



nProtect GameGuard Manual V2.0

-게임가드 적용 매뉴얼-

■ (주)잉카인터넷 게임보안 사업본부 게임가드개발팀 개발파트

DV-110422

Ver. 1.0

업체 제공용

차 례

1장. 게임가드 소개

- 1. 게임가드 소개3
- 2. 기존 게임 보안 프로그램의 문제점3
- 3. 게임가드의 특징4

2장. 게임가드 구성

- 1. 게임가드 시스템 구성5
- 2. 게임가드 실행 프로세스7

3장. 게임가드 설치

- 1. 게임가드 업데이트 서버 설치8
- 2. 게임가드 로그 서버 설치 (별도 서버 구축 필요)11
- 3. 게임가드 클라이언트 모듈 설치11

4장. 게임가드 적용

- 1. 게임 클라이언트에 게임가드 적용 방법12
- 2. 게임 실행 중 게임가드 실행 여부 확인 방법15
- 3. 사용자의 불법 프로그램 사용 여부 확인15
- 4. 게임가드 메시지의 종류 및 처리 방법 16
- 5. 게임가드 라이브러리 함수 설명16
- 6. 적용 예18
- 7. 패킷 암호화 함수24
- 8. VirtualTable 후킹검사 함수30

5장. 게임가드 서비스

- 9. 게임가드 FAQ31
- 10. 주요 에러 코드32
- 11. 고객 지원 33

1장 게임가드 소개

1. 게임가드 소개

최근 수년 간 온라인 게임은 많은 사용자들의 엔터테인먼트 수단으로 대중화 되어 왔습니다. 그러나 일부 악의적인 의도를 가진 사용자들이 정당하지 않은 방법 즉 게임핵을 이용한 반칙성 플레이나 해킹을 통해 타인의 계정을 도용하는 등, 게임의 질서나 규칙이 상당히 훼손되고 있는 실정입니다.

㈜잉카인터넷은 온라인 게임에서의 해킹툴 및 악성 코드에 대한 진단 및 차단 서비스를 다년간 제공하여 왔으며 고객의 여러 요구사항을 수용하고 게임 보안 기술의 KNOW-HOW를 바탕으로 새로운 개념의 해킹 차단 방식이 적용 된 nProtect GameGuard를 개발하게 되었습니다.

nProtect GameGuard는 클라이언트 측에서 사용자의 게임핵 사용이나 게임 해킹 시도로부터 게임 클라이언트를 방어해주는 게임 보안 프로그램입니다. 또한 계정 정보나 개인 정보를 유출시킬 수 있는 악성 코드를 스캔/차단하므로 선량한 사용자들을 보호해줍니다.

2. 기존 게임 보안 프로그램의 문제점

- 기존의 게임 보안 프로그램은 게임핵 프로그램 파일의 패턴값(Digest value)을 기반으로 검출하는 방식을 사용하여 왔습니다. 따라서 새 버전의 게임핵 프로그램이 나올 때 마다 패턴 추가와 모듈 업데이트가 불가피하였습니다.
- 패턴에 추가된 게임핵이라 하더라도 실행 파일 압축 프로그램(UPX 등)으로 게임핵 프로그램 파일을 변형시키면 패턴값이 달라져 더 이상 검출되지 않습니다.
- 일단 패턴 검사를 피하게 되면 ArtMoney, CheatEngine, TSearch, GameMaster, GameWizard와 같은 게임 프로세스의 메모리를 스캔하여 능력치나 장비 등의 특정 정보를 읽어 그 주소 번지를 알려주고 수정할 수 있는 메모리핵에 대한 보호가 전혀 되지 않습니다.
- 게임 보안 프로그램이 주기적으로 패턴 비교를 위해 프로세스 검사와 파일 입출력을 수행하므로 CPU 시간을 점유하여 게임의 진행에 다소 영향을 주기도 합니다.
- 검사 주기 - 보통 5초~20초 - 안에 핵 프로그램이 실행되어 사용되는 것을 막지 못하며, 순간적으로 실행되어 게임 프로세스에 붙은 뒤 종료해버리는 형태의 핵 프로그램에 대해서는 전혀 대응하지 못 합니다.
- 보안 프로그램 자체에 보안 홀이 존재하여 보안 프로그램의 프로세스 강제 종료, 보안 모듈의 버전 조작, DLL 바꾸기, 게임과의 메시지 송수신 차단 등 보안 프로그램 자체를 해킹하여 보안 기능 작동을 무력화시키는 경우도 있습니다.
- 최근에는 보안성에만 치중하여 안정성이 떨어지는 보안 프로그램들도 있습니다. 정상적인 프로그램들의 기능이 작동하지 않게 되어 사용자에게 불편을 주거나, 백신 프로그램이나 게임방 프로그램 등과 충돌하여 시스템이 불안정해지거나 다운되는 문제가 발생하기도 합니다.

3. 게임가드의 특징

- 악성코드 및 스파이웨어 진단 및 차단

안티 바이러스 엔진(백신)을 내장하고 있어 BackOrifice, Netbus, SubSeven의 Backdoor/Trojan 등의 악성 코드와 스파이웨어를 진단 및 차단하여 게임 사용자에게 안전한 게이밍 환경을 제공합니다.

- 게임 프로세스에 대한 접근 차단

실시간으로 게임에 대한 모든 접근을 감시하여 불법적이거나 의도적으로 조작하려는 접근을 차단하므로 다양한 종류의 게임핵, 메모리 에디터 등을 근본적으로 차단할 수 있습니다.

또한 강력한 안티 디버깅 기술을 사용하여 디버거로 게임 프로세스를 조작하는 것을 차단합니다.

- 스피드핵 진단

시스템의 타이머들을 실시간으로 감시하여 타이머를 조작하는 모든 종류의 스피드핵을 검출해 낼 수 있습니다.

