX

```
EHSVC_API  
int __stdcall  
_AhnHS_ResumeService (  
    DWORD dwResumeOption  
);
```

PARAMETERS

Parameter	Value	Description
dwResumeOption	DWORD	Currently, only AHNHS_CHKOPT_MESSAGEHOOK option is available. If other options are sent, HS_ERR_INVALID_PARAM error will be returned.

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x003)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: This error occurs only in the development process, so no

other workaround is required.

HS_ERR_SERVICE_NOT_RUNNING (Value = 0x301)

- Description: HackShield has not started.
- Cause: Occurs when `_AhnHS_StartService` function is called as HackShield has not been started. This error occurs only in the development process, so no other workaround is required.
- Workarounds: This error occurs only in the development process, so no other workaround is required.

HS_ERR_INVALID_PARAM (Value = 0x002)

- Description: Wrong parameters.
- Cause: Occurs when `dwPauseOption` is not `AHNHS_CHKOPT_MESSAGEHOOK`.
- Workarounds:

_AhnHS_CheckHackShieldRunningStatus

DESCRIPTION

Activates the feature that periodically checks the HackShield operation status and sends the status callback to the game.

This function is called to check if HackShield is operating normally on games.

Runs from after HackShield has been started (`_AhnHS_Initialize`, `_AhnHS_StartService` function call) until it stops (`_AhnHS_StopService`, `_AhnHS_Uninitialize` function call).

SYNTAX

```
EHSVC_API  
int __stdcall  
_AhnHS_CheckHackShieldRunningStatus ();
```

PARAMETERS

None

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal If this value is returned, the HackShield operation status is checked periodically and the callback is returned to the game.
- Workarounds:

HS_ERR_NOT_INITIALIZED (Value = 0x003)

- Description: Initialization failed.
- Cause: Occurs when `_AhnHS_StartService` function is called as HackShield has not been started. This error occurs when the system library used for running HackShield is not loaded properly.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

Note

If this function is applied, AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING status callback will occur periodically. This callback notifies HackShield's normal operation status, not hacking detection, so it can be used to enhance the security of games.

AhnHS_SendHsLog

DESCRIPTION

If you want to send the HackShield log to AhnLab, Inc., call the `_AhnHS_SendHsLog` function, along with the event code, game user ID, and HackShield path. Implement a log saving button and a termination button for users to be able to send logs selectively on the game termination dialogue box of the callback function. Make sure that the game is terminated only after `_AhnHS_SendHsLog` function is called as the log saving function is clicked.

SYNTAX

```
void
__stdcall
_AhnHS_SendHsLog (
    IN DWORD dwError,
    IN const char* szUserID
    IN const char* szHShieldPath

);
```

PARAMETERS

Parameter	Value	Description
dwError	DWORD	Enter the event code (Icode) that comes over to the first argument of the HackShield callback function.
szUserID	const char *	Game user ID
szHShieldPath	const char *	The HackShield path where Ehsvc.dll exists. Enter the folder path excluding ehsvc.dll.

RETURN VALUE

None.

_AhnHS_VerifyProtectedFunction

DESCRIPTION

This function protects the internal functions of the game module from being called by external modules.

Do not use when using the following functions – only use the macro defined by the header file.

(AHNHS_PROTECT_FUNCTION or AHNHS_PROTECT_FUNCTIONEX)

SYNTAX

```
int  
__stdcall  
_AhnHS_VerifyProtectedFunction();
```

PARAMETERS

None.

RETURN VALUE

None.

REMARKS

Features

- Option handling
(Option handling stage can be omitted in modules from which the AhnHS_Initialize or AhnHS_StartService function is not called.)
Add the following option in AhnHS_Initialize function.

AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL_V2

```
nRet = _AhnHS_Initialize ( szFullFilePath,  
                           HS_CallbackProc, // callback function  
                           1000,           // game code  
                           "B228F291B7D7FAD361D7A4B7", // License key  
                           AHNHS_CHKOPT_ALL|AHNHS_CHKOPT_ABNORMAL_FUNCTION_CALL_V2, // option flag  
                           AHNHS_SPEEDHACK_SENSING_RATIO_NORMAL );
```

- Add macro to protected function.
Apply macro to the protected function as below:

Example of using macro>

```
int CPlayer::move ( int x, int y )
```

```

{
    CheckTheHealth();
    //apply MACRO
    AHNHS_PROTECT_FUNCTION
    KillMonster( x, y );
    return 1;
}

int CPlayer::move ( int x, int y )
{
    CheckTheHealth();
    int nRet =0;
    // apply MACRO.
    AHNHS_PROTECT_FUNCTIONEX (nRet)
    use nRet value for debugging
    KillMonster( x, y );
    return 1;
}

```

- Add Callback Code

AHNHS_ACTAPC_DETECT_ABNORMAL_FUNCTION_CALL

```

)
    int __stdcall HS_CallbackProc ( long ICode, long IParamSize, void* pParam
    {
        TCHAR szMsg[MAX_PATH];

        switch ( ICode )
        {
        case AHNHS_ACTAPC_DETECT_ABNORMAL_FUNCTION_CALL:
            ExitGame ( );
            break;
        }
    }

```

Caution

- The feature will operate only when StartService is properly called.
 - Only operates on NT and later versions (Win98 is not supported.)
 - To apply the macro, a header file (HShield.h) and library (HShield.lib, winmm.lib, version.lib) must be linked to the project to apply.
 - The game module that calls the protected function must be digitally signed.
- But, there are the following exceptions:.

- ① When the protected function is called by the same module.
- ② When the protected function is called by the EXE module.

_AhnHS_QueryPerformanceCounter

DESCRIPTION

This is a function provided by HackShield to get secure QueryPerformanceCounter value.

SYNTAX

```
BOOL
__stdcall
_AhnHS_QueryPerformanceCounter(
    OUT LARGE_INTEGER *lpPerformanceCount,
    OUT int *pErr);
```

PARAMETERS

Parameter	Value	Description
lpPerformanceCounter	LARGE_INTEGER	Pointer for parameter that receives current QueryPerformanceCounter value
pErr	Int	Detail on error when _AhnHS_QueryPerformanceCounter() has failed

HS_ERR_NOT_INITIALIZED(Value = 0x003)

- Description: Initialization failed.
- Cause: Occurs when the system library used in HackShield operation is not loaded properly.
- Workarounds: If this problem persists, contact AhnLab, Inc

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

RETURN VALUE

TRUE

- Description: Success

FALSE

- Description: Failure
- Workarounds: Check the second parameter value.

REMARKS

Features

If the return value for `_AhnHS_QueryPerformanceCounter` is false, check the details on the error with the parameter's value.

If the second parameter is not `ERR_NOT_INITIALIZED` or `ERR_INVALID_FILES`, it means the `QueryPerformanceCounter` function has failed.

```
int nErr = 0;
LARGE_INTEGER liCurrent = { 0 };

if( FALSE == _AhnHS_QueryPerformanceCounter ( & liCurrent, &nErr ) )
{
    if (nErr == HS_ERR_NOT_INITIALIZED)
        // Failed to initialize HackShield
    else if(nErr == HS_ERR_INVALID_FILES)
        // Failed to initialize HackShield
    else
    {
        //QueryPerformanceCounter API FAILED.
    }
}
else
{
    // Success
}
```

_AhnHS_QueryPerformanceFrequency

DESCRIPTION

This is a function provided by HackShield to get secure QueryPerformanceFrequency value.

SYNTAX

```
BOOL  
__stdcall  
_AhnHS_QueryPerformanceFrequency (  
    OUT LARGE_INTEGER *lpFrequency,  
    OUT int *pErr);
```

PARAMETERS

Parameter	Value	Description
lpFrequency	LARGE_INTEGER	Pointer for parameter that receives current QueryPerformanceFrequency value
pErr	Int*	Detail on error when _AhnHS_QueryPerformanceFrequency() has failed

HS_ERR_NOT_INITIALIZED(Value = 0x003)

- Description: Initialization failed.
- Cause: Occurs when the system library used in HackShield operation is not loaded properly.
- Workarounds: If this problem persists, contact AhnLab, Inc.

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

RETURN VALUE

TRUE

- Description: Success

FALSE

- Description: Failure
- Workarounds: Check the second parameter value.

REMARKS

Features

If the return value for `_AhnHS_QueryPerformanceFrequency` is false, check the details on the error with the parameter's value.

If the second parameter is not `ERR_NOT_INITIALIZED` or `ERR_INVALID_FILES`, it means the `QueryPerformanceFrequency` function has failed.

```
int nRet = 0;
LARGE_INTEGER liFrequency = { 0 };

if( FALSE == _AhnHS_QueryPerformanceFrequency ( &liFrequency, &nRet ) )
{
    if (nRet == HS_ERR_NOT_INITIALIZED)
        // Failed to initialize HackShield
    else if(nRet == HS_ERR_INVALID_FILES)
        // Failed to initialize HackShield
    else
    {
        //QueryPerformanceFrequency API FAILED.
    }
}
else
{
    // Success
}
```

_AhnHS_GetTickCount

DESCRIPTION

This is a function provided by HackShield to get secure GetTickCount value.

SYNTAX

```
unsigned long  
__stdcall  
_AhnHS_GetTickCount (OUT int *pErr);
```

PARAMETERS

Parameter	Value	Description
pErr	Int*	Detail on error when _AhnHS_GetTickCount() has failed

HS_ERR_NOT_INITIALIZED(Value = 0x003)

- Description: Initialization failed.
- Cause: Occurs when the system library used in HackShield operation is not loaded properly.
- Workarounds: If this problem persists, contact AhnLab, Inc.

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

RETURN VALUE

TRUE

- Description: Success

FALSE

- Description: Failure
- Workarounds: PARAMETERS값을 확인

REMARKS

Features

If the return value for `_AhnHS_GetTickCount` is false, check the details on the error with the parameters value.

If the second parameter is not `ERR_NOT_INITIALIZED` or `ERR_INVALID_FILES`, it means the `GetTickCount` function has failed.

```
int nRet = 0;

unsigned long ulTime = _AhnHS_GetTickCount ( &nRet )
if ( ulTime == 0 )
{
    if (nRet == HS_ERR_NOT_INITIALIZED)
        // Failed to initialize HackShield
    else if(nRet == HS_ERR_INVALID_FILES)
        // Failed to initialize HackShield
    else
    {
        // GetTickCount API FAILED.
    }
}
else
{
    // Success
}
```

3. HackShield Update

3.1. Overview

HackShield update feature is designed to help game developer build within the patch period and easily update HackShield through the launcher. HackShield-exclusive update supports quick HackShield module and engine patches and manages hacking tools quickly.

Functions

HackShield Module Update

HackShield update is provided for easy update of the HackShield module when a HackShield patch is released.

Engine Update

In order to immediately manage the hacking tools, HackShield update can update signatures and the heuristic engines.

HackShield update ON/OFF feature

HackShield update can be turned off/on according to the patch set settings, in case you need to cancel the update process while the HackShield update running.

Splash image feature

Displays the image specified by the user in the middle and bottom right of the screen during HackShield update.

Features

Interface Function (API)

The game launcher or executable file calls this API provided by HSUpChk.lib in order to update HackShield modules and engines.

Update Environmental File Creation Program

Provides update server configuration file in order to set the server to update and build multiple update servers through multiple URLs.

Test Program

Provides Amazon.exe, a test program implemented by HSUpChk.lib APIs. Amazon.exe provides the existing HackShield test and HackShield update test

functions.

System Architecture

HackShield update provides HSUpChk.lib for the client to call the API and perform update. The general architecture and operating principles of HackShield update are as follows:

HSUpChk.lib (HackShield update library)

Used by the client. When update function is called, HsUpdate.exe will be also called in order to update HackShield module.

HSUpdate.env (Update Environmental File)

Created by HSUpSetEnv.exe tool. Contains update server configuration information.

HSUpSetEnv.exe (Update Environmental File Creation Program)

A tool that creates HSUpdate.env update configuration file, you can configure the update server.

noupdate.ui (update inactivation (OFF) configuration file)

noupdate.ui is a file to temporarily inactivate (OFF) the updating process.

The HackShield update features will be deactivated (OFF) if you place the noupdate.ui file under the HackShield patchset (same location as ahn.ui and autoup.exe) in the update server.

The HackShield update features will be activated (ON) if you delete the noupdate.ui file under the HackShield patchset (same location as ahn.ui and autoup.exe) in the update server.

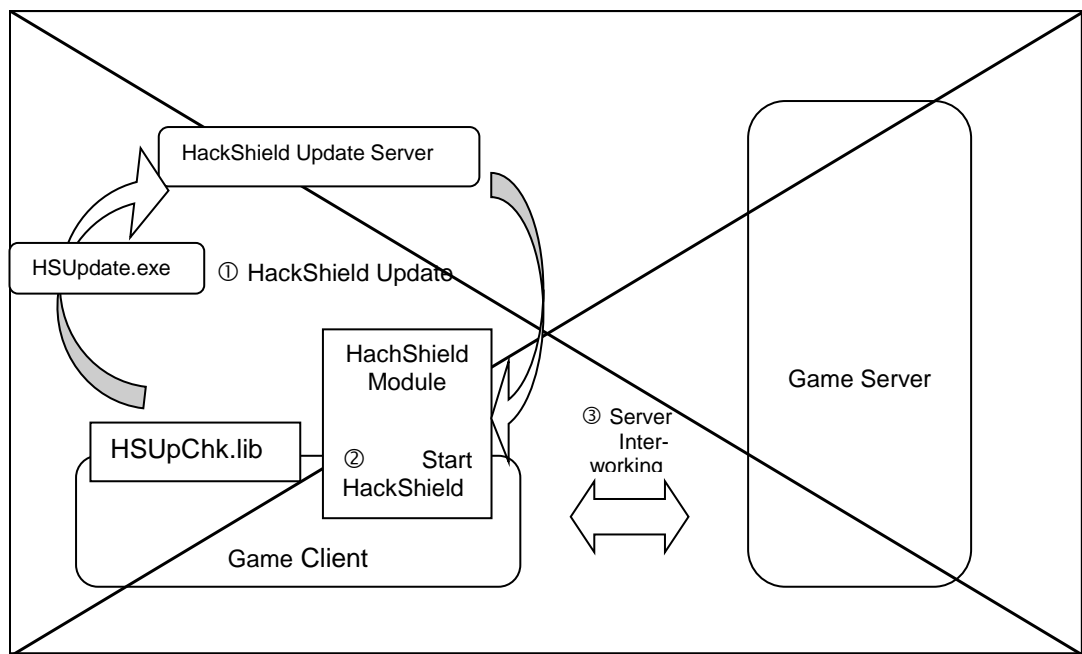


Fig. 3-1 HackShield update

- ① `_AhnHS_HSUpdate`, an API provided by `HSUpchk.lib` is called to access the HackShield update server and update HackShield module and engine.
- ② `_AhnHS_Initialize` and `_AhnHS_Start` functions provided by `HShield.lib` are called to execute HackShield.
- ③ Accesses the game server and starts server interworking.

3.2. Application Programming

This chapter describes how to update HackShield by using APIs provided by `HSUpChk.lib`.

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

Programming Application

Follow the preparation below before starting programming by using `HSUpChk.lib`.

3.2.1.1. Update-related File

Update-related File

Table 3-1 update-related File

File name	Installation folder	Description
HSUpChk.lib	[Program source folder]	Header file to be used by the server
HSUpChk.h	[Program source folder]	DLL file to be used by the server
AhnUpCtl.dll	[Game]\HShield	Update dll
AhnUpGS.dll	[Game]\HShield	Update dll
HSInst.dll	[Game]\HShield	UI dll
HSUpdate.env	[Game]\HShield	Update configuration file
HSUpdate.exe	[Game]\HShield	Update executable file
V3Hunt.dll	[Game]\HShield	Update dll
V3InetGS.dll	[Game]\HShield	Update dll

Table 3-2 Files that need to be installed on update server

PatchSet Path: [HackShield SDK] \PatchSet\

File name	Description
ahn.ui	Update information file (engine file version information)
ahn.uic	Update information file (engine file version information)
ahni2.dll	Update file
ahnupctl.dll	Update file
autoup.exe	Update file
v3bz32.dll	Update file
patch\39\3n.mh-	Heuristic engine file (compressed file)
patch\39\ahn.ui	Update information file (patch file version information)
patch\39\ahn.uic	Update information file (patch file version information)
patch\39\ahnupctl.dl-	Update file (compressed file)
patch\39\ahnupgs.dl-	Update file (compressed file)
patch\39\ehsvc.dl-	HackShield interface DLL (compressed file)
patch\39\hshield.da-	HackShield dat file (compressed file)
patch\39\hsinst.dl-	Update file (compressed file)
patch\39\hsupdate.ex-	Update executable file (compressed file)
patch\39\v3hunt.dl-	Update file (compressed file)
patch\39\v3inetgs.dl-	Update file (compressed file)
win\eb\b_echo_sl\asc_c om.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\asc_d h.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\asc_f se.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\asc_i	Hacking tool detection engine interface DLL

ntg.dl-	(compressed file)
win\eb\b_echo_sl\asc_mmgr.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\asc_unp.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_fact.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_pe.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_fact.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_file.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_memory.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_operations.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_processing.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_echo_sl\ase_gfs_util.dl-	Hacking tool detection engine interface DLL (compressed file)
win\eb\b_sign_hs\asc.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_sign_hs\ascure.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_sign_hs\asgame.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_sign_hs\aspe3f.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_sign_hs\modules.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_sign_hs\options.sc-	Hacking tool pattern engine file (compressed file)
win\eb\b_v3_echo_hs\v3pro32s.dl-	Hacking tool detection engine interface DLL (compressed file)

Table 3-3 Data files used for update

Data file path: [HackShield SDK] \ Data\

File name	Description
noupdate.ui	Configuration file to temporarily inactivate (OFF) update.

The following files are automatically created after update.

Table 3-3 Automatically created files after HackShield update

File name	Installation folder	Description
ahn.ui	[HackShield Folder] /Update/	Update information file
ahn.uic	[HackShield Folder] /Update/	Update information file
ahni2.dll	[HackShield Folder] /Update/	Update DLL
ahnupctl.dll	[HackShield Folder] /Update/	Update DLL
autoup.exe	[HackShield Folder] /Update/	Update executable file (File patch processing)
v3bz32.dll	[HackShield Folder] /Update/	Update DLL (File compression processing)

3.2.1.2. Application

Server management method

HackShield update server must be prepared, set and managed by the publisher (by region)

Table 3-4 HackShield update server specification

Category	Recommended specifications	Minimum requirements
Operating system	Windows Server 2003	Windows Server 2000
CPU	Intel Xeon 3.2 Dual	Intel Pentium IV 2.4
RAM	2G or more	1G or more
HDD	40G or more	40G or more

1. Configures an update server which is accessible via FTP or HTTP through IIS setting.

Note

For FTP, a separate account must be set or anonymous access must be allowed.

Also, passive-mode access must be allowed. In order to check if the passive-mode access is allowed, access FTP through the IE and check if the file list is viewed.

2. Upload all files under [HackShield SDK] \PatchSet folder to the server.
For the file list, see Table 3-2.

Caution

To upload the patch set, a subfolder must be created under the root folder in the server.

Caution

Note the following information when setting update address.

Do not include “ahnlab” in the update domain address as setting the address. Some worm viruses restrict access to website with the domain address including “ahnlab”.

Example: When the update address is set to ahnlabGame.co.kr/real/, this site will be blocked, as “ahnlab” is included in the address.

Caution – IIS (Internet Information Services) Settings

DLL or EXE executable file maybe not be downloadable from HackShield update server. If the web server permissions in IIS is set to “Scripts and Executables”, the executable will run on the server, so it cannot be downloaded.

In this case, change the IIS (Internet Information Services) settings. 1. Go to Control Panel > Administrative Tools > IIS (Internet Information Services)

2. Right-click on website > Properties

3. Open Home Directory tab

4. Set permissions to None

Client Application

1. Include HSUpChk.lib file in the project.
2. Include the provided HSUpChk.h file in the source file.

Note

There are two types of HackShield update API. You can select and apply one of the two APIs.

AhnHSUpdate : HackShield basic update feature

AhnHSUpdateEx : HackShield basic update feature + Env file change prevention feature +	Host File Check feature
--	-------------------------

3. Use update API (_AhnHS_HSUpdateEx or _AhnHS_HSUpdate).

```

DWORD      dwRet = 0;
TCHAR      szFullFilePath[MAX_PATH]={0,};

// Specify the path for HackShield folder.
_tcsncpy ( szFullFilePath, _T( "\\HShield" ) );

AHNHS_EXT_ERRORINFO HsExtError={0,};

// Specify the monitoring information.
HsExtError.szServer = "127.0.0.1"    //monitoring address
HsExtError.szUserId = "Test"        //user ID
HsExtError.szGameVersion = "3.0.0.1" //Game version

// Call _AhnHS_HSUpdate function.
dwRet = _AhnHS_HSUpdateEx( szFullFilePath, // HackShield folder path
                           1000 * 600,    // All update timeout
                           1234,          // Game code
                           AHNHSUPDATE_CHKOPT_HOSTFILE|
                           AHNHSUPDATE_CHKOPT_GAMECODE,
                           HsExtError,
                           1000* 20 ); // Server connection timeout

// When using Ex function, you must enter the game code to the env file as the
// HSUpSetEnv.exe setting tool

//.
if( dwRet != ERROR_SUCCESS)
{
    // error
    switch ( dwRet )
    {
        case HSERROR_ENVFILE_NOTREAD:
            ExitClient();
            break;
        case HSERROR_ENVFILE_NOTWRITE:
            ExitClient();
            break;
        case HSERROR_NETWORK_CONNECT_FAIL:
            ExitClient();
            break;
        case HSERROR_HSUPDATE_TIMEOUT:
            ExitClient();
            break;
        case HSERROR_MISMATCH_ENVFILE:
            ExitClient();
            break;
        case HSERROR_HOSTFILE_MODIFICATION:
            ExitClient();
            break;
        ...
    }
}

```

```

    }
}

```

```

DWORD      dwRet = 0;
TCHAR      szFullFilePath[MAX_PATH]={0,};

// Specify the path for HackShield folder.
_tcscpy ( szFullFilePath, _T( ".\\HShield" ) );

// Call _AhnHS_HSUpdate function.
dwRet = _AhnHS_HSUpdate (  szFullFilePath, // HackShield folder path
                           1000 * 600,    // All update timeout
                           1000* 20 );    // Server connection timeout

if( dwRet != ERROR_SUCCESS)
{
    // error
    switch ( dwRet )
    {
        case HSERROR_ENVFILE_NOTREAD:
            ExitClient();
            break;
        case HSERROR_ENVFILE_NOTWRITE:
            ExitClient();
            break;
        case HSERROR_NETWORK_CONNECT_FAIL:
            ExitClient();
            break;
        case HSERROR_HSUPDATE_TIMEOUT:
            ExitClient();
            break;
        ...
    }
}

```

1. Copy all files under [HackShield SDK] \Bin\Win\x86\Update folder to [Game Directory]\HShield folder of the client for distribution.

After copying, the folder file structure is as follows:

Example - Installation directory structure for HackShield update module

[Game Directory]\HShield\

```

3n.mhe
AhnRpt.exe
ahnrpt.ini
AhnUpCtl.dll
AhnUpGS.dll
BldInfo.ini
EHSvc.dll
hshield.dat
HSInst.dll
HSLogMgr.exe
HSUpdate.env
HSUpdate.exe
V3Hunt.dll

```

```
V3InetGS.dll
v3pro32s.dll

[Game Directory]\HShield\asc
0asc.scd
0scure.scd
0sgame.scd
0spe3f.scd
asc_com.dll
asc_dh.dll
asc_fse.dll
asc_intg.dll
asc_mmgr.dll
asc_unp.dll
fse_base.dll
fse_fact.dll
fse_pe.dll
gfs_base.dll
gfs_fact.dll
gfs_file.dll
gfs_mem.dll
gfs_os.dll
gfs_proc.dll
gfs_util.dll
moduler.scd
option.scd
```

2. Execute [HackShield SDK]\Bin\Win\x86\Util\HSUpSetEnv.exe file.

Caution

HSUpSetEnv.exe is a tool which creates an update configuration file. You are not allowed to distribute this tool.