- 오토마우스 및 매크로 프로그램 차단

게임 클라이언트로 임의의 마우스나 키보드 입력을 보내는 프로그램에 대해서 원천적인 차단 기능을 가지고 있어 대다수의 범용 오토 마우스나 매크로 프로그램들은 기본적으로 차단됩니다.

최근에 나오고 있는 키보드/마우스 필터 드라이버나 PORT I/O 커널 드라이버를 사용하는 매크로 프로그램도 게임가드의 드라이버 스캔 및 제어 기능을 통해 대부분 차단이 가능합니다.

- 메모리 패턴 스캔

파일 패턴 스캔의 한계를 극복하기 위해 게임가드는 실행된 후에 메모리에 올라간 데이터를 기반으로 진정한 의미의 패턴(해킹툴의 주요 특징 코드) 스캔을 하여, 실행 파일 압축 프로그램을 통한 패턴 회피는 물론 새 버전의 해킹툴까지도 검출해낼 수 있는 강력한 해킹툴 스캔 엔진을 내장하고 있습니다.

- 강력하면서도 안정적인 동작

수년 간 16개국 70여개 게임에 서비스 해오면서 축적된 노하우를 바탕으로 강력한 보안 기능을 가지면서도 다양한 게임, 다양한 국가의 PC 환경에서도 안정적으로 동작하도록 개발하였습니다.

- 보안 모듈 자체 보안

게임가드의 모든 모듈은 고유의 인증 방식을 사용하여 신뢰성 여부와 변조 여부를 확인하므로 모듈 조작으로 인한 취약점은 거의 없습니다.

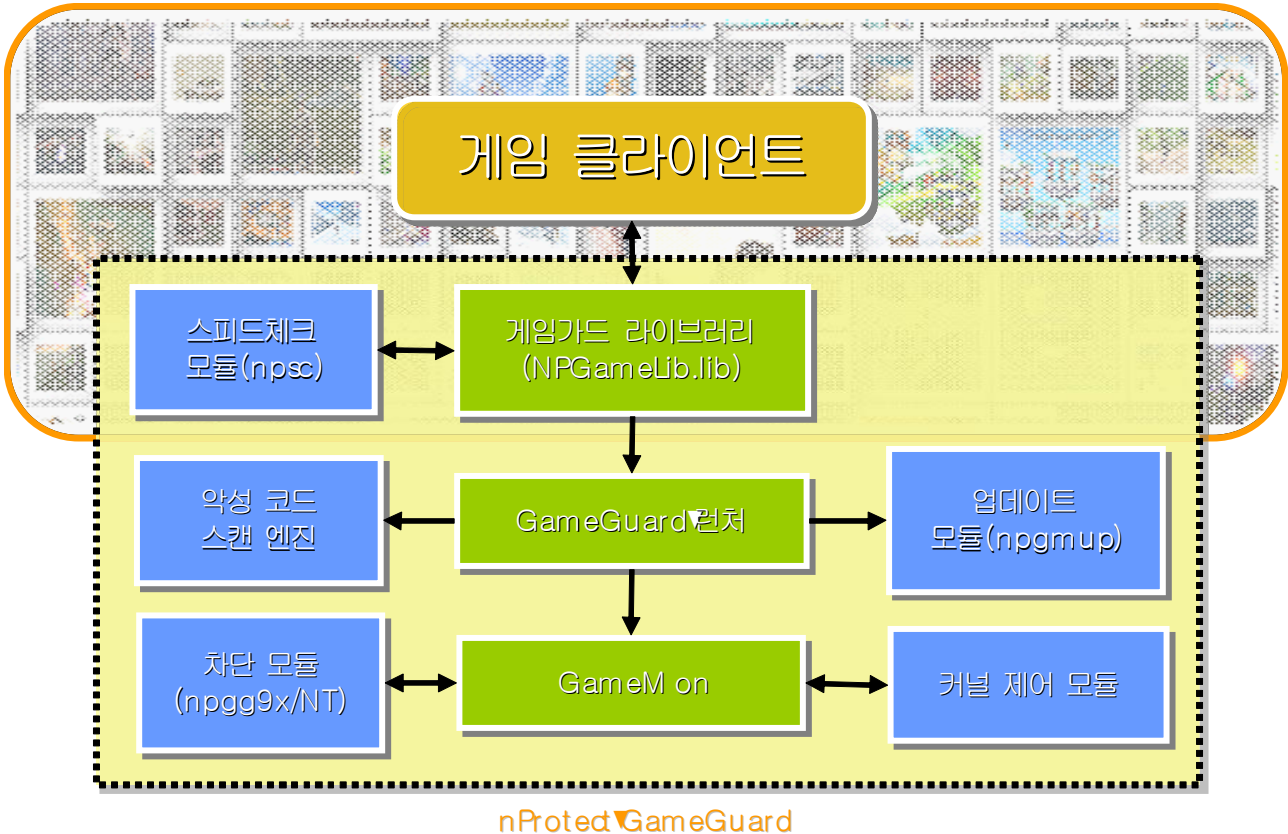
강력한 안티 디버깅 기술을 사용하여 게임가드가 디버깅 당하거나 조작되어 무력화되는 경우도 거의 없습니다.

- CPU 점유율 최소화

게임가드의 메모리 패턴 스캔 방식은 주기적인 검사가 아니라 새로운 프로세스의 생성을 감지하여 그 프로세스만 검사하는 방식이므로 게임 중 다른 프로그램을 실행하지 않는 경우에는 CPU 점유가 1% 이내입니다. 따라서 게임 실행 퍼포먼스에 거의 지장을 주지 않습니다.

2장 게임가드 구성

1. 게임가드 시스템 구성



1) nProtect 게임라이브러리 (NPGGameLib.lib)

- 게임 클라이언트에 같이 링크될 정적 라이브러리입니다.
- 간단한 함수 호출로 게임가드 최신 모듈 업데이트, GameMon 실행, 스피드체크 모듈 로드, GameMon과의 통신 기능을 해줍니다.
- 이 라이브러리에서 GameMon 초기화 실패, 스피드해킹 감지, 게임해킹 감지, 게임 변조, GameMon 종료 등의 메시지를 Callback 함수를 통해 알려줍니다.

2) 스피드체크 모듈 (npssc.des, nppt9x.vxd, npptNT2.sys)

- 시스템의 타이머를 감시하여 스피드해킹의 사용 여부를 감지하는 모듈입니다.
- 시스템 포트를 제어하므로 9x와 NT에 각각 다른 커널 모드 드라이버를 사용합니다.
- 게임 프로세스에서 동작해야만 가장 확실한 검사가 가능하므로 게임에서 로드합니다

3) 게임가드 런처 (GameGuard.des)

- 악성 코드 검사 및 게임가드 최신 업데이트를 수행하는 모듈입니다.
- 내부적으로 업데이트 모듈(npgmup.des)를 사용하여 업데이트 작업을 수행합니다.

4) 업데이트 모듈 (npgmup.des)

- 게임가드 업데이트 서버에 접속하여 최신 파일을 업데이트 받는 모듈입니다.
- 파일의 Sign, Hash 검사를 통해 변조되었거나 조작된 파일은 정상적인 모듈로 업데이트 합니다.

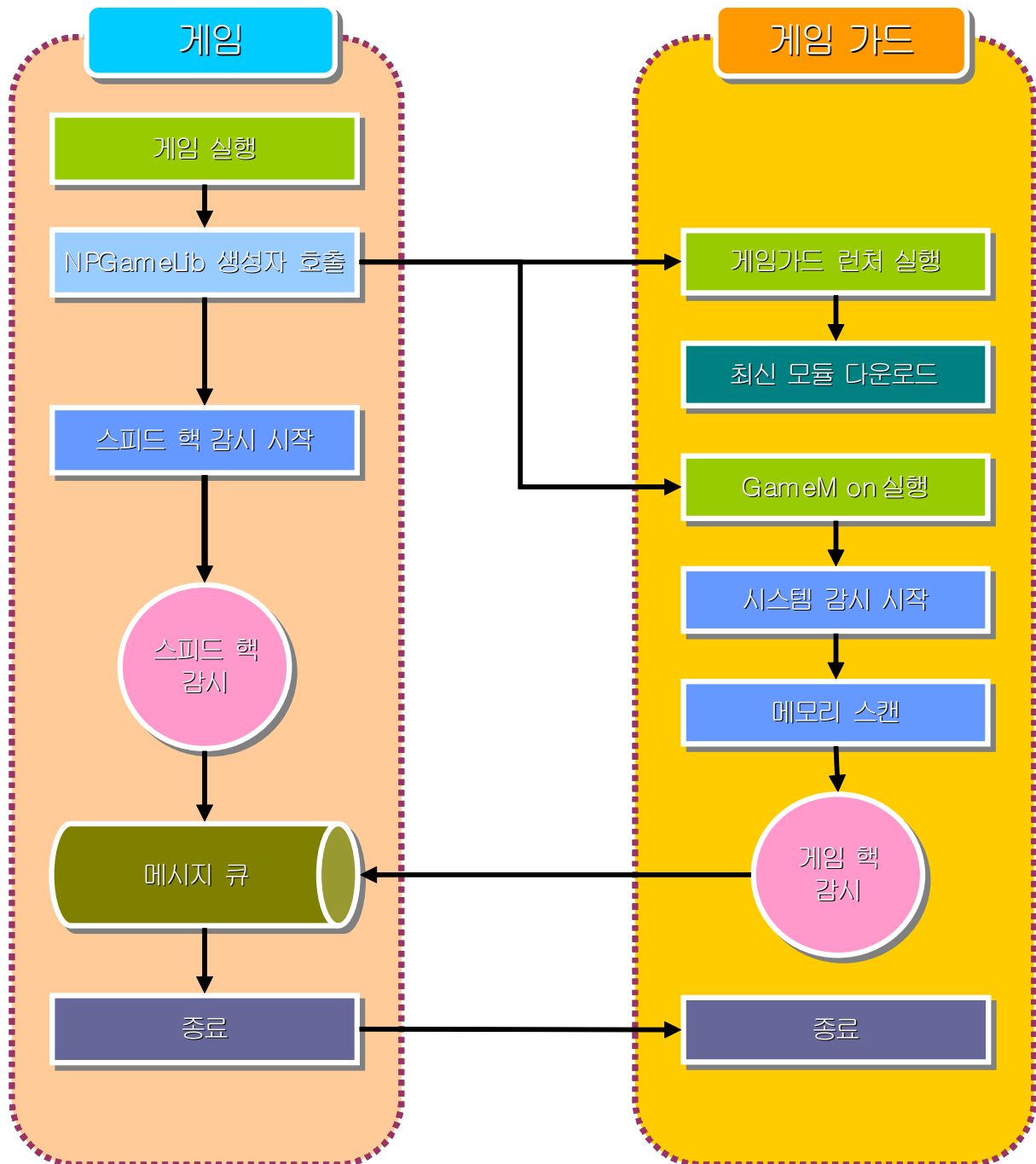
5) GameMon (GameMon.des)

- 게임과 함께 실행되어 시스템을 모니터링 하면서 게임 프로세스를 보호하는 핵심 모듈입니다.
- 실행 중인 프로세스들을 스캔하여 불법 프로그램이 발견되면 게임으로 메시지를 보냅니다.
- 내장된 커널 모드 드라이버를 로딩시켜 커널 레벨의 기능들을 수행합니다.
- 강력한 디버깅 방지 코드 및 변조 검사 코드를 내장하고 있어 디버깅이나 변조가 매우 힘듭니다.