3. Refer to [9.5 HSUpSetEnv Tool](#) to create HSUpdate.env file.
4. Copy HSUpdate.env file to [GameDirecotry]\HShield folder.

Caution

For normal update, the parameter ([Game Directory]\HShield) which is first sent to _AhnHS_HSUpdate function must have all update modules in [HackShield SDK]\Bin\Win\x86\Update.

5. When using Splash image, change the image file name to “splash.jpg”, and then copy it to the [Game Directory]\HShield folder.
6. When changing the HShield Update image on the bottom right side of the screen during update, change the image file name to hsupdate.jpg and copy it

to the [Game Directory]\HShield folder.

Default update image size: 200 * 149. As a progress bar appears under the image, refer to the progress bar when creating update image.



Fig 3-2 Default HackShield Update Image

Note

1. Update image

- _ filename: hsupdate.jpg (not case sensitive)
- _ Size: 200 * 149
- _ Copy to: [Game Directory]\HShield
- _ Description: If hsupdate.jpg exists in the HackShield folder, the image file will be loaded and used as update image. If it does not exist, the default image will be used.

2. Splash image

- _ filename: splash.jpg (not case sensitive)
 - _ Size: 290 * 190
 - _ Copy to: [Game Directory]\HShield
 - _ Description: If splash.jpg exists in the HackShield folder, the splash image will be loaded and appear in the middle of the screen. If it does not exist, no image will be shown.
-

3.3. Application Programming Interface

AhnHS_HSUpdateEx

DESCRIPTION

Updates HackShield files. It checks the HSUpdate.env file and the hosts file.

SYNTAX

```
DWORD __stdcall
_AhnHS_HSUpdateEx(
    LPCTSTR          szUpdateDir
    DWORD            dwTimeout,
    INT64            i64GameCode,
    DWORD            dwOption,
    AHNHS_EXT_ERRORINFO HsExtErrorInfo,
    DWORD            dwTimeoutPerConnection = 0
);
```

PARAMETERS

Parameter	Value	Description
szUpdateDir	LPCTSTR	Folder with the update file installed
dwTimeout	DWORD (milliseconds)	Amount of time to wait for update. Set as INFINITE when this value is 0. * The recommended value is 600000 (10 minutes). Do not set to 0 unless absolutely necessary.
i64GameCode	INT64	Enter the saved game code value in the HSUpdate.env file by using HSUpSetEnv.exe tool. (mandatory when the monitoring function is used.)
dwOption	DWORD	Additional update feature option AHNHSUPDATE_CHKOPT_HOSTFILE AHNHSUPDATE_CHKOPT_GAMECODE
HsExtErrorInfo	AHNHS_EXT_ERRORINFO	Structure with the server URL address, user ID, and game version

dwTimeOutPer Connection (milliseconds)	DWORD	<p>Amount of time to wait for a connection to the server.</p> <p>* Set to reasonable value depending on network status.</p> <p>* When set to 0, the timeout feature will not operate during server connection.</p>
--	-------	--

szUpdateDir

[in] szUpdateDir parameter shall set the absolute paths of the distributed HackShield-related files. The folder shall have all update modules. It is recommended that GetModuleFileName function should be used for proper setting of the absolute path. GetCurrentDirectory function may not get a desired path.

dwTimeOut

Amount of time to wait for after calling [in] function. The second argument is in millisecond. In case no response arrives during this time, the function call is failed. It is recommended to set 10 minutes for standby. In case update fails, allow or disallow the game execution depending on the game developer's policy.

i64GameCode

A parameter used to set with (HSUpdate.env) update module so that [in] update configuration file is not allowed to be changed. Enter the game code saved in the update environment file, and if the game code is different or if there is no code, HSERROR_MISMATCH_ENVFILE error will be returned.

This code is used for the monitoring server to identify the game, so that it must be entered to use the monitoring function.

dwOption

Option to define the feature performed in [in] _AhnHS_HSUpdateEx function.

AHNHSUPDATE_CHKOPT_HOSTFILE: If the hosts file contains an IP address specified for the update servers, the HSERROR_HOSTFILE_MODIFICATION error will be returned. It prevents the vulnerability which can be exploited by malicious people to bypass update procedure.

AHNHSUPDATE_CHKOPT_GAMECODE: Compares the game code saved in the HSUpdate.env file and i64GameCode passed as a parameter. If they do not match, HSERROR_MISMATCH_ENVFILE error is returned.

HsExtErrorInfo

A structure which includes the information on the [in] server IP, user account, and game version.

In the szServer member, the current HackShield monitoring server IP is included. In the szUserID member, the user account information is included. In the szGameVersion member, the client version is included.

As applying the monitoring service to the HackShield update, the above information must be included in the structure. If the monitoring service is not applied to the HackShield update, it is required to initialize the argument to ZeroMemory only.

dwTimeOutPerConnection

Amount of time to wait for a connection to the [in] server. The third factor is in milliseconds, and if there is no response during the specified period, the connection will be failed.

When setting more than one servers in HSUpdate.env with HSUpSetEnv.exe, the dwTimeOutPerConnection value will apply the same for each server.

When set to '0', the timeout during server connection will not be checked.

RETURN VALUE

HACKSHIELD_ERROR_SUCESS (Value = 0x00000000)

- Description: Successful update
- Cause: Normal.
- Workarounds:

HSERROR_ENVFILE_NOTREAD (Value = 0x30000010)

- Description: Cannot read HSUpdate.env file.
- Cause: There is no env configuration file or HSUpSetEnv.exe tool version is not compatible.
- Workarounds:
 - ① Check whether env file exists in the location.
 - ② Check whether HSUpSetEnv.exe tool is the latest version that exists within the sdk.

HSERROR_ENVFILE_NOTREADFOUND (Value = 0x30000011)

- Description: HSUpdate.env file does not exist.
- Cause: env file does not exist or the user does not have access rights to this file.
- Workarounds:
 - ① Check if the env file exists in the HShield folder.

HSERROR_UPDATE_INITIALIZE_FAILED (Value = 0x3000001C)

- Description: Module Initialization failed.
- Cause: An error has occurred when initializing HSUpdate.exe.
- Workarounds: It could be caused by an internal update error.
Please contact AhnLab, Inc.

HSERROR_ENVFILE_NOTWRITE (Value = 0x30000020)

- Description: Cannot write HSUpdate.env file.
- Cause: There is no read property or access permission when creating

HSUpdate.env file.

- Workarounds:
 - ① Check whether env file attribute has access permission.
 - ② When you delete the file and create a new one, check whether the above problem has occurred.

HSERROR_NETWORK_CONNECT_FAIL (Value = 0x30000030)

- Description: Cannot connect to the server.
- Cause: The update server (ftp/http) connection is unavailable.
- Workarounds:
 - ① Check the connection through ftp tool or IE whether the update server is connected properly.
 - ② Check whether the update address and ID/password in the Env file are correct.
 - ③ Some worms may restrict connection to security program site. Scan your system with an anti-virus program.
 - ④ The update server cannot read the *.ui and the file cannot be downloaded from the update server.
 - Check whether the update server can read *.ui file extension.

HSERROR_LIB_NOTEDIT_REG (Value = 0x30000050)

- Description: An error occurred while inputting the network result.
- Cause:
- Workarounds:

HSERROR_NOTFINDFILE (Value = 0x30000060)

- Description: Cannot file HackShield update program-related files.
- Cause:
- Workarounds:

HSERROR_PROTECT_LISTLOAD_FAIL (Value = 0x30000070)

- Description: Cannot file HSUpdate.pt authentication file.
- Cause:
- Workarounds:

HSERROR_HSUPDATE_TIMEOUT (Value = 0x30000090)

- Description: Update failed due to timeout.
- Cause: A timeout can be set in _AhnHS_HSUpdate function. An error will occur if update is not performed within the specified period. It can also occur when there is a problem in the network.
- Workarounds:

HSERROR_STRING_CONVERSION_FAILED (Value = 0x300000B0)

- Description: Unicode string conversion failed.
- Cause: The string conversion by Unicode supporting API function has failed.
- Workarounds: Check the szUpdateDir passed as an argument.

HSERROR_MISMATCH_ENVFILE (Value = 0x300000C0)

- Description: Game Client and the update configuration file do not match.
- Cause: i64GameCode sent when calling _AhnHS_HSUpdateEx function is different from GameCode saved in the HSUpdate.env file, or the game code is not saved in the HSUpdate.env file.
- Workarounds: Check if the game code saved in the HSUpdate.env file and i64GameCode passed as a parameter are identical.

The feature can be turned ON/OFF with AHNHSUPDATE_CHKOPT_GAMECODE option.

HSERROR_HOSTFILE_MODIFICATION (Value = 0x300000D0)

- Description: The HackShield update URL is included in the hosts file.
- Cause: The hosts file contains an IP address for the update server specified in the HSUpdate.env file.
- Workarounds: Check whether the HackShield update URL is included in the hosts file. The feature can be turned ON/OFF with AHNHSUPDATE_CHKOPT_HOSTFILE option.

HSERROR_AUTOUPDATE_FAIL (Value = 0x300000E0)

- Description: An error has occurred in the HackShield automatic update process.
- Cause: It could be caused by an internal update error. Please contact AhnLab, Inc.
- Workarounds:

HSERROR_UPDATE_WIN32_ERROR (Value = 0x3000FFFF)

- Description: An error has occurred in the HackShield automatic update process.
- Cause: It could be caused by an internal update error. Please contact AhnLab, Inc.
- Workarounds:

REMARKS

Update function shall be called before the initialization function (_AhnHS_Initialize) is called. Can be called by an executable file like a game process launcher separately from the HackShield function.

AhnHS_HSUpdate

DESCRIPTION

Updates HackShield files.

SYNTAX

```
DWORD __stdcall  
_AhnHS_HSUpdate(  
    LPCTSTR szUpdateDir,  
    DWORD dwTimeOut,  
    DWORD dwTimeOutPerConnection = 0  
);
```

PARAMETERS

Parameter	Value	Description
szUpdateDir	LPCTSTR	Folder with the update file installed
dwTimeOut	DWORD (milliseconds)	Amount of time to wait for update. Set as INFINITE when this value is 0. * The recommended value is 600000 (10 minutes). Do not set to 0 unless absolutely necessary.
dwTimeOutPerConnection	DWORD (milliseconds)	Amount of time to wait for a connection to the server. * Set to reasonable value depending on network status. * When set to 0, the timeout feature will not operate during server connection.

szUpdateDir

[in] szUpdateDir parameter shall set the absolute paths of the distributed HackShield-related files. The folder shall have all update modules. It is recommended that GetModuleFileName function should be used for proper setting of the absolute path. GetCurrentDirectory function may not get a desired path.

dwTimeOut

Amount of time to wait for after calling [in] function. The second argument is in millisecond. In case no response arrives during this time, the function call is failed. It is recommended to set 10 minutes for standby. In case update fails, allow or disallow the game execution depending on the game developer's policy.

dwTimeOutPerConnection

Amount of time to wait for a connection to the [in] server. The third factor is in milliseconds, and if there is no response during the specified period, the connection will be failed.

When setting more than one servers in HSUpdate.env with HSUpSetEnv.exe, the dwTimeOutPerConnection value will apply the same for each server.

When set to '0', the timeout during server connection will not be checked.

RETURN VALUE

HACKSHIELD_ERROR_SUCESS (Value = 0x00000000)

- Description: Successful update
- Cause: Normal.
- Workarounds:

HSERROR_ENVFILE_NOTREAD (Value = 0x30000010)

- Description: Cannot read HSUpdate.env file.
- Cause: There is no env configuration file or HSUpSetEnv.exe tool version is not compatible.
- Workarounds:
 - ① Check whether env file exists in the location.
 - ② Check whether HSUpSetEnv.exe tool is the latest version that exists within the sdk.

HSERROR_ENVFILE_NOTWRITE (Value = 0x30000020)

- Description: Cannot write HSUpdate.env file.
- Cause: There is no read property or access permission when creating HSUpdate.env file.
- Workarounds:
 - ③ Check whether env file attribute has access permission.
 - ④ When you delete the file and create a new one, check whether the above problem has occurred.

HSERROR_NETWORK_CONNECT_FAIL (Value = 0x30000030)

- Description: Cannot connect to the server.

- Cause: The update server (ftp/http) connection is unavailable.
- Workarounds:
 - ① Check the connection through ftp tool or IE whether the update server is connected properly.
 - ② Check whether the update address and ID/password in the Env file are correct.
 - ③ Check the connection through ftp tool or IE whether the update server is connected properly.
 - ④ Check whether the update address and ID/password in the Env file are correct.
 - ⑤ Some worms may restrict connection to security program site. Scan your system with an anti-virus program.
 - ⑥ The update server cannot read the *.ui and the file cannot be downloaded from the update server.
 - Check whether the update server can read *.ui file extension.

HSERROR_LIB_NOTEDIT_REG (Value = 0x30000050)

- Description: An error occurred while inputting the network result.
- Cause:
- Workarounds:

HSERROR_NOTFINDFILE (Value = 0x30000060)

- Description: Cannot file HackShield update program-related files.
- Cause:
- Workarounds:

HSERROR_PROTECT_LISTLOAD_FAIL (Value = 0x30000070)

- Description: Cannot file HSUpdate.pt authentication file.
- Cause:
- Workarounds:

HSERROR_HSUPDATE_TIMEOUT (Value = 0x30000090)

- Description: Update failed due to timeout.
- Cause: A timeout can be set in _AhnHS_HSUpdate function. An error will occur if update is not performed within the specified period. It can also occur

when there is a problem in the network.

- Workarounds:

HSERROR_AUTOUPDATE_FAIL (Value = 0x300000E0)

- Description: An error has occurred in the HackShield automatic update process.
 - Cause: It could be caused by an internal update error.
Please contact AhnLab, Inc.
 - Workarounds:

REMARKS

Update function shall be called before the initialization function (_AhnHS_Initialize) is called. Can be called by an executable file like a game process launcher separately from the HackShield function.

4. Extended Detection

Server-side

4.1. Overview

Extended Server-side Detection (AntiCpX) is an advanced version of the server-side SDK which provided file/memory manipulation detection functions. The most significant improvement is that new version of server-side detection protects the entire code area while the previous version protected only the code loaded on the memory for each function.

Functions

Extended Server-side Detection (AntiCpX) includes functions of the previous version and provides enhanced “memory integrity function.”

Game client file integrity check

Apply AntiCpXSvr.dll (libanticpxsvr.so, libanticpxsvr_st.a) and AntiCpXSvr.h in the server, and HShield.lib and HShield.h, HackShield modules, in the client. The game developer can check the manipulation status of the game executable file in real time through the communication between the server and the client.

Caution

It can be detected by extended server-side detection when the game client is infected by a virus. Please end the callback according to the company policy.

Packet Integrity

In order to prevent hacking attacks which captures packets and generates packets in certain circumstances, a separate module which guarantees packet integrity operates. The game developer can block packet capturing and creation on the network. However, this function protects only the messages related to the prevention of the server interface crack.

Operational State of HackShield

Apply AntiCPXSvr.dll (libanticpxsvr.so, libanticpxsvr_st.a) and AntiCpXSvr.h in the server, and HShield.lib and HShield.h, HackShield modules, in the client. The game developer can easily check the operational status of HackShield through the communication between the server and the client. The server shall send a query message only after HackShield starts.

Memory Integrity (Extended Server Interface)

Apply AntiCpXSvr.dll (libanticpxsvr.so, libanticpxsvr_st.a) and AntiCpXSvr.h in the server, and HShield.lib and HShield.h, HackShield modules, in the client. The

game developer can check the memory manipulation status of the game process in real time through the communication between the server and the client. Extended server-side detection protects the entire code area while the previous version protected only the code loaded on the memory for each function.

HackShield Engine Integrity Verification

Apply AntiCPXSvr.dll (libanticpxsvr.so, libanticpxsvr_st.a) and AntiCpXSvr.h, in the server, and HShield.lib and HShield.h, HackShield modules, in the client. The game developer can check the integrity of heuristic engine (3n.mhe) file through the communication between the server and the client.

Features

Interface Function (API)

Interface DLLs and a library are provided for the game developer to use the functions of the extended server-side detection (AntiCpX) and check the return values. Using the provided interface DLL and library, the developer can check the integrity of the game file and operational state of HackShield.

HSB data file information generation program

HSBGen.exe provided by the extended server interface (AntiCpX) creates and stores file data and memory information for the client program to check file/memory integrity through the communication between the client and the server. The game developer can store file data and memory information in the server and use the file/memory crack prevention function through server-side detection by referring to file data and memory information.

Extended Server-side Detection Files

WBin\Win\[Platform]\AntiCrack\AntiCpXSvr.dll : Dll file for extended server-side detection.

WBin\Win\x86\AntiCrack\HSBGen.exe : hsb file creation tool

WBin\Win\x86\AntiCrack\HSPub.key : Server interface authentication key file

WBin\Win\x86\HShield\3n.mhe : Heuristic engine version management file

WBin\Win\x86\HShield\hshield.dat : HackShield version management file

WInclude\AntiCpXSvr.h : Extended server-side header file

WLib\Win\x86\AntiCpXSvr.lib : Library files for extended server-side detection (for Windows 32 Bit)

WLib\Win\x64\AntiCpXSvr.lib : Library files for extended server-side detection (for Windows 64 Bit)

WBin\Linux\[Platform]\AntiCrack\libanticpxsvr.so: Extended server-side dynamic library file (for Linux)

WBin\Solaris\x86\AntiCrack\libanticpxsvr.so: Dynamic library files for extended server-side detection (for Solaris)

WLib\Linux\[Platform]\libanticpxsvr_st.a : Extended server-side static library file (for Linux)

WLib\Solaris\x86\AntiCrack\libanticpxsvr_st.a : Extended server-side static library file (for Solaris)

※ The file for Solaris supports 32bit x86 platform only.

System Architecture

Extended Server Interface (AntiCpX) provides AntiCpXSvr.dll (libanticpxsvr.so, libanticpxsvr_st.a), HShield.lib, and EHSvc.dll is applied as DLL and library in the server and the client. General architecture and operating principles of the extended server-side detection (AntiCpX) are as follows:

AntiCpXSvr.dll (Interface dll)

Dynamic library file for Windows.

Used by the server. Provides an API which creates request messages and checks client file and memory status based on the response message from the client.

AntiCpXSvr.lib

Static dynamic library file for Windows.

Used by the server. Provides an API which creates request messages and checks client file and memory status based on the response message from the client.

libanticpxsvr.so

Dynamic library file for Linux and Solaris.

Used by the server. Provides an API which creates request messages and checks client file and memory status based on the response message from the client.

libanticpxsvr_st.a

Static library file for Linux and Solaris.

Used by the server. Provides an API which creates request messages and checks client file and memory status based on the response message from the client.

HShield.lib (HackShield library)

Used by the client. Provides an API which receives request messages from the server, encrypts the response data, and sends the encrypted response to the server.

HSBGen.exe (File information creation program)

Creates file data and memory information for the server in order to check the integrity of the game file and the memory integrity of the game. Different file data created for each file creation. If there are multiple servers, each server shall have different file data for greater security.

AntiCpx.hsb (Client CRC File)

Created by HSBGen.exe. Designed to guarantee integrity for the client program.

HSPub.key (Server Interface Authentication Key File)

Used to authenticate the client to access HackShield. Shall be stored in the same folder where AntiCpx.hsb file is stored.

3n.mhe (Heuristic engine File)

An engine file used by the client. When stored in the same folder where AntiCpX.hsb file of the server is stored, the client which uses the previous-version 3n.mhe engine will be disconnected.

hshield.dat (HackShield Version File)

A data file used by the client. When stored in the same folder where AntiCpX.hsb file of the server is stored, the client which uses the previous-version hshield.dat file will be disconnected.

Note

When 3n.mhe or hshield.dat files are stored in the same folder where the server hsb file is stored, the client which uses the previous-version HackShield will be disconnected. (You do not need to reset the server when uploading 3n.mhe & hshield.dat files.)

When “Disconnected” appears as the table below, ANTICPX_RECOMMEND_CLOSE_SESSION (Error: ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION/ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION) will be returned from the _AhnHS_VerifyResponseEx function.

When the error is returned, disconnect the client connection.

Table 4-1 Server interaction version management

File name	Version	Connection
3n.mhe	Client < Server	Disconnected
	Client ≥ Server	Connected

hshield.dat	Client < Server	Disconnected
	Client ≥ Server	Connected

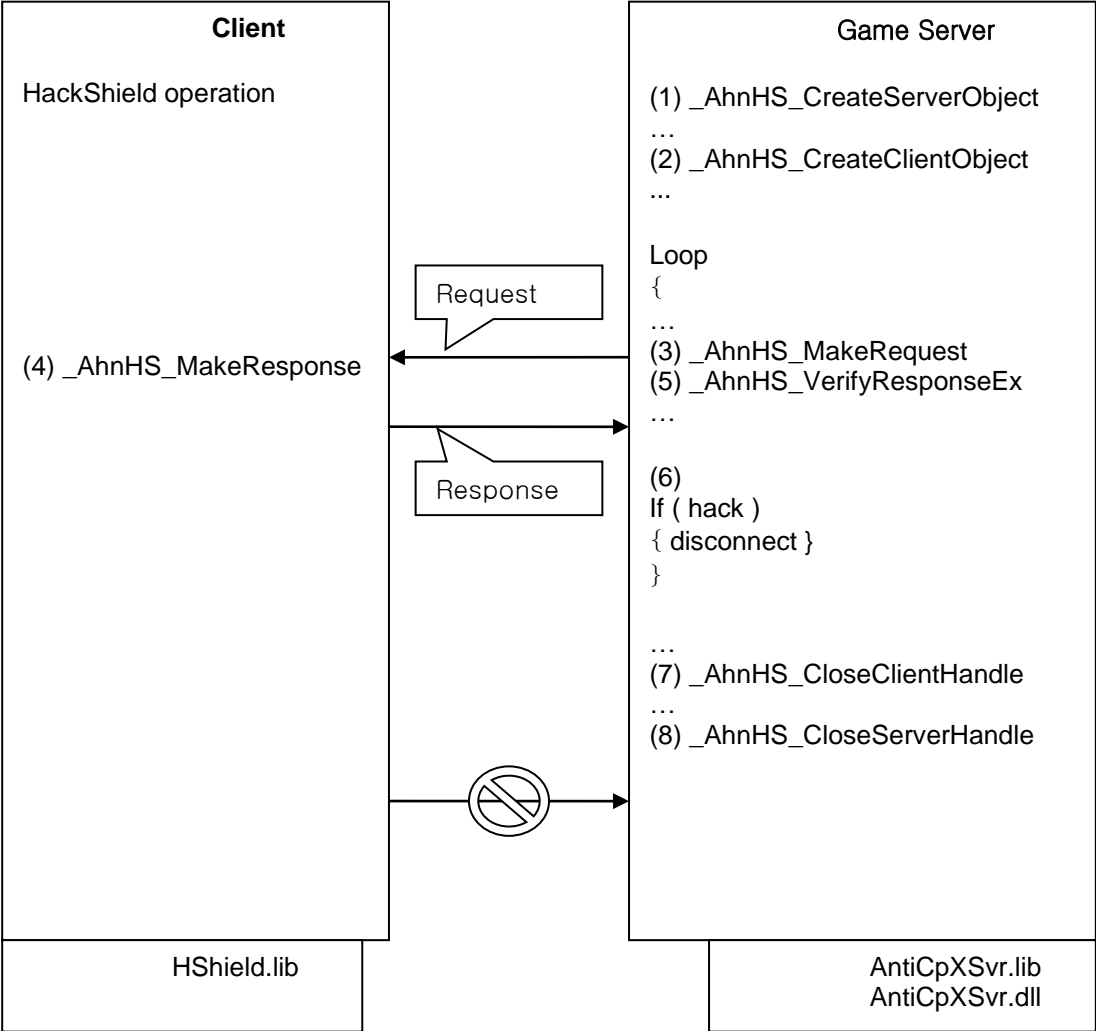


Fig. 4-1 AntiCpX operation

When the game server is first executed, `AHNHS_SERVER_HANDLE` will be created by `_AhnHS_CreateServerObject` function. Then, whenever a client is accessed, `_AhnHS_CreateClientObject` function and server handle are collected as parameters and `AHNHS_CLIENT_HANDLE` is created. In order to check the client manipulation status, `_AhnHS_MakeRequest` function regularly creates and sends request messages.