6) 게임가드 차단 모듈 (npgg9x.des, npggNT.des)

- GameMon이 사용하는 모듈로서 게임 프로세스로의 접근을 차단해주는 모듈입니다.
- 게임을 대상으로 한 프로세스 오픈, 메모리 스캔, 메모리값 조작, 디버깅 등을 막아줍니다.
- 게임을 대상으로 한 키보드 및 마우스 이벤트를 차단합니다.
- OS 구조 차이로 윈도우 95/98/ME용과 윈도우 NT/2K/XP/2K3/LH용이 분리되어 있습니다.

2. 게임가드 실행 프로세스



3장 게임가드 설치



1. 게임가드 업데이트 서버 설치

1) nProtect GameGuard 구동에 필요한 구성 요소

nProtect GameGuard 는 웹 환경에서 실행되도록 설계된 웹보안 서비스로 웹서버에 설치되어 사용자에게 다운로드 되는 서비스로서 서비스 전 몇 가지 준비해야 할 사항이 있습니다.

가. 서버 : 서버는 웹서비스 환경의 구축이 가능한 어떠한 서버 및 운영체제라도 가능하나 nProtect GameGuard 서비스 시 부하를 고려하여 충분한 용량의 하드디스크와 메모리를 확보해야 합니다. 별도의 nProtect 서버구축이 어려우시다면 기존의 게임 패치서버에 nProtect 폴더를 생성하여 웹서비스 되어도 무방합니다.

기본사양	권장사양
CPU : Pentium III 800 Dual 이상	CPU : Pentium 1G Dual 이상
RAM : 512 MB 이상	RAM : 1G 이상
HDD : 여유공간 20MB	HDD : 여유공간 20MB이상

- 서버는 웹 서비스와 ftp 서비스가 가능해야 합니다.
(게임가드 업데이트방식이 http, ftp 이기에 익명연결 및 읽기권한으로 웹서비스가 가능해야 합니다.)
- (리눅스일 경우에는 아파치, 윈도우서버일 경우에는 IIS 설치를 권장합니다.)
- 웹서비스가 이루어 지는 폴더에 ftp로 접근이 가능해야 합니다.
(INCA Internet 에서 정기 업데이트 및 긴급 패치시 이용하니 ftp 로그인 및 쓰기권한이 있어야 합니다. .)
- Windows Server 2003 에서의 IIS는 아래의 사항이 추가되어야 정상적인 웹서비스가 가능합니다.
인터넷 정보 서비스(IIS)관리 -> 웹 사이트 등록 정보 -> HTTP 헤더 탭의 MIME 형식-> 새 형식에서 .*을 추가합니다. (콘텐츠 형식(MIME) : unknown 으로 설정하셔도 무방합니다.)

서버구성후 서버OS, 서버대수, 네트워크 대역폭, 서버 도메인 또는 IP, FTP 로그인 계정의 정보를 ㈜잉카인터넷 담당자에게 제공해야 합니다.

서비스를 위해 충분한 네트워크 대역폭을 고려해야 하며 게임가드 업데이트시에는 서버사양 보다는 네트워크 대역폭에 의존하므로 한 서버에 NIC 추가장착 방식도 무방합니다.

2대 이상의 서버일 경우에는 LB(Load Balancing) 구현 또는 도메인으로 서비스되어도 무관하며 그중 메인서버에 ftp로 접근 및 파일업로드후 웹서버 동기화가 가능해야 합니다.

(LB구현이 어려우시면 게임가드에서 자체적으로 각 서버를 랜덤으로 쿼리하는 방식으로 대체합니다.)

나. nProtect GameGuard 모듈 적재

* 모듈 List

- 게임가드 모듈

GameMon.npz	499KB
npgg9x.npz	40KB
npggNT.npz	51KB
npgmup.npz	71KB
nppt9x.npz	1KB
npptNT.npz	2KB
npsec.npz	60KB
GameName.npz	1KB
Splash.npz	31KB
update.cfg	2KB

- 백신 모듈

TeCtrl.dll.npz	12KB
TyAv32.dll.npz	25KB
tyavd.npz	260KB
tyavn.npz	217KB
TeAs.dll.npz	43KB
teasbase.npz	616KB
teasname.npz	39KB
category.tsf	1KB
teas.tls	1KB
daily.tsf	7KB
=> TOTAL : 1.979 MB	

2) nProtect GameGuard 서비스를 위한 기본적인 서버 구성

nProtect GameGuard 서비스를 위해서는 서버에 기본적인 웹서비스 환경이 구축되어 있어야 합니다. 이러한 환경이 구축된 서버라고 하면 nProtect 모듈 및 엔진을 정의된 규칙에 따라 업로드 시에는 즉시 서비스가 가능할 것입니다.

그렇지 못할 경우에는 나열되는 서비스 환경을 구축하고 서비스 이상 유무를 Test 한 후 문제 사항을 수정해야 합니다.

1) web service & ftp service (익명연결)

: 서버에 설치된 운영체제에 따라 각각의 웹 서버를 서버에 설치한 후 웹서버의 정상 동작을 확인합니다.

- 아래의 경로에서 정상적으로 해당 파일이 다운로드 받아져야 합니다.
- nProtect 폴더로부터 웹서비스 환경이 되어야 합니다.
-

Ex) `http:// nProtect.domain/nProtect/GameGuard/RealServer/update.cfg`

`http://IP/nProtect/GameGuard/RealServer/update.cfg`

`ftp:// nProtect.domain/nProtect/GameGuard/RealServer/update.cfg`

`ftp://IP/nProtect/GameGuard/RealServer/update.cfg`

2) ftp service

: 외부에서 엔진 업데이트를 위해 ftp service 를 준비해야 합니다.

서비스 구축에 문제가 있을 경우 담당자의 Email 로 대체합니다.

3) 중국의 네트워크 로컬라이징 관련

중국은 네트워크가 크게 두 종류(CT, CNC)로 구분되어 있어 CT의 네트워크에 속한 망은 CNC로의 접속이 원활하지 않으며 CNC와 CT로의 접속도 동일한 환경입니다. 이에 CT와 CNC에 각각 nProtect 서버를 구축후 아래와 같이 네트워크 로컬라이징 관련 작업이 필요합니다.

1. 게임런처에서 ServerIndex를 사용하여 레지스트리에 정보기입

1) 아래 소스를 참고하여 index 1은 CT, index 2는 CNC를 기입해 주시면 됩니다.

```
BOOL SetServerIndex(DWORD dwIndex)
```

```
{
```

```
    BOOL ret = FALSE;
```

```
    HKEY hReg;
```

```
    DWORD result;
```

```
    DWORD datasize = sizeof(DWORD);
```

```
    If (RegCreateKeyEx(HKEY_CURRENT_USER,_T
```

```
        ("SOFTWARE\\WINCA\\Internet\\WWnProtectGameGuard\\WWUpdate"),
```

```
        0, 0, 0, KEY_ALL_ACCESS, NULL, &hReg, &result) == ERROR_SUCCESS)
```

```
    {
```

```
        if (result == REG_OPENED_EXISTING_KEY)
```

```
        {
```

```
if (RegSetValueEx(hReg, _T("ServerIndex"), 0, REG_DWORD, (LPBYTE) &dwIndex, datasize))
{
    ret = TRUE;
}
}
RegCloseKey(hReg);
}

return ret;
}
```

작업이 완료된 후에는 게임가드 실행시 해당 레지스트리 정보를 참조하여 CT와CNC의 서버를 판단하여 업데이트를 시도하게 됩니다. 만일 각 망에 두대이상의 서버 구축 필요시에는 각 망별 도메인의 작업을 해주시면 보다 효율적인 업데이트가 가능합니다.

2. 게임가드 로그 서버 설치 (별도 서버 구축 필요)

- ① 유저 피씨에서 핵툴 사용시 게임가드가 게임쪽으로 핵툴 발견 메시지를 던지면 클라이언트에서 그 사용자 정보와 메시지를 서버쪽에 알려서 서버에서 해당 정보의 로그취합이 가능합니다.
- ② 상기 1번 사항의 설정에 무리가 있으시다면 아래와 같이 게임가드 서버 이외에 별도의 핵툴 로그수집용 웹서버의 구축이 필요합니다.

<요구사항>

- I. IIS 웹서버 + PHP : PHP 로 만들어진 관리자용 페이지에 사용
- II. MySQL : 핵툴 로그를 저장시킬 DataBase
- III. OS : Windows Server
- IV. NpggData_Server.exe : 게임가드로부터 핵툴 로그를 전송받음. (INCA 제공)
- V. NpggData_sql.exe : 핵툴 로그를 MySQL DataBase 에 저장. (INCA 제공)
- VI. 하드디스크 최소 2GB의 공간 필요(100만건의 데이터가 풀로 쌓일 경우)
- VII. 서버 구축 완료후 터미널 계정을 잉카인터넷 담당자에게 제공하면 상기 설정을 진행하겠습니다.

3. 게임가드 클라이언트 모듈 설치

게임 클라이언트가 설치된 폴더에 다음 게임가드 파일들을 위치시킵니다.

GameName.ini ; 게임가드 설정 파일

GameGuard.des ; 게임가드 런처(업데이트) 프로그램

두 파일 모두, 존재유무만 확인하여 존재 하지 않을 경우에만 유저 클라이언트에 덮어쓰기 부탁드립니다.

추후, 게임가드 모듈에서는 게임가드 런처 파일(GameGuard.des, 게임명.ini)의 자체적인 업데이트가 가능합니다.

(게임이 종료된 후에 게임가드 폴더안의 GameGuard.des 파일은

게임폴더내의 GameGuard.des 파일과 비교하여 최신버전을 게임폴더로 복사합니다.)

4장 게임가드 적용

1. 게임 클라이언트에 게임가드 적용 방법

■ 게임가드 라이브러리와 헤더 파일 복사

NPGameLib.lib 파일을 게임 클라이언트 프로젝트의 라이브러리 폴더로 복사합니다.