The client creates a proper response message suitable for the request message of the server. The response message is created by `_AhnHS_MakeResponse` function. The validity of the response message of the client is verified by `_AhnHS_VerifyResponseEx` function.

Caution

Make sure `_AhnHS_MakeRequest` is not called until `_AhnHS_VerifyResponseEx` is called by sending the response message created by `_AhnHS_MakeResponse` function to the server.

Create dump file when an exception occurs

You can create a dump file when an exception occurs while performing a feature in the extended server API. The feature can be performed by setting the registry value. The feature is set to create no dump file by default.

(Caution) Only available in Windows OS.

Within the `HKLM\SOFTWARE\AhnLab\HShield\Dump` key:

- `DumpOnOff`
- `DumpType`
- `DumpPath`

`HKLM\SOFTWARE\AhnLab\HShield\Dump` is the key created when an exception occurs, and `DumpOnOff`, `DumpType` and `DumpPath` keys must be directly created and entered.

`DumpOnOff` is "DWORD", and a dump file can be remained only if the value is '0x1'. If the value is '0x0', a dump file is not remained.

`DumpType` value is "DWORD". You can refer to and set as the value defined by `MINIDUMP_TYPE` (Refer to MSDN Library). To analyze the exception accurately, it is recommended to set 'MiniDumpWithFullMemory' to '0x2'. If the dump file size becomes larger, and takes up more disk space, change the value to '0x1', which is 'MiniDumpNormal'.

`DumpPath` is "string value". Specify the directory to create the dump file. The location must be accessible.

(Caution) If `DumpPath` is not created, or if the location is inaccessible, or incorrect, the dump file will remain where the current game server was executed.

4.2. Application Programming

This chapter describes how to check file and memory integrity using the APIs provided by the extended server-side detection (AntiCpX).

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

Programming Application

Do the following using the extended server-detection (AntiCpX) before starting programming:

4.2.1.1. AntiCpXSvr-related File

AntiCpXSvr-related File

Table 4-2 AntiCpXSvr-related File

File name	Installation folder	Description
AntiCpXSvr.h	[Program source folder]	Header file to be used by the server
AntiCpXSvr.lib	[Program source folder]	Import library of AntiCPXSvr.dll (For Windows)
AntiCpXSvr.dll	[Program execution folder]	DLL file to be used in the server
Libanticpxsvr.so	[Program execution folder]	Extended server module for Linux/Solaris (Dynamic library file)
Libanticpxsvr_st.a	[Program execution folder]	Extended server module for Linux/Solaris (Static library file)
AntiCpx.hsb	[AhnHS_CreateServerObject API parameter folder]	Client integrity verification file
HSPub.key	[Same folder as HSB file]	Server Interface Authentication Key File
3n.mhe	[Same folder as HSB file]	Heuristic engine version management file
hshield.dat	[Same folder as HSB file]	HackShield version management file
HShield.h	[Program source folder]	Header file to be used in the client. HackShield functions included.
HShield.lib	[Program source folder]	Library file to be used in the client. HackShield function included.
EHSvc.dll	[Program source folder]	dll file to be used in the client.

		HackShield functions included.
--	--	--------------------------------

4.2.1.2. Application

Server Application

1. Create AntiCpx.hsb using HSBGen.exe.
((See [10.5.HSBGen Tool.](#))
2. Copy HSPub.key file to the folder where AntiCpx.hsb file is stored.
3. Copy 3n.mhe and hshield.dat files in the same folder as hsb file. (This process is not mandatory, but recommended for HackShield version management.)

Note

Upload 3n.mhe and hshield.dat files used by the client in the folder the hsb is located to disallow connection by a client with the wrong 3n.mhe and hshield.dat version.

4. When the game server is first executed, AHNHS_SERVER_HANDLE will be created by _AhnHS_CreateServerObject function. This handle shall be kept until the game server is terminated. (Note: The server handle is used to make a client handle in the following two steps.)
5. Each time a client accesses the server, AHNHS_CLIENT_HANDLE will be created by _AhnHS_CreateClientObject function with the server handle used as a parameter. This handle shall be maintained during the session in which the client is connected to the network. (Note: The client handle is used to create a request message in the following three steps.)
6. In order to monitor client manipulation, a request message is created and sent. The request message is created by _AhnHS_MakeRequest function with the client being used as a client handle.
7. The client will create a proper response message to the request message of the server. The response message is created by _AhnHS_MakeResponse function.
8. Check whether the response message of the client is valid. The validity of the response message is checked by _AhnHS_VerifyResponseEx function.
9. If ANTICPX_RECOMMAND_CLOSE_SESSION occurs in _AhnHS_VerifyResponseEx function, stop game client connection after handling the error according to the error value received as an argument.

Caution

(Apply _AhnHS_VerifyResponseEx or _AhnHS_VerifyResponse.)

10. When the client is disconnected, the client handle created in Step 2 will be closed.
11. When the server process is terminated, the server handle created in Step 1 shall be closed (with the client handle being closed.)

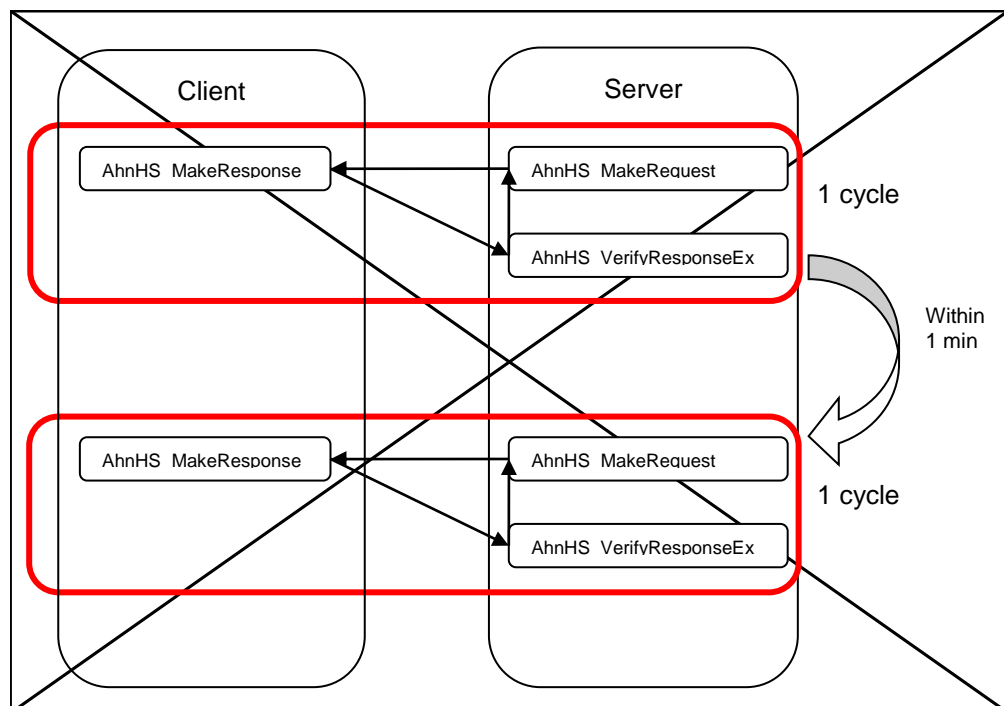
Client Application

1. Include HShield.lib file in the project.
2. Include the provided HShield.h file in the source file.
3. Receive an encrypted version request message from the server.
4. Call `_AhnHS_MakeResponse` and create an encrypted version response message to be sent to the server.
5. Send the encrypted response message to the server.

Server interface cycle

Description: A cycle is for the time taken in the following actions; the server sends a request message, the client creates the response message and sends the message to the server, and then the server receiving the response message.

Cycle It is recommended to set the cycle to within 1 minute.
(You can set the cycle according to the game characteristics, but the longer the cycle, the longer the hacking detection time.)



4.3. Application Programming Interface

_AhnHS_CreateServerObject

DESCRIPTION

Creates a server handle by loading .hsb file created by HSBGen.exe. Usually, one server handle is created for the server process which services one game, and the server handle is maintained till the game server process is terminated.

SYNTAX

```
AHNHS_SERVER_HANDLE __stdcall  
_AhnHS_CreateServerObject (  
    IN const char *pszFilePath  
);
```

PARAMETERS

Parameter	Value	Description
pszFilePath	const char *	HackShield Briefcase (.hsb) file Full Path

RETURN VALUE

In case a server handle is not normally created, ANTICPX_INVALID_HANDLE_ will be returned. It may occur in the following cases:

- ① When the path of HackShield Briefcase (.hsb) file is not correct or the file does not exist.
- ② HSPub.key file does not exist in the path
- ③ System resources (memory) is insufficient.
- ④ The HSB file does not exist, or there is no file access permission.
- ⑤ If an exception occurs while executing _AhnHS_CreateServerObject function, MiniDump file will be created. But, it is only created in Windows.
The dump file is created with the name, 'CREATESERVEROBJECT_year_month_date hour-minute-second.dmp'.
Send the file to AhnLab, Inc.)

Example

The following is an example of calling _AhnHS_CreateServerObject function

Example

```
// The file path must contain '/', not '\'.
// The path shall include the file name.

strcpy(g_szHsbFilePath,
       "C:\\GameServer\\HShield\\anticpx.hsb" );

hServer = _AhnHS_CreateServerObject (g_szHsbFilePath);

if ( hServer == ANTICPX_INVALID_HANDLE_VALUE )
{
    // error
}
```

_AhnHS_CloseServerHandle

DESCRIPTION

Closes the server handle.

SYNTAX

```
void __stdcall  
_AhnHS_CloseServerHandle (  
    IN AHNHS_SERVER_HANDLE hServer  
);
```

PARAMETERS

Parameter	Value	Description
pszFilePath	AHNHS_SERVER_HANDLE	Handle created by _AhnHS_CreateServerObject function.

RETURN VALUE

None.

Example

The following is an example of calling _AhnHS_CloseServerHandle function.

Example

```
_AhnHS_CloseServerHandle ( hServer );
```

_AhnHS_CreateClientObject

DESCRIPTION

Receives input of the server handle, and creates a client handle. A client handle will be created each time the client accesses. The client handle is maintained and recycles while the session is maintained.

SYNTAX

```
AHNHS_CLIENT_HANDLE  
_AhnHS_CreateClientObject (  
    IN AHNHS_SERVER_HANDLE hServer  
);
```

PARAMETERS

Parameter	Type	Description
hServer	AHNHS_SERVER_HANDLE	Server handle

RETURN VALUE

Client handle

In case a server handle is not normally created, ANTICPX_INVALID_HANDLE_ will be returned. It may occur in the following cases:

- ① If the hServer argument is invalid.
Check whether the hServer server handle has been properly created through _AhnHS_CreateServerObject function.
- ② If an exception occurs while executing _AhnHS_CreateServerObject function, MiniDump file will be created. But, it is only created in Windows.
The dump file is created with the name, 'CREATESERVEROBJECT_year_month_date hour-minute-second.dmp'.
Send the file to AhnLab, Inc.)

Example

The following is an example of calling _AhnHS_CreateClientObject function.

Example

```
hClient = _AhnHS_CreateClientObject ( hServer );  
if ( hClient == ANTICPX_INVALID_HANDLE_VALUE )
```

```
{  
    // error  
}
```

_AhnHS_CloseClientHandle

DESCRIPTION

The created client handle shall be closed when the client session is terminated. At this time, memory and system resources allocated for the client handle shall be also released.

SYNTAX

```
void __stdcall  
_AhnHS_CloseClientHandle (  
    IN AHNHS_CLIENT_HANDLE hClient  
);
```

PARAMETERS

Parameter	Type	Description
hClient	AHNHS_CLIENT_HANDLE	Client handle

RETURN VALUE

None.

Example

The following is an example of calling `_AhnHS_CloseClientHandle` function.

```
Example  
  
_AhnHS_CloseClientHandle ( hClient );
```

_AhnHS_MakeRequest

DESCRIPTION

Creates a request message by inputting a client handle for the current session. The request message is displayed as `AHNHS_TRANS_BUFFER` structure, and the member parameters are as follows:

```
typedef struct _AHNHS_TRANS_BUFFER
{
    unsigned short nLength;
    unsigned char byBuffer[ANTICPX_TRANS_BUFFER_MAX]; // Maximum
    buffer size for the packets

} AHNHS_TRANS_BUFFER, *PAHNHS_TRANS_BUFFER;
```

`nLength` ; Buffer length used for the creation of the request message
`byBuffer` ; Maximum byte buffer which may be used for the creation of the request message

Caution

`byBuffer` indicates the maximum buffer sizes available for the creation of the request message. Data shall be sent on the network by `nLength`.

SYNTAX

```
unsigned long __stdcall
_AhnHS_MakeRequest (
    IN AHNHS_CLIENT_HANDLE hClient,
    OUT PAHNHS_TRANS_BUFFER pRequestBuffer
);
```

PARAMETERS

Parameter	Type	Description
<code>hClient</code>	<code>AHNHS_CLIENT_HANDLE</code>	Client handle
<code>pRequestBuffer</code>	<code>PAHNHS_TRANS_BUFFER</code>	Data buffer/length to send

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned when the function is successfully called.

- Cause: Normal
- Workarounds:

ERROR_ANTICPXSVR_INVALID_PARAMETER (Value = 0xE9040003)

- Description: Incorrect input data.
- Cause: hClient and pRequestBuffer values are NULL.
- Workarounds: Check if hClient and pRequestBuffer are not NULL.

ERROR_ANTICPXSVR_BAD_FORMAT (Value = 0xE9040004)

- Description: HSB file reading failed.
- Cause: HSB file has not been created properly.
- Workarounds: Check whether the latest HSBGen.exe tool has been used, and create HSB file again.

ERROR_ANTICPXSVR_NOT_YET_RECEIVED_RESPONSE (Value = 0xE9040005)

- Description: No response to the request message has arrived.
- Cause: _AhnHS_MakeRequest function is called in the following ways: 1) Calls _AhnHS_MakeRequest function. 2) Receives Ack message from the client. 3) Calls AhnHS_VerifyResponse function. 4) Calls _AhnHS_MakeRequest function again. If the synchronization is not successful, a trouble may occur.
- Workarounds: Check if there is any synchronization issue in the client session management.

ERROR_ANTICPXSVR_NOT_ENOUGH_MEMORY (Value = 0xE9040007)

- Description: Insufficient memory space.
- Cause: There is not enough memory in the server.
- Workarounds: Check whether there is memory leak in the server.

ERROR_ANTICPXSVR_BAD_MESSAGE (Value = 0xE9040008)

- Description: Buffer encryption failed.
- Cause: Encryption of the buffer to send a request has failed.
- Workarounds: The above error may be caused by the system architecture issue. Please contact AhnLab, Inc.

ERROR_ANTICPXSVR_MAKEREQ_EXCEPTION (Value = 0xE9040014)

- Description: An exception occurred while executing _AhnHS_MakeRequest function.

(This error can be converted only in module for Windows.)

- Cause: An exception could be caused by various problems.
- Workarounds: MiniDump file will be created.
The dump file is created with the name, 'MAKEREQUEST_year_month_date hour-minute-second.dmp'. Send the file to AhnLab, Inc.

Example

The following is an example of calling _AhnHS_MakeRequest function.

Example

```
AHNHS_TRANS_BUFFER stReqTransBuf;

ulRet = _AhnHS_MakeRequest ( hClient, &stReqTransBuf );

if ( ulRet != ERROR_SUCCESS )
    return ulRet;

bytesSent = send( ConnectSocket, stReqTransBuf.byBuffer,
stReqTransBuf.nLength, 0 );

...
```

Caution

For hack detection, it is recommended to call _AhnHS_MakeRequest right after the user connects to the game server.

_AhnHS_VerifyResponseEx

DESCRIPTION

Checks the client response to the request message issued by `_AhnHS_MakeRequest` function.
(`_AhnHS_VerifyResponse()` function can be called internally.)

SYNTAX

```
unsigned long __stdcall  
_AhnHS_VerifyResponseEx (  
    IN AHNHS_CLIENT_HANDLE hClient,  
    IN unsigned char *pbyResponse,  
    IN unsigned long nResponseLength,  
    OUT unsigned long *pnErrorCode  
);
```

PARAMETERS

Parameter	Type	Description
hClient	AHNHS_CLIE NT_HANDLE	Client handle
pbyResponse	char *	Data buffer received from the client
nResponseLength	unsigned long	Data length received from the client
pnErrorCode	Unsigned long	Return value from _AhnHS_VerifyResponse() function

RETURN VALUE

ANTICPX_RECOMMAND_CLOSE_SESSION (Value = 101)

- Description: Hacking has been detected in the client, you must end the connection with the game client from the game server.
(Hacking mainly means return of error recommended to disconnect from the client, among error codes described in the `_AhnHS_VerifyResponsef()` function.)

Internally,

`_AhnHS_VerifyResponse` function is called. If the return value of the function is one of the followings, `ANTICPX_RECOMMAND_CLOSE_SESSION` is returned.

- `ERROR_ANTICPXSVR_BAD_MESSAGE`
- `ERROR_ANTICPXSVR_REPLY_ATTACK`
- `ERROR_ANTICPXSVR_UNKNOWN_CLIENT`
- `ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK`
- `ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK`
- `ERROR_ANTICPXSVR_MEMORY_ATTACK`
- `ERROR_ANTICPXSVR_OLD_VERSION_CLIENT_EXPIRED`

ERROR_ANTICPXSVR_NANOENGINE_FILE_ATTACK
 ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION
 ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION
 ERROR_ANTICPXSVR_VERIFY_EXCEPTION
 - ERROR_ANTICPXSVR_INVALID_PARAMETER
 ERROR_ANTICPXSVR_ABNORMAL_HACKSHIELD_STATUS
 ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED

- Cause: Hacking has been detected from the client.
- Workarounds: None

ANTICPX_RECOMMAND_KEEP_SESSION (Value = 102)

- Description: The response from the client is Normal. Continue the connection with the game client from the game server.
- Cause: None
- Workarounds: None

ERROR_ANTICPXSVR_VERIFYEX_EXCEPTION (Value = 0xE904001A)

- Description: An Exception has occurred when executing _AhnHS_VerifyResponseEx function

(This error can be converted only in module for Windows.)

- Cause: An exception could be caused by various problems.
- Workarounds: MiniDump file will be created. The dump file is created with the name, 'VERIFYRESPONSEEX_year_month_date hour-minute-second.dmp'. Send the file to AhnLab, Inc.

Example

The following is an example of calling _AhnHS_VerifyResponse function.

Example

```

DWORD dwError = 0;
ulRet = _AhnHS_VerifyResponseEx ( hClient,
                                stResponseBuf.byBuffer,
                                stResponseBuf.nLength,
                                &dwError );

if ( ulRet == ANTICPX_RECOMMAND_CLOSE_SESSION )
{
    Log ("[AVREx] Disconnect the Client: 0x%x", dwError );
    // Client Out -> _AhnHS_CloseClientHandle
    // and close the hClient handle
}
  
```

_AhnHS_VerifyResponseEx_WithInfo

DESCRIPTION

Checks the client response to the request message issued by _AhnHS_MakeRequest function.
(_AhnHS_VerifyResponse() function can be called internally.)

SYNTAX

```
unsigned long __stdcall
_AhnHS_VerifyResponseEx_WithInfo (
    IN AHNHS_CLIENT_HANDLE hClient,
    IN unsigned char *pbyResponse,
    IN unsigned long nResponseLength,
    OUT unsigned long *pnErrorCode,
    OUT unsigned long *pnSpecificError
);
```

PARAMETERS

Parameter	Type	Description
hClient	AHNHS_CLIE NT_HANDLE	Client handle
pbyResponse	char *	Data buffer received from the client
nResponseLength	unsigned long	Data length received from the client
pnErrorCode	unsigned long*	Return value from _AhnHS_VerifyResponse() function
pnSpecificError	unsigned long*	Error code to replace pnErrorCode value

RETURN VALUE

ANTICPX_RECOMMEND_CLOSE_SESSION (Value = 101)

- Description: Hacking has been detected in the client, you must end the connection with the game client from the game server.
(Hacking mainly means return of error recommended to disconnect from the client, among error codes described in the _AhnHS_VerifyResponsef() function.)

Internally,
_AhnHS_VerifyResponse function is called.

If the return value of the function is one of the followings,
ANTICPX_RECOMMEND_CLOSE_SESSION is returned.

- ERROR_ANTICPXSVR_BAD_MESSAGE
- ERROR_ANTICPXSVR_REPLY_ATTACK
- ERROR_ANTICPXSVR_UNKNOWN_CLIENT

- ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK
- ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK
- ERROR_ANTICPXSVR_MEMORY_ATTACK
- ERROR_ANTICPXSVR_OLD_VERSION_CLIENT_EXPIRED
- ERROR_ANTICPXSVR_NANOENGINE_FILE_ATTACK
- ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION
- ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION
- ERROR_ANTICPXSVR_VERIFY_EXCEPTION
- ERROR_ANTICPXSVR_INVALID_PARAMETER
- ERROR_ANTICPXSVR_ABNORMAL_HACKSHIELD_STATUS
- ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED

- Cause: Hacking has been detected from the client.
- Workarounds: None

ANTICPX_RECOMMAND_KEEP_SESSION (Value = 102)

- Description: The response from the client is Normal. Continue the connection with the game client from the game server.
- Cause: None
- Workarounds: None

ERROR_ANTICPXSVR_VERIFYEX_EXCEPTION (Value = 0xE904001A)

- Description: An exception occurred while executing _AhnHS_VerifyResponseEx function.
(This error can be converted only in module for Windows.)
- Cause: An exception could be caused by various problems.
- Workarounds: MiniDump file will be created.
The dump file is created with the name, 'VERIFYRESPONSEEX_year_month_date hour-minute-second.dmp'.
Send the file to AhnLab, Inc.

Remark

Specific Error Code is a code that describes the Error Code.

The specific error codes are as below.

(Specific Error Code only describes
ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED error code.)

ErrorCode	SpecificErr
ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED	Client-Side CallBack Code defined in AntiCpXSvr.h (AHNHS_ACTAPC_DETECT_SPEEDHACK, AHNHS_ENGINE_DETECT_GAME_HACK, AHNHS_ACTAPC_DETECT_MULTI_LOADING , AHNHS_ACTAPC_DETECT_AUTOMOUSE, ...)

Example

The following is an example of _AhnHS_VerifyResponseEx_WithInfo function.

Example

```

// unsigned long is 64bit in linux 64bit, so be careful not to use it as
// 32bit variable

unsigned long ulError = 0;
unsigned long ulSpecificError = 0;
unsigned long ulRet = ERROR_SUCCESS;
ulRet = _AhnHS_VerifyResponse ( hClient,
                                stResponseBuf.byBuffer,
                                stResponseBuf.nLength,
                                &ulError,
                                &ulSpecificError);

if ( ulRet == ANTICPX_RECOMMAND_CLOSE_SESSION )
{
    BOOL bKick = false;
    if ( ulError == ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED )
    {
        switch (ulSpecificError)
        {
            case : AHNHS_ACTAPC_DETECT_MULTI_LOADING:
                // disconnect when there is multi-loading in client.
                bKick = true;
                break;

            ...
            default:
                break;

        }

    }

    if ( true = bKick )
    {
        Log ("[AVREx] Disconnect the Client: 0x%x (0x%x)", ulError,
ulSpecificError);
        // Client Out -> _AhnHS_CloseClientHandle
        // and close the hClient handle
    }
}

```

_AhnHS_VerifyResponse

DESCRIPTION

Checks the client response to the request message issued by _AhnHS_MakeRequest function.

SYNTAX

```
unsigned long __stdcall  
_AhnHS_VerifyResponse (  
    IN AHNHS_CLIENT_HANDLE hClient,  
    IN unsigned char *pbyResponse,  
    IN unsigned long nResponseLength  
);
```

PARAMETERS

Parameter	Type	Description
hClient	AHNHS_CLIENT_HANDLE	Client handle
pbyResponse	char *	Data buffer received from the client
nResponseLength	unsigned long	Data length received from the client

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned when the function was successfully called.
- Cause: Normal.
- Workarounds:

ERROR_ANTICPXSVR_INVALID_PARAMETER (Value = 0xE9040003)

- Description: Incorrect input data.
- Cause: hClient and pbyResponse values are NULL.
- Workarounds: Check whether the hClient and pbyResponse values are NULL.

ERROR_ANTICPXSVR_BAD_FORMAT (Value = 0xE9040004)

- Description: Invalid format.
- Cause: There is a failure in the decoding process.
- Workarounds: Check whether the buffer has been properly sent to the _AhnHS_VerifyResponse of the client.

ERROR_ANTICPXSVR_NO_WAITING (Value = 0xE9040006)

- Description: No response to the request message has arrived.
- Cause: _AhnHS_MakeRequest function is called in the following ways: 1) Calls _AhnHS_MakeRequest function. 2) Receives Ack message from the client. 3) Calls AhnHS_VerifyResponse function. 4) Calls _AhnHS_MakeRequest function again. If the synchronization is not successful, a trouble may occur.
- Workarounds:
 - ① Check if there is any synchronization issue in the client session management.
 - ② Check if _AhnHS_VerifyResponse function has been called while _AhnHS_MakeRequest function had not been called.

ERROR_ANTICPXSVR_NOT_ENOUGH_MEMORY (Value = 0xE9040007)

- Description: Insufficient memory space.
- Cause: There is not enough memory in the server.
- Workarounds: Check whether there is memory leak in the server.

ERROR_ANTICPXSVR_BAD_MESSAGE (Value = 0xE9040008)

- Description: Message encryption/decryption failure.
- Cause: Invalid buffer value returned from the client.
- Workarounds: Check if the buffer value from the client has been normally received as pbyResponse.

ERROR_ANTICPXSVR_REPLY_ATTACK (Value = 0xE9040009)

- Description: Retransmission attack for packet analysis was detected.
- Cause: The buffer has been sent when the HackShield client was not operating.
- Workarounds: Attack by hacker to analyze packet.

ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK (Value = 0xE904000A)

- Description: HackShield Module manipulation has been detected.
- Cause: The HackShield file, ehsvc.dll, is manipulated.

- Workarounds: Check whether the client's ehsvc.dll file is manipulated.

ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK (Value = 0xE904000B)

- Description: Client file manipulation has been detected.
- Cause: The game client file has been manipulated.
- Workarounds: Check whether the game client has been updated properly.
Check whether the client is infected by virus.

ERROR_ANTICPXSVR_MEMORY_ATTACK (Value = 0xE904000C)

- Description: Memory manipulation has been detected.
- Cause: The client memory has been manipulated.
- Workarounds: Check whether there is a hacking tool that manipulates the client memory.

ERROR_ANTICPXSVR_OLD_VERSION_CLIENT_EXPIRED (Value = 0xE904000D)

- Description: The old version of client is connected.
- Cause: The client for the existing hsb has attempted to connect when hsb file has been uploaded without terminating the server.
- Workarounds: Check the version of the client.
If the version is low, patch the client to the latest version.

Note

It may occur when GrantOldSession = 0 in the HSBGen.ini has been set when creating hsb file, or when GrantOldSession = 1, but the number of allowed client versions has exceeded MaxAllowedNumber.

E.g.)
[VERCT]
GrantOldSession = 1
MaxAllowedNumber = 1

This means only the client for the hsb file in the server will be allowed.

ERROR_ANTICPXSVR_UNKNOWN_CLIENT (Value = 0xE904000E)

- Description: Does not pair with the client specified during creation of the HSB file.
- Cause: The version is not same as the client version used to create hsb by HSBGen.

- Workarounds: Create hsb file again and conduct the test again if the error occurred in the development phase. If the error occurred to a certain user during the service, check whether the client file has been successfully patched or infected with viruses.

ERROR_ANTICPXSVR_NANOENGINE_FILE_ATTACK
(Value = 0xE9040010)

- Description: 3n.mhe file manipulation has been detected.
- Cause: Occurs when 3n.mhe file is manipulated.
- Workarounds: Check whether 3n.mhe file is manipulated.

ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION
(Value = 0xE9040011)

- Description: The server does not support this HackShield version.
- Cause: Occurs when hshield.dat file that in the client is a lower version than the hshield.dat file in the server.
- Workarounds: Check whether the hshield.dat file in the client and the hshield.dat file in the server are the same version.

ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION
(Value = 0xE9040012)

- Description: Heuristic engine version is not supported by the server.
- Cause: Occurs when 3n.mhe file that in the client is a lower version than the 3n.mhe file in the server.
- Workarounds: Check whether 3n.mhe is in the client HackShield folder or is an older version.

ERROR_ANTICPXSVR_VERIFY_EXCEPTION
(Value = 0xE9040015)

- Description: An exception occurred while executing _AhnHS_VerifyResponse function or _AhnHS_VerifyResponseEx function.

(This error can be converted only in module for Windows.)

- Cause: An exception could be caused by various problems.
- Workarounds: MiniDump file will be created. The dump file is created with the name, 'VERIFYRESPONSE_year_month_date hour-minute-second.dmp'. Send the file to AhnLab, Inc.

ERROR_ANTICPXSVR_ABNORMAL_HACKSHIELD_STATUS
(Value = 0xE9040018)

- Description: The HackShield operation status is not Normal.

- Cause: The feature needed in HackShield that is running on the client is not operating properly.
- Workarounds: Check whether there is hacking tool that is attacking the HackShield running on the client.

ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED
(Value = 0xE9040019)

- Description: A hacking tool has been detected from the client.
- Cause: A hacking tool has been detected from the client.
- Workarounds: Check whether there is hacking tool that is attacking the HackShield running on the client.

ERROR_ANTICPXSVR_UNKNOWN
(Value = 0xE90400FF)

- Description: Undefined error.
- Cause: This error does not occur in normal cases; if this error occurs, additional check is required.
- Workarounds: Please contact AhnLab, Inc.

Caution

ERROR_ANTICPXSVR_BAD_MESSAGE
 ERROR_ANTICPXSVR_REPLY_ATTACK
 ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK
 ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK
 ERROR_ANTICPXSVR_MEMORY_ATTACK
 ERROR_ANTICPXSVR_OLD_VERSION_CLIENT_EXPIRED
 ERROR_ANTICPXSVR_NANOENGINE_FILE_ATTACK
 ERROR_ANTICPXSVR_UNKNOWN_CLIENT
 ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION
 ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION
 ERROR_ANTICPXSVR_VERIFY_EXCEPTION
 ERROR_ANTICPXSVR_INVALID_PARAMETER
 ERROR_ANTICPXSVR_ABNORMAL_HACKSHIELD_STATUS
 ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED

In case the above error code is returned, it is recommended that the session should be disconnected from the client and the game should be terminated. Other message may be added depending on the game developer's policies.

Example

The following is an example of calling `_AhnHS_VerifyResponse` function.

Example

```
ulRet = _AhnHS_VerifyResponse ( hClient,
                                stResponseBuf.byBuffer,
                                stResponseBuf.nLength );

if ( ulRet == ERROR_ANTICPXSVR_BAD_MESSAGE ||
    ulRet == ERROR_ANTICPXSVR_REPLY_ATTACK ||
    ulRet == ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK ||
    ulRet == ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK ||
    ulRet == ERROR_ANTICPXSVR_MEMORY_ATTACK ||
    ulRet == ERROR_ANTICPXSVR_OLD_VERSION_CLIENT_EXPIRED ||
    ulRet == ERROR_ANTICPXSVR_NANOENGINE_FILE_ATTACK ||
    ulRet == ERROR_ANTICPXSVR_UNKNOWN_CLIENT ||
    ulRet == ERROR_ANTICPXSVR_INVALID_HACKSHIELD_VERSION ||
    ulRet == ERROR_ANTICPXSVR_INVALID_ENGINE_VERSION ||
    ulRet == ERROR_ANTICPXSVR_VERIFY_EXCEPTION ||
    ulRet == ERROR_ANTICPXSVR_INVALID_PARAMETER ||
    ulRet == ERROR_ANTICPXSVR_ABNORMAL_HACKSHIELD_STATUS ||
    ulRet == ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED )
{
    // error
    // Client Out -> _AhnHS_CloseClientHandle
    // and close the hClient handle
}
```

_AhnHS_MakeResponse

DESCRIPTION

Used by the client. Decrypts the encrypted version request message form the server, encrypts the current client file version, and creates a response message.

SYNTAX

```
int __stdcall
_AhnHS_MakeResponse (
    unsigned char *pbyRequest,
    unsigned long ulRequestLength,
    PAHNHS_TRANS_BUFFER pResponseBuffer
);
```

PARAMETERS

Parameter	Type	Description
pbyRequest	unsigned char *	[IN] Request Message buffer
ulRequestLength	unsigned long	[IN] Request Message length
pResponseBuffer	PAHNHS_TRANS_BUFFER	[OUT] Response Message buffer

RETURN VALUE

ERROR_SUCCESS . (Value = 0x00000000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

ERR_ANTICPXCNT_INVALID_PARAMETER . (Value = 0xE4010001)

- Description: Incorrect parameters.
- Cause: The pbyRequest and pResponseBuffer values are NULL.
- Workarounds: Check whether the pbyRequest and pResponseBuffer values are NULL.

ERR_ANTICPXCNT_INVALID_ADDRESS (Value = 0xE4010002)

- Description: Invalid memory address access.
- Cause: There could be a problem in the system architecture.

- Workarounds: Please contact AhnLab, Inc.

ERR_ANTICPXCNT_NOT_ENOUGH_MEMORY (Value = 0xE40100013)

- Description: Insufficient memory space.
- Cause: Memory allocation has failed.
- Workarounds: Check whether there is a leak in the server memory.

ERR_ANTICPXCNT_CRC_TABLE_INIT_FAILED (Value = 0xE4010004)

- Description: Failed in initialization.
- Cause: The pbyRequest buffer size exceeds ANTICPX_TRANS_BUFFER_MAX (400).
- Workarounds: Check whether the HShield.h and library are the latest version in the build, and whether the pbyRequest buffer sent by the server has been delivered correctly.

ERR_ANTICPXCNT_BAD_LENGTH (Value = 0xE4010005)

- Description: Invalid message size.
- Cause: The size of the message buffer is incorrect.
- Workarounds: Check whether the latest version of HShield.h and library are used.

ERR_ANTICPXCNT_INSUFFICIENT_BUFFER (Value = 0xE4010006)

- Description: The size of the passed buffer is invalid.
- Cause: The versions of Ehsvc.dll, HShield.lib and Hshield.h are not identical.
- Workarounds: Check whether the latest version of HShield.h and library are used.

ERR_ANTICPXCNT_NOT_SUPPORTED (Value = 0xE4010007)

- Description: Not supported in the current version.
- Cause: The passed message type is not defined in HackShield.
- Workarounds: Check whether the client HackShield version and server HackShield version are identical.

ERR_ANTICPXCNT_FILE_NOT_FOUND (Value = 0xE4010008)

- Description: Cannot find the client file.
- Cause: The client file cannot be found.
- Workarounds: Please contact AhnLab, Inc.

ERR_ANTICPXCNT_INVALID_MESSAGE_SIZE (Value = 0xE4010009)

- Description: The size of the entered message is invalid.
- Cause: The size of the message passed from the server is invalid.
- Workarounds: Check whether the client HackShield version and server HackShield version are identical.

ERR_ANTICPXCNT_BAD_FORMAT (Value = 0xE401000A)

- Description: Invalid format.
- Cause: The size of the message passed from the server is invalid.
- Workarounds: Check whether the client HackShield version and server HackShield version are identical.

ERR_ANTICPXCNT_DEBUGGER_DETECTED (Value = 0xE401000B)

- Description: Debugging detected.
- Cause: The client is in debugging.
- Workarounds: Check whether it is currently in debugging.

ERR_ANTICPXCNT_BAD_HSHIELD_MODULE (Value = 0xE401000C)

- Description: Incorrect HackShield module path or wrong HackShield module.
- Cause: No HackShield module exists.
- Workarounds: Check whether the HackShield module path and HackShield module are correct.

ERR_ANTICPXCNT_BAD_CLIENT_FILE (Value = 0xE401000D)

- Description: Invalid client module.
- Cause: A problem occurred while accessing the client file.
- Workarounds:

ERR_ANTICPXCNT_BAD_REQUEST (Value = 0xE401000E)

- Description: Invalid request message from the server.
- Cause: Decoding failure message occurs as the buffer passed from the server was abnormal.
- Workarounds: Check the buffer data passed from the server and whether the length returned by pbyRequest is correct.

ERR_ANTICPXCNT_HSHIELD_CORE_ENGINE_NOT_WORKING (Value = 0xE401000F)

- Description: HackShield core engine is not properly running.
- Cause: This error does not occur in normal cases; if this error occurs, additional check is required.
- Workarounds: Please contact AhnLab, Inc.

ERR_ANTICPXCNT_UNKNOWN . (Value = 0xE40100FF)

- Description: Returned when the system does not operate properly due to hacking attacks.
- Cause: This error does not occur in normal cases; if this error occurs, additional check is required.
- Workarounds: Please contact AhnLab, Inc.

Example

The following is an example of using `_AhnHS_MakeResponse`.

Example

```
ulRet = _AhnHS_MakeResponse ( stRequestBuf.byBuffer,
                             stRequestBuf.nLength,
                             &stResponseBuf );

if ( ulRet != ERROR_SUCCESS )
{
    // error
}
```

5. Monitoring Service

5.1. Overview

AhnLab HackShield Hacking Monitoring System centrally monitors hack attacks and errors in the HackShield-applied programs. Hack attacks and errors can cause damage to the service, so it is important to monitor hack attacks and errors in real-time to prevent damages. AhnLab HackShield Hacking Monitoring System provides reporting feature that shows statistical data on the service status.

Functions

Real-time Hacking/error Monitoring

Monitors hacking and error status in the HackShield-applied program and the data that HackShield provided in real time.

Report

Provides various statistical reports based on the collected log.

Policy Management

Sets various policies depending on the situation and backs up the DB.

Features

Interface Function (API)

Provides interface DLL and library for the developer to easily use Ehsvc functions and to check the results. Using the provided interface DLL and library, the developer can check the integrity of the game file and operational state of HackShield.

Monitoring Server Program

Provides a monitoring program for the developer to manage and check the Oracle DB in real time in order to check hacking and error data from the client in real time.

Test Program

Provides Amazon.exe, a test program, implemented by the APIs provided by Ehsvc. Amazon.exe provides the existing HackShield test function and the Ehsvc test function.

System Architecture

Provides HShield.lib and EHSvc.dll which are applied to the server and the client. The general architecture and the operating principles of the monitoring service are as follows.

Ehsvc.dll (Interface dll)

Provides an API which can set basic information in order to notify hacking attack or error occurrence to the monitoring server.

HShield.lib (HackShield library)

Provides an API which can set basic information in order to notify hacking attack or error occurrence to the monitoring server.

HSMS_Setup.exe (Monitoring Server Side Installation File)

A server-side program which manages the data from the client in the DB and allows management on the web.

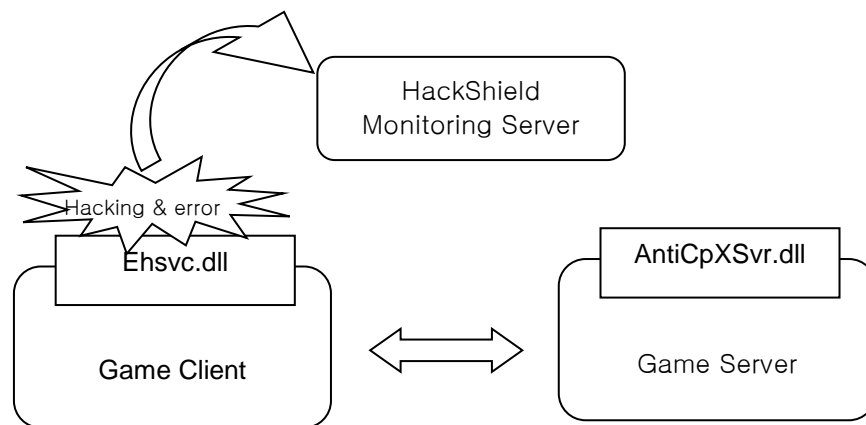


Fig. 5-1 Monitoring service

When the game client calls _AhnHS_StartMonitor function, the client will notify the monitoring server of the hacking and error data if there is any.

The monitoring server manages the client program using a certain game code. The data that the client sends in relation to the game code will be managed in the DB. The developer can easily access the DB through the web.

5.2. Application Programming

Set monitoring server information and user information using the API provided by Ehsvc.

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

Programming Application

Follow the preparation below before starting programming.

5.2.1.1. Monitoring-related Files

Monitoring-related Files

Table 5-1 Monitoring-related Files

File name	Installation folder	Description
HShield.h	[Program folder] source	Header file to be used in the client. HackShield functions included.
HShield.lib	[Program folder] source	Library file to be used in the client. HackShield function included.
EHSvc.dll	[Program folder] source	dll file to be used in the client. HackShield functions included.

5.2.1.2. Application

Server Application

1. Install HackShield Monitoring Server Program in the server by referring to the HackShield Monitoring Server Installation Guide.
2. Run the server by referring to the HackShield Monitoring Server Operation Guide.

Client Application

1. Include HShield.lib file in the project.
2. Include the provided HShield.h file in the source file.
3. Create a parameter for AHNHS_EXT_ERRORINFO structure defined in HShield.h and initialize it.

```
AHNHS_EXT_ERRORINFO HsExtError = { 0, };
```

4. Substitute data in szServer(Monitoring Serveraddress), szUserId(User account), and szGameVersion(game Version) in HsExtError structure.

```
HsExtError.szServer = "127.0.0.1"    //Monitoring address  
HsExtError.szUserId = "Test"        //User ID  
HsExtError.szGameVersion = "3.0.0.1" //Game Version
```

Note

The monitoring IP/Port can be modified after through HSUpdate.env file.

5. Send the structure parameters created in the above step and the full path of Ehsvc.dll as parameters, and call _AhnHS_StartMonitor function.

Caution

_AhnHS_StartMonitor function must be called before _AhnHS_Initialize function.

Otherwise, the errors created before HackShield is executed will be not transmitted.

Caution

When applying the HackShield update library and HackShield library to the same game module, HackShield update library must be applied before calling _AhnHS_StartMonitor function.

Otherwise, HackShield may not be updated properly.

```

lstrcat (szFullFileName, _T("\\HShield\\Ehsvc.dll" ));

dwRet = _AhnHS_StartMonitor (  HsExtError      // [in]
                               szFullFileName  // [in]
                               );

if( dwRet != ERROR_SUCCESS)
{
    // In case a fail occurs, it means that only monitoring is not made.
    // Save only error logs.
}

dwRet = _AhnHS_Initialize ( ...

```

6. If user ID cannot be acquired in Step 4, call _AhnHS_SetUserId function when the user ID is available in order to set the user ID.

5.3. Application Programming Interface

_AhnHS_StartMonitor

DESCRIPTION

Sets server information to send the data in case of hacking or error occurrence.
Called before _AhnHS_Initialize function.

SYNTAX

```
int  
__sdtdcall  
_Ahn_StartMonitor ( IN AHNHS_EXT_ERRORINFO HsExtErrorInfo,  
                   IN LPCSTR szFileName  
                   );
```

PARAMETERS

Parameter	Value	Description
HsExtErrorInfo	AHNHS_EXT_ERRORINFO	Structure with server URL address, User ID, and game version
szFileName	LPCSTR	Full path of Ehsvc.dll

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned when the function was successfully called.
- Cause: Normal
- Workarounds:

HS_ERR_INVALID_FILES (Value = 0x1C001)

- Description: Incorrect input data
- Cause: The parameter is NULL.
- Workarounds: Check whether HsExtErrorInfo and szFileName are Normal.

HS_ERR_UNKNOWN (Value = 0x1C002)

- Description: Unknown error.
- Cause: There could be an exception in the function, or a problem in the function structure.
- Workarounds: Send HShield.log file and AhnReport to AhnLab, Inc.

Example

The following is an example of calling `_AhnHS_StartMonitor` function.

```
Example

AHNHS_EXT_ERRORINFO HsExtError;    //Structure defined in HShield.h

HsExtError.szServer = "127.0.0.1"    //Monitoring address
HsExtError.szUserId = "Test"         //User ID
HsExtError.szGameVersion = "3.0.0.1" //Game Version

lstrcat (szFullFileName, _T("\\HShield\\Ehsvc.dll" ));

dwRet = _AhnHS_StartMonitor (  HsExtError      // [in]
                               szFullFileName  // [in]
                               );

if( dwRet != ERROR_SUCCESS)
{
    // In case a fail occurs, it means that only monitoring is not made.
    // Save only error logs.
}

dwRet = _AhnHS_Initialize ( ...
```

_AhnHS_SetUserId

DESCRIPTION

Saves the user ID of the message to be sent to the monitoring server.
Although `_AhnHS_StartMonitor` receives user ID, user information may not be provided when HackShield is initialized. Call this function when the user ID is provided to get user information. Before the ID is acquired, error data is sent without ID information.

SYNTAX

```
void __stdcall  
_AhnHS_SetUserId ( IN LPCSTR szUserID )
```

PARAMETERS

Parameter	Value	Description
szUserID	LPCSTR	User information of the game client (Up to 120 bytes are allowed for szUserID) Ex) Alphabets (128 characters), other characters (40 characters)

RETURN VALUE

None.

Example

The following is an example of calling `_AhnHS_SetUserId` function.

Example

```
_AhnHS_SetUserId ( szUserID );
```

_AhnHS_SetUserCustomInfo

DESCRIPTION

If user information is sent through this function, the information detected by HackShield will be sent to the monitoring server.

But, the detected information will not be sent to the monitoring server if this function is called from the HackShield callback function that is implemented from the game.

SYNTAX

```
int  
__stdcall  
_AhnHS_SetUserCustomInfo (  
    IN const char* szUserCustomInfo  
);
```

PARAMETERS

Parameter	Value	Description
szUserCustomInfo	const char *	User information to send to monitoring server (Up to 512 bytes are allowed for szUserCustomInfo.)

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal If this value is returned, HackShield will send additional information to the monitoring server.

HS_ERR_INVALID_PARAM (Value = 0x002)

- Description: Wrong parameters.
- Cause: Occurs when the szUserCustomInfo value is NULL.

EXAMPLE

The following is an example of calling _AhnHS_SetUserCustomInfo function.