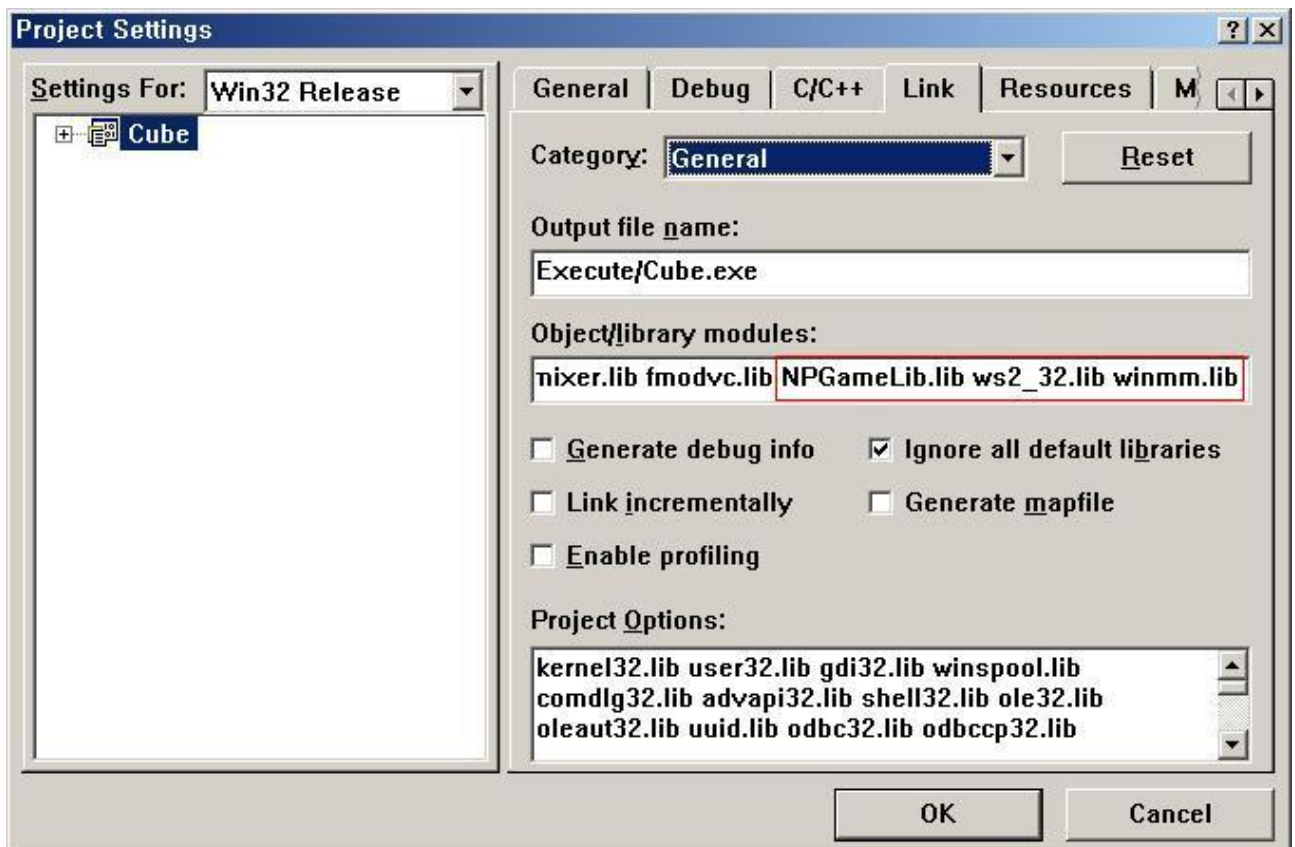
NPGameLib.h 파일을 게임 클라이언트 프로젝트의 헤더 파일 폴더로 복사합니다.

■ 게임가드 라이브러리를 Link 리스트에 추가

제공되는 NPGameLib.lib 파일을 게임 클라이언트 프로젝트 세팅의 Link 리스트에 추가시킵니다.

NPGameLib.lib에서 ws2_32.lib와 winmm.lib를 참조하므로 이 lib들도 함께 Link시킵니다.

☞ 메인메뉴 [Project] - [Settings] - [Link] 탭

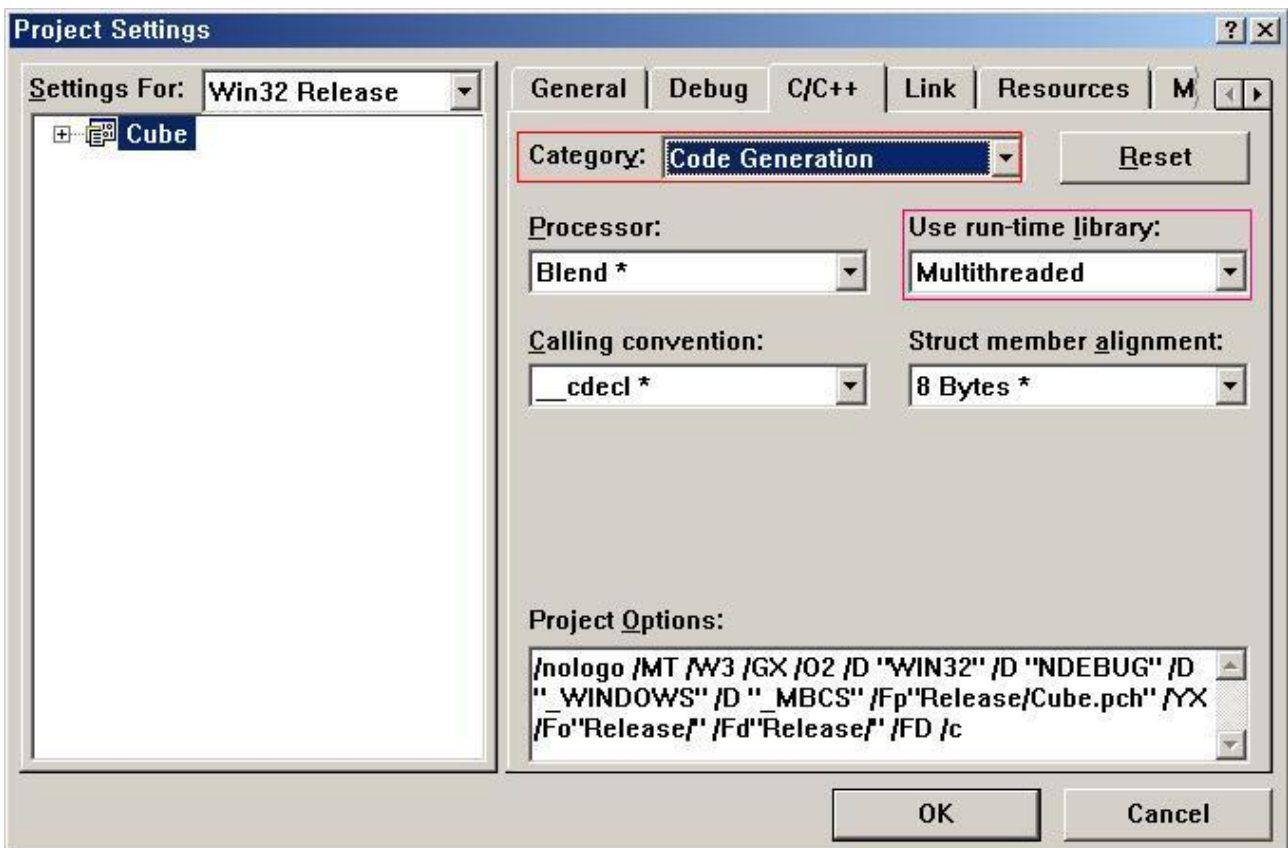


< Visual C++ 6.0의 링크탭 >

■ C Runtime Library 변경

NPGameLib.lib는 Multithreaded C Runtime Library로 컴파일 되었으므로, 게임 클라이언트의 프로젝트 세팅에서 Code Generation 항목에서 이를 변경합니다.