Example

```
CHAR szUserCustomInfo[512] = { 0, };  
StringCchcopy( szUserCustomInfo, "Extra Data : 1234 " );  
_AhnHS_SetUserCustomInfo ( szUserCustomInfo );
```

_AhnHS_SendUserCustomInfo

DESCRIPTION

Sends user information to the monitoring server through parameters.

SYNTAX

```
int  
__stdcall  
_AhnHS_SendUserCustomInfo (  
    IN const char* szUserCustomInfo,  
    IN DWORD dwTimeout = DEFAULT_HSMS_TIME_OUT  
);
```

PARAMETERS

Parameter	Value	Description
szUserCustomInfo	const char *	Custom information (Up to 512 bytes are allowed for szUserCustomInfo)
dwTimeout	DWORD	Timeout until custom information is sent to the monitoring server DEFAULT_HSMS_TIME_OUT = 5000 (about 5 seconds) (if the dwTimeout is 0, it will be returned without timeout.)

RETURN VALUE

HS_ERR_OK (Value = 0x000)

- Description: Returned when the function was successfully called.
- Cause: Normal If this value is returned, HackShield will send additional information to the monitoring server.
- Workarounds:

HS_ERR_UNKNOWN (Value = 0x001)

- Description: Unknown error occurred.
- Cause: There could be an exception in the function, or a problem in the function structure.
- Workarounds: Send HShield.log file and AhnReport to AhnLab, Inc.

HS_ERR_INVALID_FILES (Value = 0x101)

- Description: HackShield has not been initialized.
- Cause: _AhnHS_Initialize function was not called or the function was called as HackShield had not been initialized.
- Workarounds: HackShield is not started or HackShield is not operating properly due to a hack attack.

HS_ERR_INVALID_PARAM (Value = 0x002)

- Description: Wrong parameters.
- Cause: AhnHS_StartService and monitoring service have not started.
- Workarounds: Monitoring service is not working properly.

HS_ERR_HSMS_WAIT_TIME_OUT (Value = 0x801)

- Description: The error information cannot be sent to the monitoring server within the specified time.

- Cause: The specified time is too short or a problem occurred while sending the error information.
- Workarounds:

HS_ERR_HSMS_NOT_RUNNING (Value = 0x803)

- Description: HackShield monitoring service is not starting.
- Cause: It occurs when you call when HackShield monitoring service has not started. This error occurs only in the development process, so no other workaround is required.
- Workarounds: This error occurs only in the development process, so no other workaround is required.

REMARKS

- Timeout for more than 5 seconds is needed when applying the function.
- If the timeout is 0, it will be returned without a waiting time.
- Call after AhnHS_StartMonitor and AhnHS_StartService.
- It gets sent by the monitoring service as custom error code (0xFFFF0001).

EXAMPLE

The following is an example of calling _AhnHS_SendUserCustomInfo function.

```
Example
// Synchronized call– 5 seconds of timeout
_AhnHS_SendUserCustomInfo ( "Param" );

// No timeout
_AhnHS_SendUserCustomInfo ( "Param", 0 );

// Synchronized call– 50 seconds of timeout
_AhnHS_SendUserCustomInfo ( "Param", 50000 );
```

6. LMP

6.1. Overview

The Local Memory Protection (LMP) function detects memory manipulation at the client end for the case when Packer ² cannot protect the server interface memory. This function is similar to the memory manipulation detection function at the existing server interface. However, LMP function allows the client to protect the memory without passing through the server.

LMP 1.0

With CSInspector tool, you can apply Protection file to EXE or DLL. THEMIDA's API Wrapping option is not supported.

- Protection file: EXE, DLL files
- Used tool: CSInspector.exe

LMP 2.0

It has the same features as LMP 1.0. As a local memory protection feature developed to support THEMIDA's API Wrapping option, it can be applied to the game client file through hsb file generation through server interaction.

- Protection file: EXE, DLL files
- Used tool: HSBGen.exe

Caution

It is recommended to apply either LMP 1.0 or 2.0 to the file to protect.

Functions

Memory protection of executable file (LMP 1.0 and LMP 2.0 supported)

Detects manipulation for the code area in the memory of the execution file.

Memory protection of DLL file (LMP 1.0 and LMP 2.0 supported)

Detect manipulation in the code area of the specified DLL file, and protects even the files not directly built by the corresponding DLL.

Caution

² Themida A packer program changes the codes for memory address to protect the memory whenever it is executed, so it could collide with server interaction that checks the memory CRC from the server.

To protect dynamically loaded dll file,

when you load the dll after calling `_AhnHS_Initialize` and `_AhnHS_StartService`,
you must call

`_AhnHS_IsModuleSecure(szDllPath)` function after loading the dll file.

Memory integrity check (LMP 1.0 and LMP 2.0 supported)

Checks the hooking status of the system file and verifies memory integrity in order to prevent memory modification before the LMP function starts to run.

Features

Options

The corresponding function will be executed when `AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION` option is activated in the initialization part.

Callback Message

In case memory manipulation is detected by the corresponding function, the callback function of HackShield will send `AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP` message.

Test Program

Provides `Amazon.exe`, a test program, implemented by the APIs provided by `Ehsvc`. `Amazon.exe` provides the existing HackShield test function and the `Ehsvc` test function.

System Architecture

Inputs the section data in the module to be protected by `CSInspector.exe` and provides `HShield.lib` and `EHsvc.dll` which are applied to the client in the format of dll and library files.

Ehsvc.dll (Interface dll)

Provides an API which can set basic information in order to notify hacking attack or error occurrence to the monitoring server.

HSshield.lib (HackShield library)

Provides an API which can set basic information in order to notify hacking attack or error occurrence to the monitoring server.

CSInspector.exe (Protection Module Setting Utility) (LMP 1.0)

Inputs section information to the client file or the third-party module to be protected in order for the HackShield to protect the code area of the corresponding module.

HSBGen.exe (Protection Module Setting Utility) (LMP 2.0)

Inputs section information to the client file to protect in order for the HackShield to protect the code area of the corresponding module.

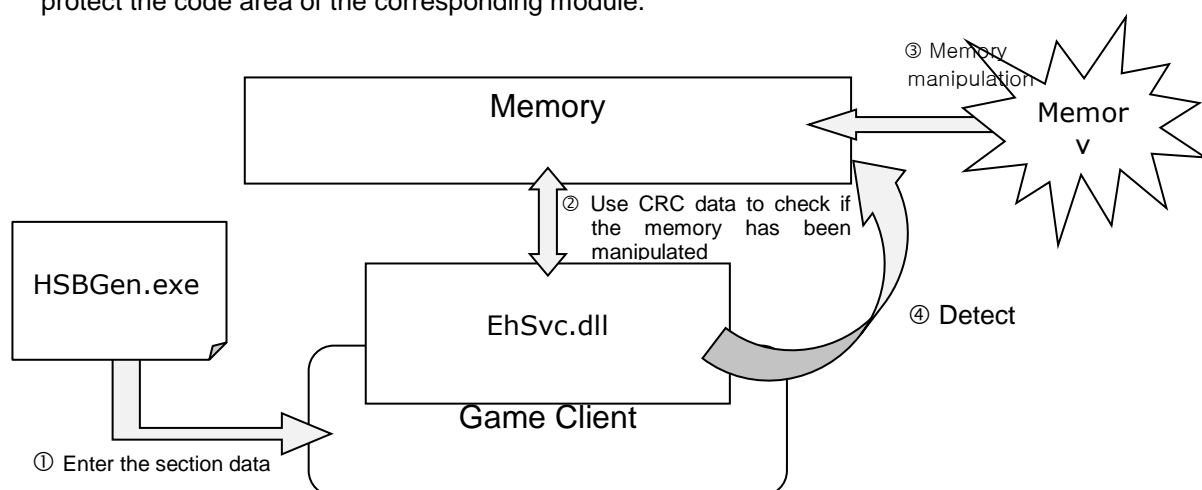


Fig. 6-1 Local Memory Protection

Operating principles of the Local Memory Protection function are as follows:

- ④ Save the section data of the game client of DLL in a certain PE structure area using HSBGen.exe tool provided by HackShield.
- ⑤ After executing HackShield, load the section data in the PE structure of the game client or DLL, and check whether memory manipulation has occurred before execution.
- ⑥ Create CRC data of the current memory based on section information.
- ⑦ Periodically check the memory manipulation status based on CRC data.

6.2. Application Programming

The user can also use LMP function through HackShield initialization function (_AhnHS_Initionlize) provided by Ehsvc.

Note

The sample codes contained in this document are based on C/C++ language in Microsoft Visual C++ 6.0. Programming language may be changed depending on the characteristics of each program and system environments.

Programming Application

Before using the LMP function, the developer shall do the following.

6.2.1.1. LMP-related Files

LMP-related Files

Table 6-1 LMP-related Files

위치	File name	Installation folder	Description
[SDK]\Include\	HShield.h	[Program source folder]	Header file to be used by the client
[SDK]\Lib	HSheild.lib	[Program source folder]	Library file to be used by the client
[SDK]\Bin\Win \x86\Util	CSInspector.exe		Utility to be set in the module to be protected
[SDK]\Bin\Win \x86\AntiCrack \	HSBGen.exe		Utility to be set in the module to be protected

6.2.1.2. Application

Client Application

1. Include HShield.lib file in the project.
2. Include the provided HShield.h file in the source file.
3. When calling _AhnHS_Initialize function, add the following option to the fifth

parameter in the source.

```
// Define the option flag to call _AhnHS_Initialize function.
// (Add to the existing option.)
dwOption = AHNHS_CHKOPT_ALL |
AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION;

// Initialize HackShield service by calling _AhnHS_Initialize.
nRet = _AhnHS_Initialize ( szFullFilePath,
                          HS_CallbackProc, // Callback function
                          1000,           // game code
                          "B228F291B7D7FAD361D7A4B7",
                          // License key
                          dwOption,
                          // option flag

AHNHS_SPEEDHACK_SENSING_RATIO_NORMAL
);
...
```

4. Add AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP in the event transmission function part in relation to HackShield.

```
int __stdcall HS_CallbackProc ( long ICode, long IParamSize, void* pParam )
{
    TCHAR szMsg[MAX_PATH];

    // Display a corresponding error message.
    switch ( ICode )
    {
        // LMP Callback
        // Note: Changed module name and the page address are returned
        // but they don't need to be exposed
        // to the user.
        case AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP:
            wsprintf(szMsg, "Memory
                        manipulation has been detected.\n" );
            MessageBox( NULL, szMsg, szTitle, MB_OK );
            break;

        ....
    }
}
```

The following are sent as parameters:

- ICode
AHNHS_ACTAPC_DETECT_MEM_MODIFY_FROM_LMP(0x10705)
- pParam
"Manipulated module name (Module base address): Manipulated actual page address"
The name and the page address of the manipulated module are returned.
The corresponding information may not need to be exposed to the user.

Protection module applied - CSInspector.exe (LMP 1.0)

When distributing packed modules, apply CSInspector.exe first before applying the packer.

1. Use for the module which is designed to protect CSInspector.exe provided for the LMP function.
2. CSInspector.exe tool operates in command-line type. Input the followings on the command-line input window:
C:\> CSInspector.exe Target.exe
3. In case there is additional DLL file to protect, apply CSInspector.exe in the same way.
C:\> CSInspector.exe Target.dll
4. After normal execution, SUCCESS message will be printed.
5. Perform packing or CRC extraction after executing CSInspector.exe.

Note

For more information about CSInspector, refer to [8.3 CSInspector Tool \(For HackShield 5.1 or higher\)](#)

Protection module applied - HSBGen.exe (LMP 2.0)

Note

For more information about HSBGen, refer to [8.1 Using HSBGen Tool](#).

Other features

`_AhnHS_IsModuleSecure`

Check integrity on whether the CSInspector.exe (LMP1.0) and HSBGen.exe (LMP2.0) inserted in the DLL to protect exists.

The following is an example of calling `_AhnHS_IsModuleSecure` function.

Example

```

_AhnHS_Initialize      (..) ;                // HackShield initialization
_AhnHS_StartService (..);                // HackShield start

LoadLibrary("C:\\GAME\\GameEngine.dll"); // DLL-loading

// HackShield must be properly initialized and DLL to protect must be loaded before
// calling.
BOOL bRet = _AhnHS_IsModuleSecure ("C:\\GAME\\GameEngine.dll");

```

Description

If the bRet is TRUE, the information entered into CSInspector is normal, and if it is FALSE, the information is either manipulated or does not exist. This feature detects hack attacks that change the dll to a dummy dll.

_AhnHS_IsModuleSecure

DESCRIPTION

Checks whether LMP feature is properly applied to DLL or EXE that is applied through CSInspector (LMP1.0) and HSBGen.exe (LMP2.0), or whether the applied LMP data is damaged or not.

Used when requesting protection for dynamically loaded DLL (with LMP applied). (No need to call statically binded DLL (with LMP applied).)

SYNTAX

```
BOOL __stdcall _AhnHS_IsModuleSecure (IN LPCSTR szModulePath )
```

PARAMETERS

Parameter	Value	Description
szModulePath	LPCSTR	DLL or EXE path (Entire path)

RETURN VALUE

TRUE

- Description: The LMP information of the module is valid, and properly added to LMP protection list.

- Cause: Normal

FALSE

- Description: There is no LMP information in the module of the passed path or it is not valid, so it will not be protected by LMP.
- Cause: The module of the passed path has been exchanged with another module, or the LMP information has been damaged.
- Workarounds: Check whether LMP has been properly applied to the module of the passed path.

Example

The following is an example of calling `_AhnHS_IsModuleSecure` function.

```
Example
BOOL bRet = _AhnHS_IsModuleSecure ("C:\\GAME\\GameEngine.dll");
```

Caution

- Call the function only after HackShield has been properly initialized (`_AhnHS_Initialize`).

Note on Server-side Detection

When using the extended server-side detection with the LMP with CSInspector, note the followings:

- Extended server-side detection (But, does not apply if using HSBGen [LMP2.0].)
 - In order to apply the LMP function to the game client file, use CSInspector.exe first before creating anticpx.hsb by using HSBGen.exe.
- If extended server-side detection are in use, apply the LMP function to the corresponding module and create anticpx.hsb again using HSBGen.exe.

Supported Packers

Currently, the LMP supports the following packers.

Table 6-2 Packets that LMP Supports

Packer name	Version	Remarks
Themida	2.1.3.0	LMP1.0 does not support

		ApiWrap. Available by LMP 2.0 or higher
Armadillo	V6.4.0.640	Partially supported according to packing options (※ Refer to Doc\Additional\Packer_HackS hield_compatibility.pdf)

7. Other features

7.1. Data file/message encryption

7.1.1. Overview

HsCryptoUtil is a data encryption/decryption SDK. Recent computer development requires stronger encryption technologies than ever. To reflect the current needs, Advanced Encryption Standard (AES) stronger than the Data Encryption Standard (DES) has been adopted as the standard. HsCryptoUtil provides stronger data encryption/decryption using 128bit AES.

Functions

Data Encryption/Decryption

File and message encryption/decryption library provides HsCryptLib.lib (for Windows), libhscrypt.so (for Linux), and HsCryptLib.h; allows the game developer to easily encrypt/decrypt messages and files.

Features

Interface Function (API)

Provides interface library for the developer to easily use the functions of HsCryptLib and check the return values. The game developer can easily encrypt/decrypt files and messages using the provided interface libraries.

Provides HsCryptoUtil.exe, a file encryption tool, and allows for decryption of the encrypted files using the APIs of HsCryptLib.

System Architecture

HsCryptLib is provided as an SDL library, not as an independent execution file. General architecture and operating principles of HsCryptLib are as follows:

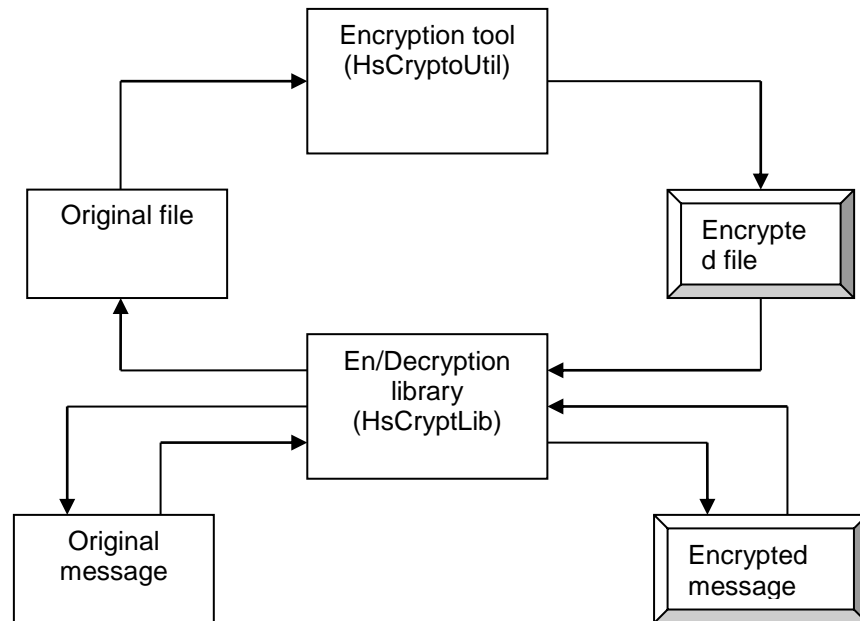


Fig. 7-1 HsCryptLib structure and operation

Interface Library: HsCryptLib.lib(for Windows), libhscrypt.so (for Linux)

An interface library file which provides APIs to encrypt/decrypt the data.

HsCryptoUtil Program

An encryption program which provides a function to encrypt files using HsCryptLib.lib.

7.1.2. Application Programming

Programming Procedure

The game developer can implement HsCryptLib as follows:

1. Preparation: Check the list of the provided HsCryptLib file, and copy necessary files.
2. Calling Encryption/Decryption Initialization Function: Create a function to initialize encryption/decryption and call the function before data decryption/encryption. Unless encryption/decryption is initialized, file and message encryption/decryption is not available.
1. Calling File Decryption Function: Write a code to call a function which decrypts part or entirety of the encrypted file. The decrypted data will be outputted to the buffer.
2. Calling Message Encryption Function: Write a code to call a function which encrypts messages. The encrypted data will be outputted to the buffer.
3. Calling Message Decryption Function: Write a code to call a function which decrypts messages. The decrypted data will be outputted to the buffer.
4. Test whether the written source code normally operates.
5. Distribute the users.

Preparation

Follow the preparation steps below before starting to program by using HsCryptLib:

HsCryptLib File

HsCryptLib File

Table 7-1 HsCryptLib File

File name	Installation folder	Description
HsCryptLib.h	[Program source folder]	Header file
HsCryptLib.lib	[Program source folder]	Windows Library File : : Multi-thread and single thread library is provided
libhscrypt.so	[Program source folder]	Linux Library File

Compiler Setting

The project file of the encryption/decryption program which uses HsCryptLib must include HsCryptLib.lib (libhscrypt.so) file in the library or source code. However, for the project using Windows HsCryptLib, check if the project uses either of a multi-thread library or single-thread library and apply a proper HsCryptLib.lib.

HsCrypt_InitCrypt

When it is ready for programming, call HsCrypt_InitCrypt first. Only after HsCrypt_InitCrypt function is successfully called, the encryption/decryption keys can be created which can be used for data encryption/decryption.

The following is an example of calling HsCrypt_InitCrypt function.

Example

```
typedef struct _HSCRYPT_KEYINFO
{
    BYTE  byInitKey[HSCRYPTLIB_INITKEY_SIZE]; // Initialization key
    BYTE  AesEncKey[HSCRYPTLIB_KEY_SIZE];      // encryption key
    BYTE  AesDecKey[HSCRYPTLIB_KEY_SIZE];      // decryption key
} HSCRYPT_KEYINFO, *PHSCRYPT_KEYINFO;

HSCRYPT_KEYINFO HsKeyInfo;
memcpy( HsKeyInfo.byInitKey, pbyInitKey, HSCRYPTLIB_INITKEY_SIZE );

dwRet = _HsCrypt_InitCrypt ( &HsKeyInfo );
```

In the above example, HSCRYPT_KEYINFO structure is declared and the initialization key is inputted in byInitKey. The size of the initialization key in HsKeyInfo.byInitKey shall be set to 16 bytes.

Calling HsCrypt_InitCrypt will allocate key values to AesEncKey (encryption key) and AesDecKey (decryption key) of HSCRYPT_KEYINFO structure. The messages and files are encrypted/decrypted based on these key values.

Caution

The encryption and decryption keys created by the initialization key shall be identical to encrypt/decrypt the files and messages.

HsCrypt_GetEncMsg

Encrypts messages. Initializes encryption/decryption by using HsCrypt_InitCrypt and encrypts the data by using HsCrypt_GetEncMsg. At this time, the encrypted data will be outputted to the buffer.

Example

```
dwRet = _HsCrypt_GetEncMsg (
    byPlainMsg,           // [in] buffer to be encrypted
    sizeof(byPlainMsg),   // [in] encryption size
    HsKeyInfo.AesEncKey,  // [in] encryption key
    byEncMsg              // [out] encrypted buffer
```

```
);
```

Note

The message sizes before encryption shall be same with that after encryption.

`_HsCrypt_GetDecMsg`

Decrypts messages. Initializes encryption/decryption and calls `_HsCrypt_GetDecMsg` in order to decrypt the data. At this time, the decrypted data will be outputted to the buffer.

Example

```
dwRet = _HsCrypt_GetDecMsg (
    byEncMsg,                // [in] buffer to be decrypted
    sizeof(byEncMsg),        // [in] decryption size
    HsKeyInfo.AesDecKey,     // [in] decryption key
    byDecMsg                 // [out] decrypted buffer
);
```

Note

The message sizes before decryption shall be same with that after decryption.

`HsCrypt_FRead`

Decrypts part of or entirety of the file using the file structure pointer. Initializes the encryption/decryption key and calls `fseek` function in order to move to the file pointer and encrypt the data. At this time, the decrypted data will be outputted to the buffer.

Example

```
dwRet = _HsCrypt_FRead (
    byPlainBuf,              // [out] decrypted buffer
    dwDecSize,               // [in] decryption size
    InputStream,             // [in] file pointer to read
    HsKeyInfo.AesDecKey,     // [in] decryption key
    &dwReadLen               // [out] decrypted size
);
```

Performs decryption/encryption using the block password (Block Cipher). The game developer can specify a block to decrypt and save the necessary part in the

buffer for return.

In order to decrypt the entire file, set the beginning of the file as the file structure pointer using fseek function and input the entire file size as the second parameter of _HsCrypt_FRead.

Caution

Input the decryption key corresponding to the encryption key. In case key management is not easy, use the key which initialized the encryption/decryption key and the decryption key which was created by _HsCrypt_InitCrypt.

7.1.3. Application Programming Interface

_HsCrypt_InitCrypt

DESCRIPTION

Initializes the encryption/decryption function.

SYNTAX

```
DWORD __stdcall  
_HsCrypt_InitCrypt (  
    IN OUT PHSCRYPT_KEYINFO pHsKeyInfo  
);
```

PARAMETERS

Parameter	Description
PHSCRYPT_KEYINFO Structure Definition typedef struct_HSCRYPT_KEYINFO { BYTE byInitKey[HSCRYPTLIB_INITKEY_SIZE]; BYTE AesEncKey[HSCRYPTLIB_KEY_SIZE]; BYTE AesDecKey[HSCRYPTLIB_KEY_SIZE]; } HSCRYPT_KEYINFO, *PHSCRYPT_KEYINFO;	[in][out] Encryption/Decryption key structure Initialization Key (16bytes) encryption key (550bytes) Decryption key

	(550bytes)
--	------------

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned after initialization success.
- Cause: None
- Workarounds: None

ERROR_HSCRYPTLIB_INITCRYPT_INVALIDPARAM (Value = 0x0001B002)

- Description: Returned when incorrect parameters were inputted.
- Cause: The passed pointer is null.
- Workarounds: Check whether the passed argument is Normal.