☞ 메인메뉴 [Project] - [Settings] - [C/C++] 탭 - [Code generation] Category



< Visual C++ 6.0의 C/C++ 탭 >

* Multithreaded DLL 대신 Multithreaded를 사용하는 이유

-> Multithreaded DLL인 경우, 실행시 msvcrtd.dll이 로드 되어 msvcrtd.dll에서 C Runtime Library 함수들이 실행되게 됩니다. 따라서 디버거로 strcpy()나 memcpy() 같은 함수에 쉽게 Breakpoint를 걸 수 있게 됩니다. 그러므로 보안성 향상을 위해 실행 파일 내부에 static 형태로 link되는 Multithreaded를 사용하는 것이 좋습니다.

만약 게임 클라이언트가 사용하는 특정 라이브러리가 Multithreaded가 아닌 다른 run-time library로 되어 있어 계속 링크 에러가 난다면 게임보안센터(GameService@inca.co.kr)로 문의 바랍니다.

※ 제공 받은 NPGameLib.lib가 MTDLL용이라면 Multithreaded DLL로 설정해야 합니다.


■ 게임가드 헤더 파일 인클루드

게임가드를 적용할 소스 파일에 NPGameLib.h를 인클루드 해줍니다.

 #include "NPGameLib.h"

■ 게임가드 클래스 객체 생성

전역 변수로 CNPGameLib 클래스의 객체를 하나 생성합니다.

 CNPGameLib npgl("게임이름"); (ex) ini파일명과 일치

혹은 CNPGameLib 클래스의 포인터만 전역으로 선언하신 후 나중에 동적으로 생성하셔도 됩니다.

```
# CNPGameLib *pNpgl = NULL;  
pNpgl = new CNPGameLib("Cube");
```

이때 생성자의 인자로 게임 이름의 문자열을 넣어줍니다. 위 예시의 게임 이름은 Cube 입니다.
이 문자열은 설정 파일인 .ini 파일의 확장자를 제외한 파일이름 부분입니다. 예) Cube.ini
CNPGameLib 객체가 생성될 때 게임가드 초기화가 시작됩니다.

■ 게임가드 초기화 결과 처리

CNPGameLib 객체가 생성된 직후 npgl.Init() 함수를 호출해 게임가드 초기화 결과를 처리합니다.

```
# DWORD dwResult = npgl.Init();  
if (dwResult != NPGAMEMON_SUCCESS)  
{  
    ...    // 각 에러코드 처리법은 아래 예제 참조  
}
```

동적으로 new를 통해 객체를 생성한 경우에는 바로 다음 라인에서 Init()을 처리해주시면 됩니다.
전역에서 객체를 생성한 경우에는 WinMain()과 같은 프로그램 실행 시작 부분에서 Init()을 처리해 주시면 됩니다.

반드시 Init()의 리턴값을 검사해서 리턴값에 따라 적절한 에러 메시지를 사용자에게 보여주고 게임을 종료합니다. (뒤에 나오는 예제 및 에러코드 참조)

■ 게임 클라이언트의 메인 윈도우 핸들 전달

게임 클라이언트의 모든 그래픽, UI 관련 초기화가 끝난 뒤, 메인 윈도우 핸들(HWND)을 게임가드로 전달합니다.

```
# npgl.SetHwnd(g_hMainWnd);
```

게임가드는 윈도우 핸들을 전달받는 후부터 콜백 함수로 메시지를 보냅니다.

■ 게임가드 콜백 함수 작성

콜백 함수는 게임가드가 보내는 메시지를 처리하도록 직접 작성하시는 함수입니다.

```
# BOOL CALLBACK NPGameMonCallback(DWORD dwMsg, DWORD dwArg)  
{  
    // 자세한 구현 방법은 아래 예제 참조  
}
```

콜백 함수로는 게임핵 발견, 스피드핵 발견, 게임 변조, 게임가드 변조 등의 메시지가 전달됩니다.

이 때 먼저 게임 종료 처리를 하고 나서, 메시지 박스는 게임 종료 직전에 출력해줍니다.

메시지 박스를 먼저 출력하지 않는 이유는 해커가 그 위치를 힌트로 디버깅 분석을 용이하게 할 수 있게 되며, 콜백 함수를 호출하는 쓰레드는 게임의 메인 쓰레드와 다르므로 메시지 박스만 띄워둔 채 게임을 계속 진행하게 만들 수도 있기 때문입니다.

따라서 콜백 함수에서는 나중에 출력할 메시지 번호나 메시지 문자열만 저장해두고, 게임이 종료되기 직전에 출력 내용이 있는지 확인해서 있으면 저장해 두었던 메시지를 보여주시면 됩니다.

콜백 함수의 메시지에 따라 게임을 계속 진행할 경우에는 return TRUE를,
게임을 종료하는 경우에는 return FALSE를 해줍니다.

■ 게임가드 종료

- 전역 변수로 CNPGameLib 객체를 생성한 경우 - 게임이 종료될 때 자동으로 CNPGameLib 객체의 소멸자가 호출되면서 게임가드 종료 처리가 수행됩니다.
- new를 사용하여 동적으로 객체를 생성한 경우 - 동적으로 생성한 CNPGameLib 객체를 게임 종료 처리 부분에서 반드시 delete 해주셔야 합니다.

■ 게임가드 실행 일시 중지 방법

클라이언트 개발 중에는, 게임가드가 실행되면 프로그램 디버깅이 안 될 수 있으므로 게임가드 실행을 일시 중지시키고 작업하시면 편리합니다.