Others

WIN32 Defined Error (Value = WIN32 Defined)

REMARKS

This function shall be called in order to set the data necessary for encryption/decryption. Allocate byInitKey of HSCRYPT_KEYINFO structure and call _HsCrypt_InitCrypt function; get AesEncKey(encryption key) and AesDecKey(Decryption key). The game developer can encrypt/decrypt messages and files using the key.

_HsCrypt_GetEncMsg

DESCRIPTION

Encrypts the message and outputs to the encrypted data buffer.

SYNTAX

```
DWORD __stdcall  
_HsCrypt_GetEncMsg (  
    IN PBYTE pbyInput,  
    IN UINT nInLength,  
    IN PBYTE pAesEncKey,  
    OUT PBYTE pbyOutput  
);
```

PARAMETERS

Parameter	Description
pbyInput	[in] Buffer to be encrypted
nInLength	[in] Encryption size
pAesEncKey	[in] Encryption key
pbyOutput	[out] Encrypted buffer

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned after successful message encryption.
- Cause: None
- Workarounds: None

ERROR_HSCRYPTLIB_GETENCMMSG_INVALIDPARAM (Value = 0x0001B003)

- Description: Returned when incorrect parameters are inputted.
- Cause: The passed parameter is not Normal.
- Workarounds: Check whether the passed parameter is null or '0'.

Others

WIN32 Defined Error (Value = WIN32 Defined)

_HsCrypt_GetDecMsg

DESCRIPTION

Decrypts the message and outputs to the decrypted data buffer.

SYNTAX

```
DWORD __stdcall  
_HsCrypt_GetDecMsg (  
    IN PBYTE pbyInput,  
    IN UINT nInLength,  
    IN PBYTE pAesDecKey,  
    OUT PBYTE pbyOutput  
);
```

PARAMETERS

Parameter	Description
pbyInput	[in] Buffer to be decrypted
nInLength	[in] Decryption size
pAesEncKey	[in] Decryption key
pbyOutput	[out] Decrypted buffer

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned after successful message encryption.
- Cause: None
- Workarounds: None

ERROR_HSCRYPTLIB_GETDECMMSG_INVALIDPARAM(Value = 0x0001B004)

- Description: Returned when incorrect parameters are inputted.
- Cause: The passed parameter is not Normal.
- Workarounds: Check whether the passed parameter is null or '0'.

Others

WIN32 Defined Error (Value = WIN32 Defined)

HsCrypt_FRead

DESCRIPTION

Decrypts only some part in the file and outputs the data to the buffer.

SYNTAX

```
DWORD __stdcall  
_HsCrypt_FRead (  
    OUT LPVOID lpOutBuffer,  
    IN DWORD dwDecryptSize,  
    IN FILE *pInputStream,  
    IN PBYTE pAesDecKey,  
    OUT PDWORD pdwReadLen  
);
```

PARAMETERS

Parameter	Description
lpOutBuffer	[out] Decrypted buffer
dwDecryptSize	[in] Decryption size
pInputStream	[in] File structure pointer to be decrypted
pAesDecKey	[in] Decryption key
pdwReadLen	[out] Decrypted size

RETURN VALUE

ERROR_SUCCESS (Value = 0x00000000)

- Description: Returned after successful decryption.
- Cause: None
- Workarounds: None

ERROR_HSCRYPTLIB_FREAD_INVALIDPARAM (Value = 0x0001B005)

- Description: Returned when incorrect parameters are inputted.
- Cause: The passed parameter is not Normal.
- Workarounds: Check whether the passed parameter is null or '0'.

ERROR_HSCRYPTLIB_FREAD_GETFILELEN (Value = 0x0001B009)

- Description: Returned when the file size is not acquired.

- Cause: The file handle is not Normal.
- Workarounds: Check whether the file handle is Normal.

ERROR_HCRYPTLIB_FREAD_SIZEZERO (Value = 0x0001B00B)

- Description: Returned when the file size is 0.
- Cause: There is no file to be decrypted.
- Workarounds: Check whether the file is Normal.

ERROR_HCRYPTLIB_FREAD_GETPOSITION (Value = 0x0001B00A)

- Description: Returned when current file pointer location was not acquired.
- Cause: The passed file handle is not Normal.
- Workarounds: Check the passed file handle parameter.

ERROR_HCRYPTLIB_FREAD_FSEEK (Value = 0x0001B00C)

- Description: Returned when moving to the current file pointer block failed.
- Cause: The passed file handle is not Normal.
- Workarounds: Check the passed file handle parameter.

ERROR_HCRYPTLIB_FREAD_DECRYPT_RANGE (Value = 0x0001B006)

- Description: Returned when the size of the block to decrypt is larger than the file.
- Cause: The size of block to be decrypted is larger than the file.
- Workarounds: Check whether the passed dwDecryptSize is larger than the file size.

ERROR_HCRYPTLIB_FREAD_DECRYPT_FREAD (Value = 0x0001B007)

- Description: Returned when file reading fails.
- Cause: The file handle is not normal or the file is locked.
- Workarounds: Check whether the file handle or other process is trying to access the file.

ERROR_HSCRYPTLIB_FREAD_DECRYPT_GETDECMMSG**(Value = 0x0001B008)**

- Description: Returned when message decryption fails.
- Cause: API Internal error.
- Workarounds: Please contact AhnLab, Inc.

ERROR_HSCRYPTLIB_EXCEPTION (Value = 0x0001B001)

- Description: Returned when an exception occurs.
- Cause:
- Workarounds: Please contact AhnLab, Inc.

Others

WIN32 Defined Error (Value = WIN32 Defined)

7.2. User Rights Support

7.2.1. Overview

HsUserUtil provides various functions which enables game hacking protection functions of HackShield for the users logged on with the general user account as well as the administrator account on the NT-series OS.

Functions

Support for Non-administrator Account Game

HsUserUtil.lib allows users logged on with the general user account, as well as the administrator account, to run the game client and the game hacking protection functions of HackShield.

Features

Interface Function (API)

Provides interface library for the developer to easily use the functions of HsUserUtil and get the returned results. Using the provided interface library, the game developer can allow general user accounts to run the game client and HackShield depending on the game developer's policies.

Test Program

Provides HsUserUtilTest.exe, a test game client program, implemented by the APIs of HsUserUtil. Game developers can check the functions and the example code of HsUserUtil by referring to the test program and easily develop client programs.

System Architecture

HsUserUtil is provided as an SDL library, not as an independent execution file. General architecture and operating principles of HsUserUtil are as follows:

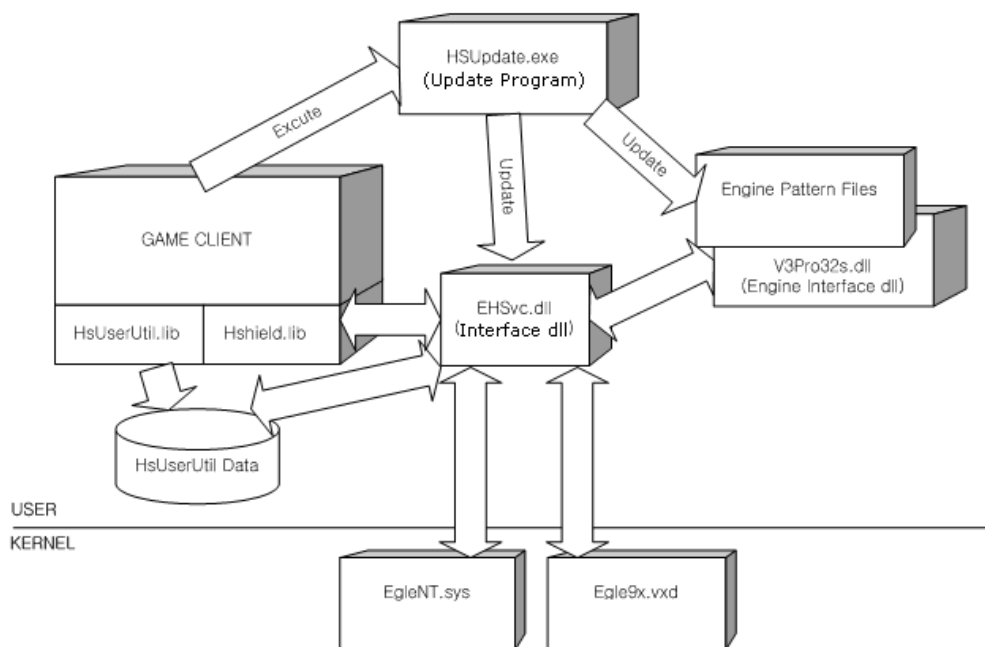


Fig. 7-2 General Architecture and Operating Principles of HsUserUtil

HsUserUtil.lib (Interface library)

An interface file, which provides APIs which allows a general user account to start the game program

HsUserUtil Data

Saves the shadow account information and related data which allows the general user account to start the game program.

EhSvc.dll (HackShield service module)

HackShield service module which is loaded to the actual game program to perform the hacking protection function. Supports the game and the game protection function through the use of HsUserUtil data in case the game has been running with the user permissions.

7.2.2. Application Programming

Programming Procedure

The game client developer can implement HsUserUtil functions in the following

order.

Note

This document describes an example in which HSUserUtil is used for the game client program. If there is a game launcher program besides the game client program, the game developer can also use HSUserUtil library in the game launcher program. HSUserUtil library shall be applied according to the game structure.

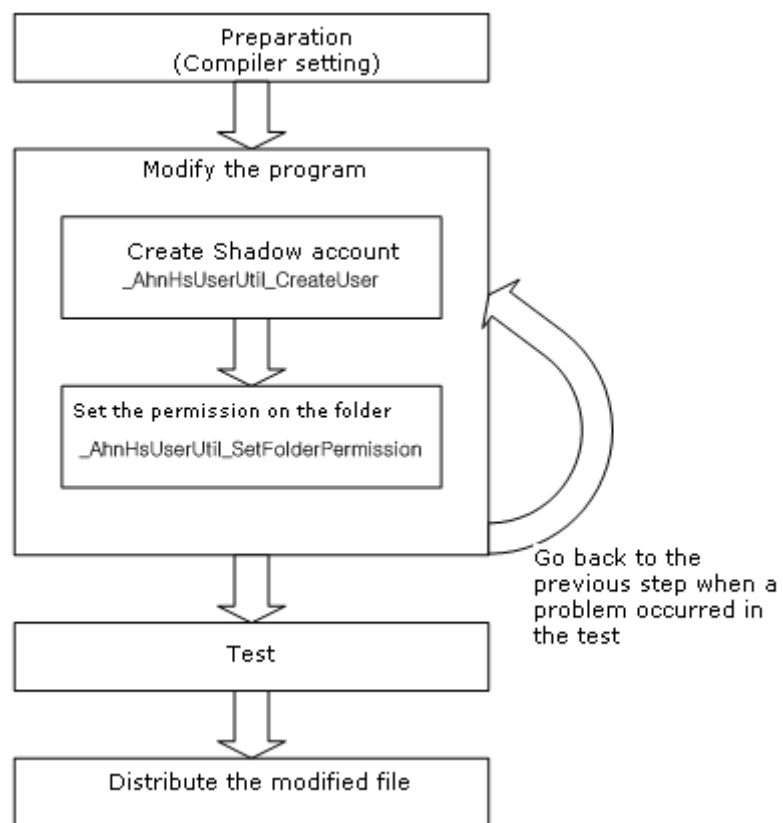


Fig. 7-3 HsUserUtil programming order

1. Preparation: Check the list of the provided HsUserUtil file, and copy necessary files.
2. Calling shadow account creation function: Write a shadow account creation function calling function so that the general user account can execute the game and use the game hacking protection function.
3. Calling NTFS permission setting function: Write an NTFS permission setting function calling code after calling the service starting function so that general user account can also write files for the game.
4. Test whether the source code is properly operating.

5. Distribute to the client users.

Preparation

Follow the steps below by using HsUserUtil before starting programming:

7.2.2.1. HsUserUtil File

HsUserUtil File

Table 7-2 HsUserUtil File

File name	Installation folder	Description
HsUserUtil.h	[Game folder] source	Header file
HsUserUtil.lib	[Game folder] source	Library file

Compiler Setting

The project file of the game client program which uses HsUserUtil must include HsUserUtil.lib file in the library or the source code list.

`_AhnHsUserUtil_CreateUser`

When it is ready for programming, call `_AhnHsUserUtil_CreateUser` first. After `_AhnHsUserUtil_CreateUser` function is successfully called, the general user account can execute the game and use the game hacking protection function.

At this time, `_AhnHsUserUtil_CreateUser` function is called by the administrator account in the game program or game launcher program.

The following is an example of calling `_AhnHsUserUtil_CreateUser` function.

Example

```
dwRet = _AhnHsUserUtil_CreateUser ( );
```

Note

`_AhnHsUserUtil_CreateUser` creates a new user account when the shadow account user information does not exist or the login was not with the existing shadow account user information. A logic which deletes an account complying with the shadow account naming rule is included so that unnecessary accounts or unused accounts will be automatically deleted when a new shadow account is created.

`_AhnHsUserUtil_SetFolderPermission`

If the game program is installed in the NTFS volume, the general user account has no file write access; may not normally run the game. For example, even when the update program for the game module was executed, general user account may be not allowed to write the latest file in the game folder or may fail to write the game data during game execution.

In order to allow the general user account to have the file write access in the game installation folder, the game developer shall use `_AhnHsUserUtil_SetFolderPermission` function and give the NTFS write access. The following is an example of calling this function.

Example

```
dwRet = _AhnHsUserUtil_SetFolderPermission ("game installation path");
```

In order to give the NTFS write access to the general user account, the full path shall be given. In case the path is an alternative path or incorrect path, malfunction may occur.

If the game installation path is not the NTFS volume or the function is called in Windows 95, 98, or ME PC which does not use the NTFS file system, the function will be invalid.

When this function is executed, the NTFS write access will be given to the user group for the corresponding path. However, this function shall be called by the administrative privileges.

Caution

The game developer shall note that the game client program execution path differs depending on the user. Changing the NTFS permission for the folder without knowing where the game will be installed – C:\, Desktop, or Windows folder – may cause security vulnerabilities.

`_AhnHsUserUtil_DeleteUser`

Deletes the shadow account that has been created through

AhnHsUserUtil_CreateUser
function.

Example

```
dwRet = _AhnHsUserUtil_DeleteUser ();
```

AhnHsUserUtil_IsEnableHSAdminRights

Checks whether the account logged in has permission to run HackShield.

Example

```
dwRet = _AhnHsUserUtil_IsEnableHSAdminRights ();
```

AhnHsUserUtil_CheckHSShadowAccount

Checks whether the shadow account created through AhnHsUserUtil_CreateUser function is properly registered in the system

Example

```
DWORD dwRet = HSUSERUTIL_ERR_OK;

dwRet = _AhnHsUserUtil_CheckHSShadowAccount();

switch ( dwRet )
{
    case HSUSERUTIL_ERR_NOT_NT:
        AfxMessageBox ( "HSUSERUTIL_ERR_NOT_NT" );
        break;
    case HSUSERUTIL_ERR_OK:
        AfxMessageBox ( "HSUSERUTIL_ERR_OK" );
        break;
    case HSUSERUTIL_ERR_SHADOWACNT_NOT_EXIST:
        AfxMessageBox ( "HSUSERUTIL_ERR_SHADOWACNT_NOT_EXIST" );
        break;
    default:
        AfxMessageBox ( "HSUSERUTIL_ERR_UNKNOWN" );
        break;
}
```

This function is used to check whether shadow account has been created.

`_AhnHSUserUtil_IsAdmin`

Checks whether the account logged in current has administrative privileges.

Example

```
if ( TRUE == _AhnHSUserUtil_IsAdmin() )
{
    AfxMessageBox ( "TRUE" );
}
else
{
    AfxMessageBox ( "FALSE" );
}
```

7.2.3. Application Programming Interface

`_AhnHsUserUtil_CreateUser`

DESCRIPTION

Creates a shadow account in order to prevent game hacking for the users logged in with general user rights.

SYNTAX

```
DWORD __stdcall
_AhnHsUserUtil_CreateUser ( );
```

PARAMETERS

None.

RETURN VALUE

HSUSERUTIL_ERR_OK (Value = 0x00000000)

- Description: Returned after the shadow account is successfully created.

- Cause: None
- Workarounds: None

HSUSERUTIL_ERR_NOT_ADMIN (Value = 0x0005A002)

- Description: There is No administrative privileges.
- Cause: The account logged-in currently is not the administrator account.
- Workarounds: This error shall not be handled when `_AhnHsUserUtil_CreateUser` can be called under the user permissions; the case may be varied depending on the application type.

HSUSERUTIL_ERR_NOT_NT (Value = 0x0005A003)

- Description: Not NT.
- Cause: The system is not NT series.
- Workarounds: Check whether the OS version is NT or later.

HSUSERUTIL_ERR_DELSHADOWACNT_FAIL (Value = 0x0005A005)

- Description: Returned when shadow account was not deleted.
- Cause: Internal error.
- Workarounds: Please contact AhnLab, Inc.

HSUSERUTIL_ERR_DELHIDEIDINFO_FAIL (Value = 0x0005A006)

- Description: Returned when shadow account deletion fails on the Windows XP screen.
- Cause: Internal error.
- Workarounds: Please contact AhnLab, Inc.

HSUSERUTIL_ERR_DELSHADOWACNTINFO_FAIL (Value = 0x0005A007)

- Description: Returned when shadow account deletion failed.
- Cause: Internal error.

- Workarounds: Please contact AhnLab, Inc.

HSUSERUTIL_ERR_ADDSHADOWACNT_FAIL (Value = 0x0005A008)

- Description: Returned when a shadow account was not created.
- Cause: Internal error.
- Workarounds: Please contact AhnLab, Inc.

REMARKS

_AhnHsUserUtil_CreateUser function can create a shadow account while the account is logged in with administrative privileges. In order to use the game protection function of HackShield with the user permission, this function shall be called at least once and a shadow account shall be created.

_AhnHsUserUtil_SetFolderPermission

DESCRIPTION

Gives NTFS write permission to the general user accounts in the directory where the game client is installed; users logged on with general user accounts can write files for update and execution.

SYNTAX

```
DWORD __stdcall  
_AhnHsUserUtil_SetFolderPermission (  
    LPTSTR szPath  
);
```

PARAMETERS

Parameter	Value	Description
szPath		Full path to set the NTFS permission Example: C:\Program Files\My Company\My Game

RETURN VALUE

HSUSERUTIL_ERR_OK (Value = 0x00000000)

- Description: Returned when NTFS authority was not given to the folder.
- Cause:
- Workarounds:

HSUSERUTIL_ERR_NOT_ADMIN (Value = 0x0005A002)

- Description: No administrative privileges.
- Cause: The account logged-in currently is not the administrator account.
- Workarounds: This error shall not be handled when _AhnHsUserUtil_CreateUser can be called under the user permissions; the case may be varied depending on the application type.

HSUSERUTIL_ERR_GETVOLUMEINFO_FAIL (Value = 0005A111)

- Description: Unable to read driver volume information.
- Cause: A problem occurred in the process of reading driver volume information.
- Workarounds: Please contact AhnLab, Inc.

REMARKS

When a user can define the path of the game client program, this function shall be used with caution. If the user installed the game in the root directory such as C: or D: drive, calling this function will give the NTFS permission to the entire drive. This may create security vulnerabilities so that this function shall be used with caution.

In case the folder has multiple folders and files, calling this function may take a few seconds or a few minutes to set the NTFS permission. However, this may occur only during the NTFS permission is initially given and does not cause any influence once the permission is given.

AhnHsUserUtil_DeleteUser

DESCRIPTION

Deletes the shadow account that has been created through _AhnHsUserUtil_CreateUser.

SYNTAX

```
DWORD __stdcall _AhnHsUserUtil_DeleteUser (void);
```

PARAMETERS

Parameter	Value	Description
void		

RETURN VALUE

HSUSERUTIL_ERR_OK (Value = 0x00000000)

- Description: Returned after successful deletion of shadow account.

- Cause:
- Workarounds:

HSUSERUTIL_ERR_LOADDLL_FAIL (Value = 0x0005A004)

- Description: DLL required to perform the feature cannot be loaded.
- Cause: NETAPI32.DLL or ADVAPI32.DLL does not exist or is damaged in the system folder.
- Workarounds: Check whether the dll exists properly.

HSUSERUTIL_ERR_NOT_NT (Value = 0x0005A003)

- Description: The OS version is not NT.
- Cause: The system is not NT series.
- Workarounds: Check the OS version.

REMARKS

_AhnHsUserUtil_DeleteUser function
deletes the shadow account created by _AhnHsUserUtil_CreateUser.

_AhnHsUserUtil_IsEnableHSAdminRights

DESCRIPTION

Checks whether the logged-in account has permission to run HackShield.

("Success" is returned when the account has administrative privileges or has user permissions as well as HackShield shadow account has been created.)

SYNTAX

DWORD __stdcall _AhnHsUserUtil_IsEnableHSAdminRights(void);

PARAMETERS

Parameter	Value	Description
void		

RETURN VALUE

HSUSERUTIL_ERR_OK (Value = 0x00000000)

- Description: Returned value when shadow account is properly registered.
- Cause: None
- Workarounds: None

HSUSERUTIL_ERR_NOT_NT (Value = 0x0005A003)

- Description: The OS version is not NT.
- Cause: Tthe system is not NT series.
- Workarounds: Check the OS version.

REMARKS

Checks if the currently logged-in account has permissions to run HackShield. If the account has administrative privileges, or user permission as well as HackShield shadow account has been created, "Success" is returned. And if not, another error is returned.

Use _AhnHsUserUtil
_CheckHSShadowAccount function to check whether HackShield shadow account exists.

[_AhnHsUserUtil_CheckHSShadowAccount](#)

DESCRIPTION

Check whether the shadow account that had been created through `_AhnHsUserUtil_CreateUser` has been properly registered.

SYNTAX

DWORD `__stdcall _AhnHsUserUtil_CheckHSShadowAccount();`

PARAMETERS

Parameter	Value	Description
void		

RETURN VALUE

HSUSERUTIL_ERR_OK (Value = 0x00000000)

- Description: Returned value when shadow account is properly registered.
- Cause: None
- Workarounds: None

HSUSERUTIL_ERR_NOT_NT (Value = 0x0005A003)

- Description: The OS version is not NT.
- Cause: The system is not NT series.
- Workarounds: Check the OS version.

HSUSERUTIL_ERR_SHADOWACNT_NOT_EXIST (Value = 0x0005A009)

- Description: HackShield shadow account does not exist.
- Cause: HackShield shadow account is created in the current system. (Check whether the shadow account is registered through the error.)
- Workarounds: Check whether HackShield shadow account has been created properly through `_AhnHsUserUtil_CreateUser` function.

REMARKS

_AhnHsUserUtil_CheckHSShadowAccount function checks whether the shadow account created by _AhnHsUserUtil_CreateUser has been properly registered

_AhnHSUserUtil_IsAdmin

DESCRIPTION

Checks whether the currently logged-in account has administrative privileges.

SYNTAX

```
BOOL __stdcall _AhnHSUserUtil_IsAdmin ();
```

PARAMETERS

Parameter	Value	Description
void		

RETURN VALUE

TRUE

Not NT series, or NT series with administrative privileges.

FALSE

NT series and No administrative privileges.

REMARKS

8. Tools

8.1. HSBGen Tool (For HackShield 4.2 or later)

Functions

The user can check the game file, memory status, and integrity of HackShield by creating HackShield Briefcase File (AntiCpX.hsb) in the server. The user can control previous version by patching a new version or setting options without restarting the server.

Each time, different HSB information files are created. If there are multiple game servers, different HSB information files shall be applied for the stronger security. However, same game clients shall have same memory data.

Add information needed for game client distributed to use enhanced LMP feature. Provide the feature that was provided with CSInspector.exe tool with HSBGen tool.

System Environment

OS	Platform
Windows 2K3, XP(above SP1), VISTA, 7	x86, x64

Using HSBGen Tool

Creating HSB Information File Based on Manual UI Setting

Create AntiCpX.hsb file using \Bin\Win\x86\AntiCrack\HSBGen.exe.

Caution

The original file before the packing process shall be used as the client executable file.

Caution

When distributing the client executable file with digital signature, sign the game client executable file digitally after using HSBGen tool.

The user can create AntiCpX.hsb file using HSBGen.exe.

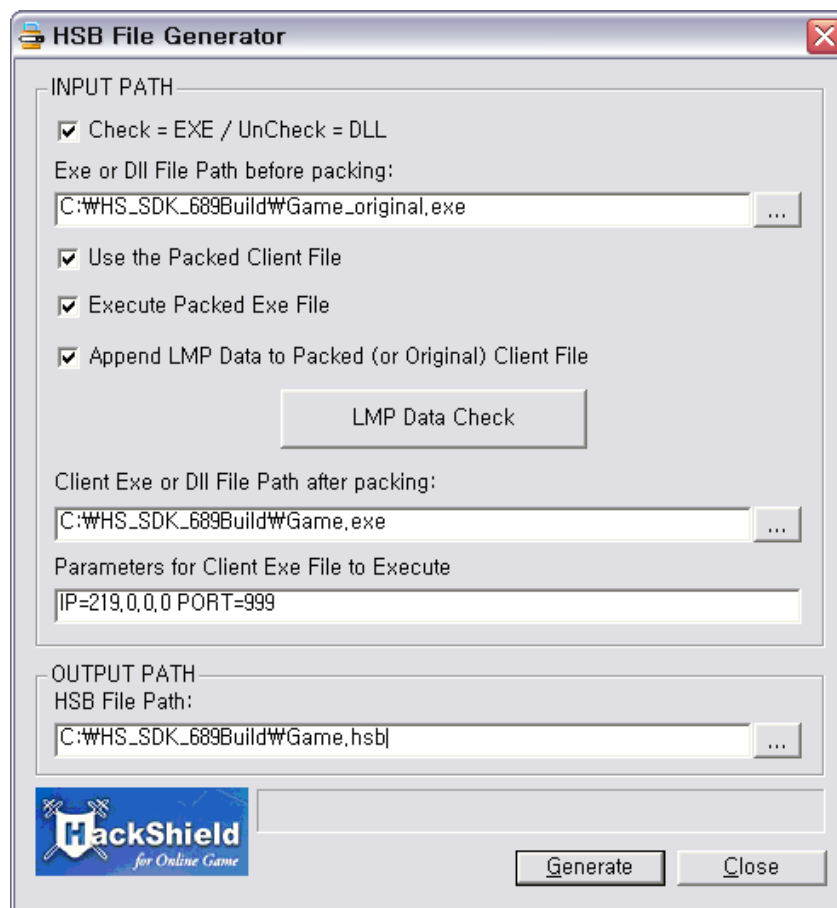


Fig. 8-1 HSB File Generator

1. Check if the target file is EXE, and uncheck if the file is DLL.

Caution.

: If the target file is DLL, the LMP information in the file will be automatically added.

2. In case AhnLab provided HSBGen.ini file, the file shall be saved in the same folder with HSBGen.exe. (Extended server-side detection shall be set differently for each game.)
3. Enter the path of the original executable file which has not been packed yet in the **Exe or Dll File Path before packing** field. Select a file by clicking on the [...] button.
4. In order to distribute a packed client executable file, input the path of the unpacked file.
5. Select **Use the Packed Client File** field, and input the path of the packed file in **Client Exe or Dll File Path after packing** field.
6. **Execute Packed Client File** automatically runs executable file entered in Client Executable File Path after packing and decides whether to create

hsb file.

Caution.

- When using the Executable Packed Client File option, prepare the file required to execute the game or the parameter information to run the game by clicking the hsb creation button.

7. If there are values needed to run the executable file when running the file entered in the Execute Packed Client File, enter the value in **Parameters for Client Executable File to Execute**.
8. For **Append LMP Data to Packed (or Original) Client File**, add LMP information to the game client file to use the LMP feature. The LMP information can be added to the unpacked original executable file or the packed client executable file.

Caution.

If the file is packed, "execute clientfile" must be selected.

9. Set the full path of **AntiCpX.hsb** file in **HSB File Path** field. Click on [...] button in order to open the file creation window. (Note: In order to set the same folder as that of the executable file, a warning message that the client file will be also distributed will pop up, which is Normal.)
10. After completing all setting, click on the **Generate** button. The progress state will be displayed and the result message will be displayed.

After anticpx.hsb file is created, patch the anticpx.hsb file to the server and patch the game client file which was used to create anticpx.hsb file.

Automatically Creating HSB Information File

HSBGen specifies argument data necessary for creation of HSB file and provides automatic HSB file creation function. The following describes the usage:

Usage:

HSBGen.exe

[Original game path] [packing] [packed game path] [file execution] [parameter] [HSB path] [file type]

[LMP]

[Original game path] Path of game to extract HSB information

[Packing] Packed executable file supported/not supported (1 = supported / 0 = not supported)

[Packed game path] Path of packed game to extract HSB information (Only when (2) is 1)

[File execution] Packed executable file executed/not executed (1 = Executed / 0 = Not executed)

- Run if packed with packer (eg. Themida) with code manipulation prevention feature

[Parameter] Packed file execution parameter

- Only used when (4) is 1. Ignore if (4) is 0.

[HSB path] HSB file path

[File type] File type to protect (1= EXE , 0 = DLL)

[LMP] LMP information added/not added (1 = Added, 0 = Not added)

Caution

Argument information is separated by ';', and space is not allowed before/after the separator.

–'”' is not allowed even though the file path contains spaces.

[Original game path] C:\HS_SDK\Amazon_ori.exe

[Packed game path] C:\HS_SDK\Amazon.exe

[HSB file path]] C:\HS_SDK\AntiCrack\AntiCrack.hsb

[Parameter] aaa bbb

No.	1	2	3	4	5	6	7	9
Packing	Y	Y	Y	Y	N	N	Y	N
File execution	Y	Y	N	N	N	N	N	N
File type	Y	Y	Y	Y	Y	Y	N	N
LMP applied	Y	N	Y	N	Y	N	Y	Y

Example 1:>

HSBGen.exe [game path] [packed executable file supported:1] [packed game path]
[packed executable file executed:1]

[Parameter] [HSB path] [File type:1] [LMP:1]

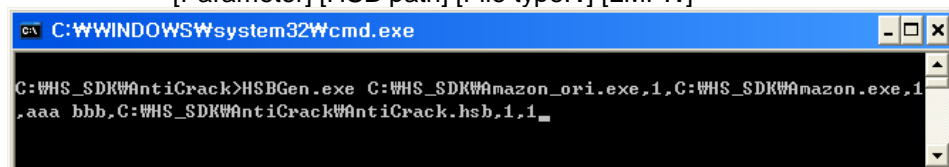


Fig. 8-2 Command-line HSBGen.exe ([Packing:1],[Execution:1], [EXE:1], [LMP:1])

Example 2:>

HSBGen.exe [game path] [packed executable file supported:1] [packed game path]
[packed executable file executed:1]

[Parameter] [HSB path] [File type:1] [LMP:0]

Fig. 8-3 Command-line HSBGen.exe ([Packing:1],[Execution:1], [EXE:1], [LMP:0])

Example 3:>

HSBGen.exe [game path] [packed executable file supported:1] [packed game path]
[packed executable file executed:0]
[Parameter] [HSB path] [File type:1] [LMP:1]

Fig. 8-4 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:1], [LMP:1])

Example 4:>

HSBGen.exe [game path] [packed executable file supported:1] [packed game path]
[packed executable file executed:0]
[Parameter] [HSB path] [File type:1] [LMP:0]

Fig. 8-5 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:1], [LMP:0])

Example 5:>

HSBGen.exe [game path] [packed executable file supported:0][HSB path] [file
type:1]
[LMP:1]

Fig. 8-6 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:1], [LMP:1])

Example 6:>

HSBGen.exe [game path] [packed executable file supported:0][HSB path] [file
type:1]
[LMP:0]

Fig. 8-7 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:1], [LMP:0])

Example 7:>
 HSBGen.exe [game path] [packed executable file supported:1] [packed game path]
 [packed executable file executed:0]
 [HSB path:0] [File type:0] [LMP:1]

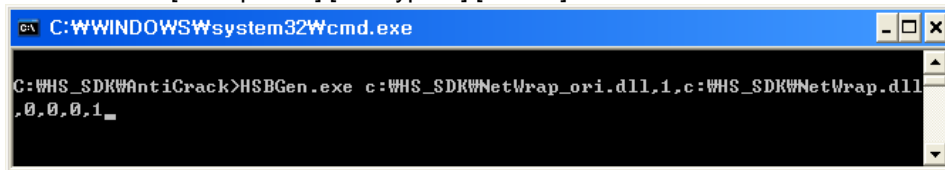


Fig. 8-8 Command-line HSBGen.exe ([Packing:1],[Execution:0], [EXE:0], [LMP:1])

Example 8:>
 HSBGen.exe [game path] [packed executable file supported:0] [HSB path:0] [file
 type:0]
 [LMP:1]

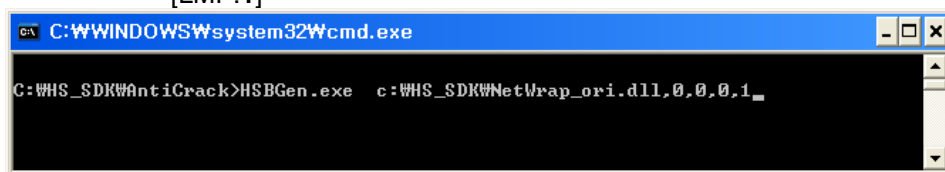
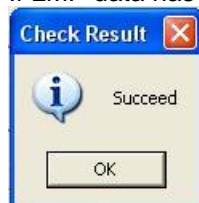


Fig. 8-9 Command-line HSBGen.exe ([Packing:0],[Execution:0], [EXE:0], [LMP:1])

Check LMP Data

1. Enter the path of the file you want to check LMP data in the **Exe or DI I File Path before packing** field. Select a file by clicking on the [...] button.
2. Click **LMP Data Check**.
3. If LMP data has been appended to the file, the following message appears:



4. If LMP data has not been appended, the following message appears:



HSBGen.ini description

```

[VERCT]

GrantOldSession=1
MaxAllowedNumber=1

[REQRE]
InitStep1=1
InitStep2=2
InitStep3=8
InitStep4=4

[DELAY]
UseDelayedSpiking=0
MinDelayCount=1
MaxDelayCount=3

[CKMEM]
PageGroupSize=40
QueryPages=10

[FPATH]
ClientFileName=D:\ \game_ori.exe
UsePackedClientFile=1
PackedClientFileName=D:\ \game_packed.exe
GetInfoFromRunningEXE=1
Parameters=-d
AppendLMPInfoToClientFile=1
HsbFileName=D:\ \anticpx.hsb
UseEXE=0

[CKHSB]
UseHSB=1

```

1. VERCT

GrantOldSession

The above can be 1 or 0. If it is '1', it allows the client version for hsb file uploaded before along with the client version for the current hsb file.

MaxAllowedNumber

It is significant only when GrantOldSession is 1, and it must always be above 1. For instance, when it is '1', it means that hsb will be allowed in the currently applied client only, and if it is '5', it means that up to four (4) client versions will be allowed for the hsb file.

2. REQRE

There are five scanning features for server interaction; GUID scan, client file scan, HackShield module scan, memory scan and HackShield status scan.

GUID Scan:	1
Client Field Scan:	2
Memory Scan:	4
HackShield Engine Scan:	8

HackShield Status Scan: 16

※ Even if 'HackShield Status Scan' is not stated, it will be applied to all steps (InitStep1, InitStep2, InitStep3, InitStep4) by default.

Each scan option is 1, 2, 4, 8 and 16 bit, and can be used simultaneously.

InitStep 1

GUID scan(1) or simultaneous scans(17) for GUID(1) and HackShield status(16) can be used.

InitStep 2

GUID scan(1) and client file scan(2) can be used.

InitStep 3

All scans can be used.

For instance, '31' means all scans will be used, and '5' means to scan the memory and HackShield module.

InitStep 4

Specifies scan to regularly repeat after the above three steps are completed. Use '20', which is HackShield status scan and memory scan.

3. DELAY

UseDelayedSpiking

To use the feature with '1', it causes confusion to the hackers by delaying the schedule and not causing error right away when returning the error from _AhnHS_VerifyResponse function. This function is not used when the value is set to '0'.

MinDelayCount, MaxDelayCount

When delay spiking feature is applied, delay is applied randomly between MinDelayCount and MaxDelayCount. 1 is a cycle of the function being called.

4. CKMEM

PageGroupSize , QueryPages

Option to specify the number of memory pages to scan at one time when scanning the memory. It is recommended to use this function by default.

5. FPATH

Game client settings information to create hsb file, a memory crc information file of the game client. This information is entered by the user through HSBGen.exe tool.

ClientFileName

Path where unpacked game client is located.

UsePackedClientFile

Setting to specify whether to use packed game client.

When using packed game client, the value is 1, and if not, the value is 0.

PackedClientFileName

Path where packed game client is located.

This information is required when UsePackedClientFile is set to 1.

GetInfoFromRunningEXE

Settings to extract memory crc information by running the game client when using packed game client.

If this value is 1, memory crc information is extracted by running the file in the path set in PackedClientFileName.

Parameters

Parameter information needed to execute the game client.

It is valid when the path is set in PackedClientFileName, and the GetInfoFromRunningEXE is 1.

AppendLMPInfoToClientFile

Settings to add the required information to the game client file to use the LMP.

If the value is '1', the information needed to use the LMP feature is added to the game client, and if the value is '0', it is not proceeded.

HsbFileName

The path to create memory crc information file (hsb file) of the game client. If you do not set this value, .hsb file cannot be created. (You must set the extension to .hsb.)

UseEXE

Set to 1 if the file is EXE and 0 if the file is DLL.

Caution.

There is no FPATH path when the hsbgen.ini file in SDK is opened. The FPATH information is saved from the first time the hsb file is created.

6. CKHSB**UseHSB**

Setting on whether to upload the hsb file, created by executing HSBGen.exe tool with the game client file when new game client file is distributed, to the game server.

If the value is '0' (UseHSB=0), the hsb file created with the new game client file is not uploaded in the game server and interoperates with the extended server. If the value is 1, the hsb file created with the new game client file is uploaded in the game server and interoperates with the extended server.

Caution when UseHSB=0 in [CKHSB] section

The hsb file with the above section applied must be uploaded to the server (once in the beginning).

It is required to check if the hsb file has been created normally with HSBGen tool, before distributing the new game client file (however, it is not required to upload the hsb file to the game server.)

As creating the hsb file, the contents in HSBGen.ini file is continuously reflected to the game client to be connected later. Therefore, pay attention to the contents in the HSBGen.ini file in the initial creation.

'Client file scan' and 'memory scan' are not supported. Apply numbers, except "2" and "4" that mean client file scan and memory scan, in InitStep2', 'InitStep3', and 'InitStep4' of [REQRE] section.

```
Ex> InitStep1=1
      InitStep2=1
      InitStep3=8
      InitStep4=1
```

Caution

The content of hsb file that has been newly created with HSBGen.exe tool is applied after the game client file, entered as creating the hsb file, accesses the server.

In other words, it does not mean the data is reflected just because the newly created hsb file is uploaded on the game server. It is applied only when the game client file, which has been entered when creating the hsb file, must be connected at least once to the game server.

8.2. HSUpSetEnv Tool (For HackShield 5.1 or later)

Functions

Creates an update setting file (HSUpdate.env).
The user can select FTP or HTTP and use multiple URLs.

Using HSUpSetEnv Tool

Creating HSUpdate.env File

Creates HSUpdate.env file using [HackShield SDK]\Bin\Win\x86\Util\HSUpSetEnv.exe.

HSUpSetEnv tool is formed of two tabs, which as [General information] and [Extended information]. They can be used as follows:

Usage:

Execute [HackShield SDK]\Bin\Win\x86\Util\HSUpSetEnv.exe file.

Caution

HSUpSetEnv.exe is a tool which creates an update configuration file. You are not allowed to distribute this tool.

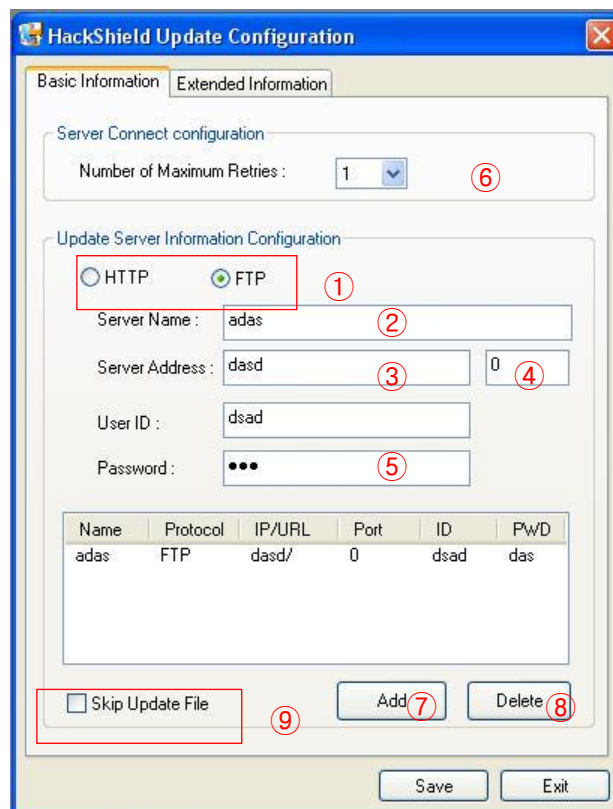


Fig. 8-10 HSUpSetEnv.exe Basic Information tab

- ① Select HTTP or FTP depending on the protocol that the update server will use.

- ② Input the update server name.
- ③ Input the server address. Specify the folder where patch set is saved.
- ④ Input the previously set port number.
- ⑤ Set account information for the FTP. Otherwise, anonymous connection will be made.
- ⑥ Set the number of retry attempts when update is failed. When access of the entire list has failed, attempts will be made according to the set number.
- ⑦ Add the server to the server list by clicking on the **Add** button. In case multiple update servers are needed, repeat the above procedures.
- ⑧ Delete a server on the server list by clicking **Delete** button.
- ⑨ Exclude HackShield update files when downloading the patch.
When this option is selected, the following files will not be installed.
(HackShield update-related files: 'ahnupctl.dll', 'ahnupgs.dll', 'hsinst.dll', 'hsupdate.exe', 'v3hunt.dll', 'v3inetgs.dll')

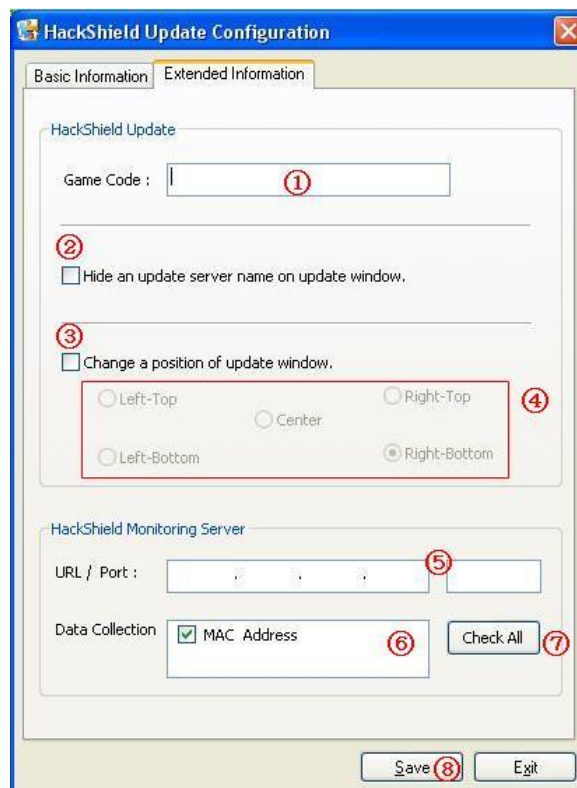
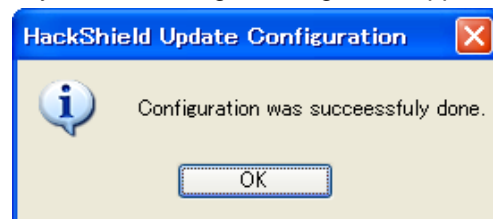


Fig. 8-12 HSUpSetEnv.exe Extension Information tab

- ① Enter the Game Code. It is valid only when `_AhnHS_HSUpdateEx` function is used. You do not need to enter when using `_AhnHS_HSUpdate`.
- ② Select this button if you do not want to display the update server name on the bottom right of the update image (HSUPDATE.JPG) used in the

window that shows the update progress.

- ③ Select this button to change the position of the update image (HSUPDATE.JPG) used in the window that shows the update progress. If you do not change the position, it will be displayed on the bottom right.
- ④ If you select to change the position of the update image (HSUPDATE.JPG) used in the window that shows the update progress, it will be activated and you may change the position.
- ⑤ When using HackShield Monitor, enter the monitoring IP / Port. (Only enter the IP / Port. To use the monitoring feature, you must call the _AhnHS_StartMonitor function.)
- ⑥ When using HackShield Monitor, select the additional information to send to the monitoring server.
- ⑦ Select all or deselect all from information list.
- ⑧ Click the **Save** button, and when the HSUpdate.env file is created normally, the following message will appear.



When the settings are complete, click **Exit** to terminate the tool.

8.3. CSInspector Tool

Functions

Specifies a certain EXE file or DLL file to protect. CSInspector shall be executed before the target files are packed. When used with the server interface functions, CSInspector shall be executed before HSBGen.exe. Currently, CSInspector is provided only as a command-line.

CSInspector shall be used with AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION option in the client.

Setting the File Subject to Protection

Specify a file to protect using \Bin\Win\x86\Util\CSInspector.exe. EXE files and DLL files can be set. The following describes the using procedure:

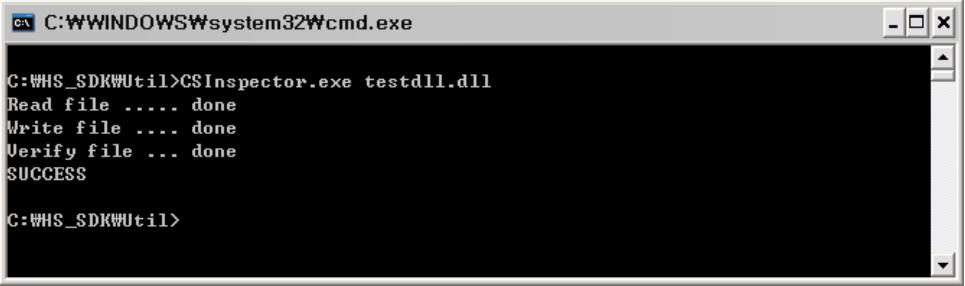
Usage:
CSInspector.exe

Caution

This must be performed before packing or CRC creation.

Example:

If the target file is testdll.dll and CSInspector.exe is under C:\HS_SDK\Util folder, do the following:



```
C:\WINDOWS\system32\cmd.exe

C:\HS_SDK\Util>CSInspector.exe testdll.dll
Read file ..... done
Write file .... done
Verify file ... done
SUCCESS

C:\HS_SDK\Util>
```

Fig. 8-13 CSInspector.exe

8.4. SetServerList Tool (For HackShield 5.1 or later)

Functions

ANTIFREESERVER feature supports two detection methods - Whitelist and Blacklist.

(Select one from the two. The method you select will be applied to the list.)

Both methods must be used with AHNHS_CHKOPT_ANTIFREESERVER option in the client.

With the Whitelist method, detection callback will occur if the game process attempts to connect from an IP address that is not in the specified IP range. Callback will also occur when connecting from local IP (127.0.0.1). It is to prevent hacking tool that connects from Bot server by hacking the game that tries to connected to the game server.

With the Blacklist method, callback will occur if any process attempts to connect from an IP address in the blocked IP address range. It is to prevent hacking tool that communicates with the server.

If detected by the Blacklist, there will be no client game callback. AhnHS_VerifyResponseEx_WithInfo function will be called, and returned to the following parameter to terminate it.

-ulError : ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED
-ulSpecificError: AHNHS_ACTAPC_DETECT_ANTIFREESERVER 0x10910)

8.5. Using SetServerList Tool

afs.dat File Creation

Create afs.dat file by using [HackShield SDK]\Bin\Win\x86\Util\SetServerList.exe. The following describes the using procedure:

Usage:

Execute [HackShield SDK]\Bin\Win\x86\Util\SetServerList.exe file.

Caution

SetServerList.exe is a tool which creates an update configuration file. You are not allowed to distribute this tool.



Fig. 8-14 SetServerList.exe

- ① Click the **Insert** button to enter the IP address.

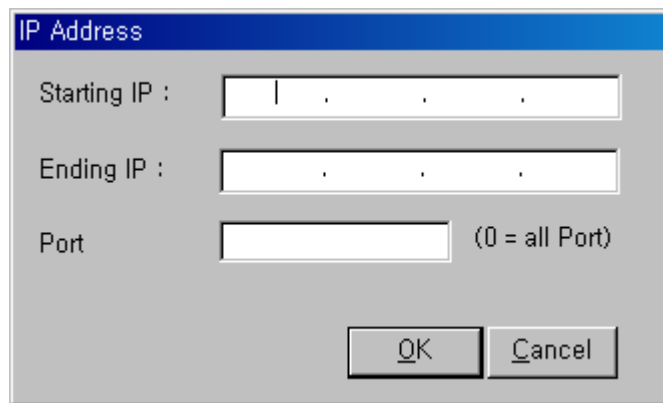


Fig. 8-15 SetServerList tool's address field

- ② Specify a valid IP address range. Enter a starting IP address and ending IP address. To add only one IP address, the starting and ending IP address must be the same.
- ③ Port can only be specified when a Blacklist-based method is being used. If a session makes connection from the port within the specified range, the following error will be sent to the game server to terminate the game.

:ERROR_ANTICPXSVR_DETECT_CALLBACK_IS_NOTIFIED

But, if the port is 0, this will apply to all ports.

- ④ Click OK to add the IP address entered in ②.
- ⑤ Double-click or select the IP address to modify on the list, and click Modify to modify the IP address.
- ⑥ Select the IP address to delete on the list and click Delete to delete the IP address.
- ⑦ Click Save to save the IP address and port number in the afs.dat file and terminate the program.. If you do not click Save, the modification will not be applied. You must click Save after modifying the IP address.
- ⑧ When distributing afs.dat file, it must be distributed to the same location as the HackShield module. (E.g. [Game Directory]/hshield/afs.dat)
- ⑨ Make sure you do not distribute SerServerList.exe.

afs.dat File Distribution

The game developer shall manage and distribute the afs.dat file.

If the game developer cannot distribute the afs.dat file, HackShield update can distribute it.

(Even if the afs.dat file is distributed through HackShield update, the file must be managed by the game developer.)

If you save the afs.dat file under the HackShield patchset (same location as ahn.ui and autoup.exe) in the update server, it will be automatically downloaded when updating HackShield.

Caution

For further information on the HackShield update structure, refer to the following section in the manual:

[HackShield update] – [System Architecture]

8.6. Using HSBHelper Tool

Functions

Use \Bin\Win\x86\AntiCrack\HSBHelper to check whether the HSB file and the corresponding game client file match. The following describes the using procedure:

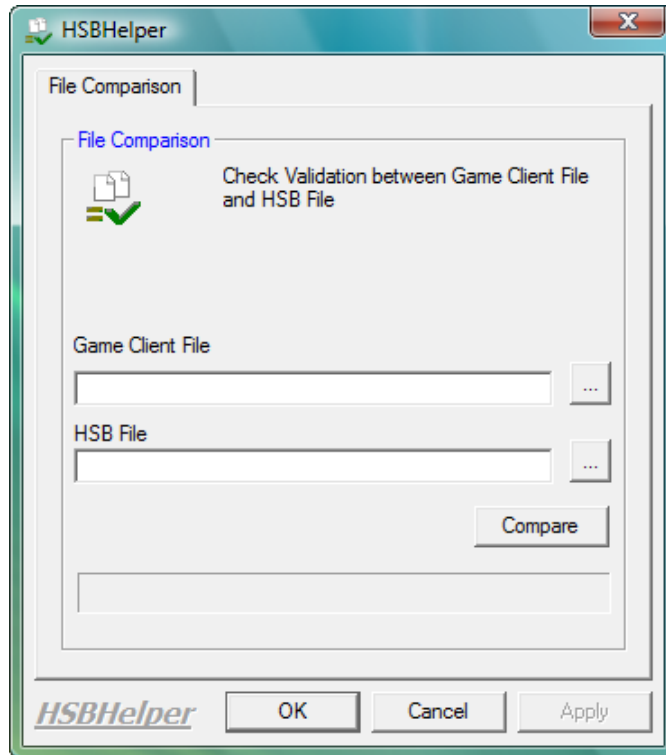
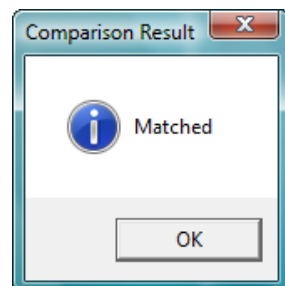


Fig. 8-16 Tool

- ① Enter the full path of the game client file.
(e.g. C:\ R.5.1.41.1(build 671)\Bin\Win\x86\Amazon.exe)
- ② Enter the full path of the HSB file.
(e.g. C:\ R.5.1.41.1(build 671)\Bin\Win\x86\AntiCrack\AntiCpX.hsb)
- ③ Click Compare.
- ④ If the files match, the following message box will appear.



- ⑤ If not, the following message box will appear.



Caution

When running HSBHelper, HSBHelper.log log file will always be created.

Run Command Line

Use a command-line to check whether the HSB file and the corresponding game client file match. The following describes the using procedure:

Usage:

HSBHelper.exe (1),(2)

(1) : Path of game to extract HSB information

(2) : HSB file path

Caution

- Arguments are separated by commas ','.
 - It returns 0 if the task was successfully completed. Any other values indicate a failure.
-

9. Appendix

9.1. FAQ

SoftICE is installed in the system, but HackShield is not normalized. How can I set the initialization options for development and debugging?

You must use `HShield.lib`, `Ehsvc.dll`, `hshield.dat` for developers to debug the game client during the development and testing process.

(Please note that there could be a problem in server interoperation if you use `Ehsvc.dll` and `hshield.dat` files along with the release version.)

HackShield Developer version location

- ✓ `EhSvc.dll` – `\SDK\Korean(kr)-SDK\Developer\Bin\Win\x86\HShield`
- ✓ `hshield.dat` - `\SDK\Korean(kr)-SDK\Developer\Bin\Win\x86\HShield`

(Copy to server when debugging when (after) applying server in terface)

- ✓ `HShield.lib` - `\SDK\Korean(kr)-SDK\Developer\Lib\Win\x86\`

(Use appropriate library file for the compile category of the game client project)

Caution

Check the exception setting for VC++ compiler whether set to Microsoft C++ Exception: Stop always, the following error could occur.
"First chance exception in Game.exe(KERNEL32.dll) 0xE0607063 Microsoft C++ Corporation"

Check the exception setting for VC++ compiler whether set to Microsoft C++ Exception: Stop always, and change the setting to Stop if not handled.

How do I use the Manifest feature in Windows Vista?

Note (Manifest)

To create an application that can run normally only with administrative privileges on Vista, you need to add 'you need permission to execute this program' information in the execution file, using Manifest. Users could run the application by right-clicking and selecting Run as Administrator, or go to Properties and select Run as Administrator, but it will be troublesome. So, add the information that 'this application requires Administrators permission' to the executable file itself to automatically give users administrative privileges.

Applying manifest file content (xml) to the game resource

- ① Run Visual Studio6.0.
- ② Add Resource to the game resource folder.
- ③ Enter the number "24" to the new custom resource (Resource type) in Resource.

Note

"24" is a manifest resource value defined in Microsoft.

- ④ Copy the following xml to IDR_DEFAULT1 added in the above step.

```
<?xml version="1.0" encoding="utf-8" ?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity version="1.0.0.0"
    processorArchitecture="X86"
    name="Game"
    type="win32" />
  <description>HspL</description>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level="requireAdministrator" />
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
```

- ⑤ After copying, you will be able to see the binary and ASCII code xml document as below.

000000	3C 3F 78 6D 6C 20 76 65	72 73 69 6F 6E 3D 22 31	<?xml version="1
000010	2E 30 22 20 65 6E 63 6F	64 69 6E 67 3D 22 75 74	.0" encoding="ut
000020	66 2D 38 22 20 3F 3E 0D	0A 3C 61 73 73 65 6D 62	f-8" ?>..<assemb
000030	6C 79 20 78 6D 6C 6E 73	3D 22 75 72 6E 3A 73 63	ly xmlns="urn:sc
000040	68 65 6D 61 73 2D 6D 69	63 72 6F 73 6F 66 74 2D	hemas-microsoft-
000050	63 6F 6D 3A 61 73 6D 2E	76 31 22 20 6D 61 6E 69	com:asm.v1" mani
000060	66 65 73 74 56 65 72 73	69 6F 6E 3D 22 31 2E 30	festVersion="1.0
000070	22 3E 0D 0A 3C 61 73 73	65 6D 62 6C 79 49 64 65	">..<assemblyIde
000080	6E 74 69 74 79 20 76 65	72 73 69 6F 6E 3D 22 31	ntity version="1
000090	2E 30 2E 30 2E 30 22 20	0D 0A 20 20 20 20 70 72	.0.0.0" .. pr
0000a0	6F 63 65 73 73 6F 72 41	72 63 68 69 74 65 63 74	rocessorArchitect
0000b0	75 72 65 3D 22 58 38 36	22 0D 0A 20 20 20 20 6E	ure="X86".. n
0000c0	61 6D 65 3D 22 4D 69 6E	69 41 22 0D 0A 20 20 20	ame="MiniA"..
0000d0	20 74 79 70 65 3D 22 77	69 6E 33 32 22 20 2F 3E	type="win32" />
0000e0	20 0D 0A 20 20 3C 64 65	73 63 72 69 70 74 69 6F	.. <descriptio
0000f0	6E 3E 4D 69 6E 69 41 3C	2F 64 65 73 63 72 69 70	n>MiniA</descrip
000100	74 69 6F 6E 3E 0D 0A 20	20 3C 74 72 75 73 74 49	tion>.. <trustI
000110	6E 66 6F 20 78 6D 6C 6E	73 3D 22 75 72 6E 3A 73	nfo xmlns="urn:s
000120	63 68 65 6D 61 73 2D 6D	69 63 72 6F 73 6F 66 74	chemas-microsoft
000130	2D 63 6F 6D 3A 61 73 6D	2E 76 33 22 3E 0D 0A 20	-com:asm.v3">..
000140	20 20 20 3C 73 65 63 75	72 69 74 79 3E 0D 0A 20	<security>..
000150	20 20 20 20 20 3C 72 65	71 75 65 73 74 65 64 50	<requestedP
000160	72 69 76 69 6C 65 67 65	73 3E 0D 0A 20 20 20 20	rivileges>..
000170	20 20 20 20 20 3C 72	65 71 75 65 73 74 65 64	<requested
000180	45 78 65 63 75 74 69 6F	6E 4C 65 76 65 6C 20 6C	ExecutionLevel 1
000190	65 76 65 6C 3D 22 72 65	71 75 69 72 65 41 64 6D	evel="requireAdm
0001a0	69 6E 69 73 74 72 61 74	6F 72 22 20 20 2F 3E 0D	inistrator" />.
0001b0	0A 20 20 20 20 20 3C	2F 72 65 71 75 65 73 74	. </request
0001c0	65 64 50 72 69 76 69 6C	65 67 65 73 3E 0D 0A 20	edPrivileges>..
0001d0	20 20 20 3C 2F 73 65 63	75 72 69 74 79 3E 0D 0A	</security>..
0001e0	20 20 3C 2F 74 72 75 73	74 49 6E 66 6F 3E 0D 0A	</trustInfo>..
0001f0	3C 2F 61 73 73 65 6D 62	6C 79 3E	</assembly>

- ⑥ Right-click on IDR_DEFAULT1, which is the newly created resource file, and then click **Properties**.

- ⑦ Change "IDR_DEFAULT1" to "1".

Note

To use 24 manifest, the ID must be set to 1.

- ⑧ Save and rebuild the game resource.

Refer to (Why Manifest should be used.)

If a hacker or user maliciously run a hacking tool with administrative privileges, the hacking tool will be executed at administrator privileges, not user permissions. And when the game runs with user permissions, the game will run in lower level, so cannot defend itself against the hacking tool. (The application with lower permission cannot access the higher permission application.) Also, in Vista, permission cannot be elevated while running the process.

To protect the hacking tool with administrative privileges, HackShield must also run at administrative privileges. So, shadow account is not needed in Vista, but manifest feature must be added.

How do I build HackShield Update server?

Can a user logged in without administrative privileges in Windows 2000 or Windows XP use HackShield?

HackShield runs the hack prevention driver at the kernel level. Therefore, in the basic setting, only users with the administrative privileges can run HackShield. In order to use the game hacking prevention function of HackShield with a general user account, a shadow account for HackShield shall be created.

During the game service, HackShield stopping function (_AhnHS_StopService, _AhnHS_Uninitialize) was not called and the game program was abnormally terminated. Can I start HackShield again by starting the game program again?

If the game program is terminated without calling the HackShield stopping function, the hack prevention driver will not be unloaded in the system. If the game program is restarted by the HackShield initialization function, the existing driver will be forcibly unloaded and the new driver will be loaded for normal initialization.

When a new hacking tool is detected, how can I block it?

In case a new hacking tool is detected, following information will be sent to AhnLab. AhnLab analyzes hacking tool date and reflect the analysis result to the weekly engine updates or emergency engine.

Games with hacking tools running

OS version

Simple description of the hacking tool

Hacking tool execution file

9.2. Index

_AhnHS_CheckHackShieldRunningStatus	60, 62, 63, 65
_AhnHS_HSUpdate	82, 87
_AhnHS_MakeGuidAckMsg.....	120
_AhnHS_PauseService	56
_AhnHS_ResumeService	58
_AhnHsUserUtil_CreateUser	161, 164
_AhnHsUserUtil_SetFolderPermission	142, 162, 163, 167, 168, 169, 170
_AntiCpSvr_Initialize	100, 129
_AntiCpSvr_AnalyzeAckMsg.....	109, 111, 114
_AntiCpSvr_AnalyzeGuidAckMsg	105
_AntiCpSvr_Finalize	102, 131
_AntiCpSvr_MakeGuidReqMsg	103
_AntiCpSvr_MakeReqMsg	106
_HsCrypt_FRead	150, 155
_HsCrypt_GetDecMsg	150, 154
_HsCrypt_GetEncMsg	149, 153
_HsCrypt_InitCrypt	149, 151

A

```

ACTAPCPARAM_DETECT_AUTOMOU
SE ..... 44
ACTAPCPARAM_DETECT_HOOKFUNC
TION ..... 43
AHNHS
  _SPEEDHACK_SENSING_RATIO_HI
  GH ..... 37
AHNHS
  _SPEEDHACK_SENSING_RATIO_HI
  GHEST ..... 37
AHNHS
  _SPEEDHACK_SENSING_RATIO_LO
  W ..... 38
AHNHS
  _SPEEDHACK_SENSING_RATIO_LO
  WEST ..... 38
AHNHS
  _SPEEDHACK_SENSING_RATIO_NO
  RMAL ..... 38
AhnHS_StopService ..... 52
AhnHS_Uninitialize ..... 54
AHNHS_ACTAPC_DETECT_ALREADY
HOOKED ..... 43
AHNHS_ACTAPC_DETECT_AUTOMAC
RO ..... 44
AHNHS_ACTAPC_DETECT_AUTOMOU
SE ..... 43
AHNHS_ACTAPC_DETECT_DRIVERFA

```

ILED	45
AHNHS_ACTAPC_DETECT_HOOKFUNCTION.....	43
AHNHS_ACTAPC_DETECT_KDTRACE	45, 46
AHNHS_ACTAPC_DETECT_KDTRACE_CHANGED	46, 47
AHNHS_ACTAPC_DETECT_PROTECT_SCREENFAILED	47
AHNHS_ACTAPC_DETECT_SPEEDHACK.....	45
AHNHS_ACTAPC_STATUS_HACKSHIELD_RUNNING	48
AHNHS_ALLOW_CSRSS_OPENPROCESS.....	32
AHNHS_ALLOW_LSASS_OPENPROCESS	32
AHNHS_ALLOW_SVCHOST_OPENPROCESS	32
AHNHS_ALLOW_SWITCH_WINDOW	33
AHNHS_CHKOPT_ALL	22, 31
AHNHS_CHKOPT_ANTIFREESERVER	32
AHNHS_CHKOPT_AUTOMOUSE	31
AHNHS_CHKOPT_DETECT_VIRTUAL_MACHINE	35, 36
AHNHS_CHKOPT_KDTARCEER.....	31
AHNHS_CHKOPT_LOCAL_MEMORY_PROTECTION.....	31
AHNHS_CHKOPT_OPENPROCESS ..	31
AHNHS_CHKOPT_PROCESSSCAN ..	31
AHNHS_CHKOPT_PROTECT_D3DX ..	33
AHNHS_CHKOPT_PROTECTSCREEN	33
AHNHS_CHKOPT_READWRITEPROCESSMEMORY	31
AHNHS_CHKOPT_SELF_DESTRUCTION	34
AHNHS_CHKOPT_SPEEDHACK	31
AHNHS_CHKOPT_STANDALONE	33
AHNHS_DISPLAY_HACKSHIELD_LOGO	33
AHNHS_DISPLAY_HACKSHIELD_TRAY_ICON	35
AHNHS_DONOT_TERMINATE_PROCESS	32
AHNHS_ENGINE_DETECT_GAME_HACK.....	42
AhnHS_Initialize	30
AhnHS_StartService	48
AHNHS_USE_LOG_FILE	32
AntiCPSvr.dll	93, 125, 138
AntiCpSvrTool.exe	93

E

EagleX9x.vxd, EagleXNT.sys or EagleX64.sys.....	15
EhSvc.dll	159
EHSvc.dll	14
ERROR_ANTICPXSVR_CLIENT_FILE_ATTACK (0xE904000B)	116
ERROR_ANTICPXSVR_DETECT_CALL_BACK_IS_NOTIFIED.....	118
ERROR_ANTICPXSVR_HSHIELD_FILE_ATTACK (0xE904000A)	116
ERROR_ANTICPXSVR_INVALID_PARAMETER (0xE9040003)	107, 114
ERROR_ANTICPXSVR_MEMORY_ATTACK (0xE904000C)	116
ERROR_ANTICPXSVR_NOT_YET_RECEIVED_RESPONSE (0xE9040005).....	107, 115
ERROR_ANTICPXSVR_UNKNOWN_CLIENT (0xE904000E)	116
ERROR_HSCRYPTLIB_EXCEPTION.....	157
ERROR_HSCRYPTLIB_FREAD_DECRYPT_FREAD	156
ERROR_HSCRYPTLIB_FREAD_DECRYPT_PT_GETDECMMSG	157
ERROR_HSCRYPTLIB_FREAD_DECRYPT_PT_RANGE	156
ERROR_HSCRYPTLIB_FREAD_INVALIDPARAM	155
ERROR_HSCRYPTLIB_FREAD_SIZEERROR	156
ERROR_HSCRYPTLIB_GETDECMMSG_INVALIDPARAM.....	154
ERROR_HSCRYPTLIB_GETENCMMSG_INVALIDPARAM.....	153
ERROR_SUCCESS... ..	129, 153, 154, 155
Extended Server-side Detection (AntiCpX)	91

H

HS_ERR_ALREADY_GAME_STARTED	51
HS_ERR_ALREADY_INITIALIZED	40
HS_ERR_COMPATIBILITY_MODE_RUNNING	39
HS_ERR_DEBUGGER_DETECT.....	40
HS_ERR_DRV_FILE_CREATE_FAILED	38, 50, 51, 157, 165, 166
HS_ERR_HSMS_NOT_RUNNING....	136
HS_ERR_HSMS_WAIT_TIME_OUT	135
HS_ERR_INIT_DRV_FAILED.....	39

HS_ERR_INVALID_FILES.....	39, 61, 65, 66, 67, 68, 69, 135
HS_ERR_INVALID_LICENSE	39
HS_ERR_INVALID_PARAM.....	57, 135
HS_ERR_NEED_ADMIN_RIGHTS	40
HS_ERR_NOT_INITIALIZED.....	52, 60
HS_ERR_NOT_INITIALIZED.....	49
HS_ERR_NOT_INITIALIZED.....	54
HS_ERR_NOT_INITIALIZED.....	56
HS_ERR_NOT_INITIALIZED.....	58
HS_ERR_OK.....	38
HS_ERR_REG_DRV_FILE_FAILED	50
HS_ERR_START_DRV_FAILED.....	51
HS_ERR_START_ENGINE_FAILED	49
HS_ERR_UNKNOWN.....	41, 135
HS_ERR_VIRTUAL_MACHINE_DETECT.....	51
HsCryptLib.....	148
HsCryptLib.lib	147
HsCryptoUtil program.....	147
HSERROR_ENVFILE_NOTREAD.....	84, 88
HSERROR_ENVFILE_NOTWRITE.....	85, 88
HSERROR_LIB_NOTEDIT_REG	89
HSERROR_NETWORK_CONNECT_FAIL.....	89
HSERROR_NOTFINDFILE	85, 89
HShield.lib	93, 125, 139
HUpdate.exe	14
HsUserUtil Data.....	159
HsUserUtil.lib.....	159
HSUSERUTIL_ERR_ADDSHADOWACNT_FAIL	166
HSUSERUTIL_ERR_DELHIDEIDINFO_FAIL.....	165
HSUSERUTIL_ERR_DELSHADOWACNT_FAIL	165
HSUSERUTIL_ERR_DELSHADOWACNTINFO_FAIL	165
HSUSERUTIL_ERR_NOT_ADMIN....	165
HSUSERUTIL_ERR_NOT_NT.....	165, 169, 170, 171
HSUSERUTIL_ERR_OK.....	164, 167, 169, 170, 171
HSUSERUTIL_ERR_SETFLDRPERMISSION_FAIL	168, 169, 170, 171

P

PFN_AhnHS_Callback.....	42
-------------------------	----

V

V3Pro32s.dll	15
--------------------	----

9.3. Revisions

Date	Revisions
2008-02-26	- Changed AhnLab_HackShield_Programming_Guide.doc into AhnLab_HackShield_2.0_Programming_Guide.doc.
2008-11-26	- [BT50597] Added AHNHS_ENGINE_DETECT_WINDOWED_HACK callback
2009-01-07	- 2008 -> 2009 - Added AntiFreeServer feature (with tool description)
2009-10-23	- Added the contents of the server-side detection (64Bit).
2010-10-27	- Deleted content on interoperability with old server