NPGameLib.h를 인클루드 하기 전에 NO_GAMEGUARD를 디파인해주시면 됩니다.

```
#define NO_GAMEGUARD  
#include "NPGameLib.h"
```

2. 게임 실행 중에 GameMon 실행 여부 확인 방법

기본적으로 게임가드는 게임 클라이언트와 수초간격으로 서로의 실행 여부를 비밀 채널을 통해 통신하므로 한쪽이 종료될 경우 다른 쪽에도 종료 사실을 알고 함께 종료됩니다.

하지만 저수준 디버깅에 대비해 게임 곳곳에서 그리고 타이머 등을 통해 npgl.Check() 함수를 호출해 게임가드가 실행되어 있는지 확인해주시면 좋습니다.

Check 주기는 5-10초에 한 번 정도가 적당합니다.

```
if (npgl.Check() != NPGAMEMON_SUCCESS) bAppExit = true; // 게임 종료
```

3. 사용자의 아이디 전달

사용자가 게임에 로그인할 때 npgl.Send() 함수로 사용자의 ID를 GameMon으로 알려주면, 그 사용자의 불법 프로그램 사용 여부를 확인할 수 있고, 고객 지원시에 유용하게 사용할 수 있습니다. 사용자 아이디를 게임가드에 전달하지 않을 경우 게임가드의 일부 기능이 OFF 가 되므로 주의하시기 바랍니다.

npgl.Send(szUserID); // 반드시 문자열 끝에 종료(NULL) 를 포함시켜서 전달해야 합니다.

불법 프로그램이 발견되면 게임 클라이언트의 콜백 함수로 NPGAMEMON_GAMEHACK_DETECT 또는 NPGAMEMON_GAMEHACK_KILLED 메시지가 전달됩니다.

이 때 콜백 함수 내에서 불법 프로그램의 프로세스 이름을 확인하려면 npgl.GetInfo() 또는 GetInfoFromGameMon() 함수를 사용하시면 됩니다.

```
strcpy(g_szHackInfo, npgl.GetInfo());
```

만약 GetInfo()의 리턴값이 NULL이면 독립적인 프로세스로 존재하지 않아 프로세스 이름을 얻을 수 없는 형태의 게임핵일 수 있습니다.

이러한 정보들을 게임 서버로 전송하여 사용자 정보와 함께 로그를 남기면 사용자들의 불법 프로그램 사용 여부를 확인하실 수 있습니다.

* GetInfo() 으로 리턴되는 정보는 해커에게 핵툴 bypass 의 유용한 정보가 될 수 있습니다. 그러므로 게임핵 발견 메시지를 유저에게 표시할때에는 구체적인 정보는 빼 주시기를 부탁드립니다.

* 추가로 다음의 게임핵진단 전용 콜백메세지를 통해서 불법 프로그램 사용자를 로깅할 수 있습니다. NPGAMEMON_GAMEHACK_REPORT 메시지는 게임가드가 해킹을 파악하여 게임클라이언트에게 암호화된 정보를 전달 합니다. 콜백 함수에 이 메시지가 전달 될 경우 해당 데이터를 암호화한 상태 그대로 서버로 전송하여 서버에서 데이터를 복호화 하여 로깅을 통해서 해당 사용자를 제재할 수 있는 근거로 활용할 수 있습니다. 구현방법은 섹션 4, 5, 6 을 참고하시기 바랍니다.

로그형식은 ‘|’ 문자로 구분되며 다음은 로그 형식입니다.

핵종류|게임가드버전|날짜|핵이름|핵패턴번호|상세정보|핵파일TimeStamp|

서버에서 전달받은 해킹데이터에 대한 복호화 처리는 “게임가드 서버인증 매뉴얼 ”의 21 Page에 “DecryptHackData” 함수 설명을 참고하시기 바랍니다.

4. 게임가드의 메시지 종류 및 처리 방법

- NPGAMEMON_COMM_ERROR - GameMon과의 통신 채널이 끊어졌습니다. 보통 GameMon이 비정상적으로 종료되었을 경우이므로 게임도 종료해줍니다.
- NPGAMEMON_COMM_CLOSE - GameMon이 정상적으로 종료되어 보내는 메시지입니다. 게임도 종료해줍니다.
- NPGAMEMON_SPEEDHACK - 스피드핵이 감지되었습니다. 게임 종료 처리를 시작하고 종료 직전에 적절한 메시지를 출력해줍니다.
- NPGAMEMON_GAMEHACK_KILLED - 게임핵이 실행되었지만 성공적으로 강제 종료시켰습니다. 게임을 계속 진행해도 무방하지만 종료하기를 권장합니다.
- NPGAMEMON_GAMEHACK_DETECT - 게임핵이 발견되었습니다. 강제 종료가 적합하지 않은 경우이므로 게임 종료 처리를 시작하고 종료 직전에 적절한 메시지 출력해줍니다.
-

- NPGAMEMON_GAMEHACK_DOUBT - 게임핵으로 의심되는 프로그램이 실행 중 입니다. 혹은 게임이나 게임가드가 변조되었습니다. 게임 종료 처리를 시작하고 종료 직전에 불필요한 프로그램을 종료하고 다시 게임을 해보라는 메시지를 출력해줍니다.
- NPGAMEMON_INIT_ERROR - GameMon 실행 에러입니다. 게임 종료 처리를 시작하고 종료 직전에 에러코드인 dwArg 값과 함께 적절한 메시지를 출력해 줍니다.
- NPGAMEMON_GAMEHACK_REPORT - 게임핵이 발견되었지만 정상 동작합니다. 게임클라이언트는 전달받은 데이터를 서버에 전송하고 정상동작 합니다.

5. 게임가드 라이브러리 함수 설명

DWORD PreInitNPGameMon(LPCSTR szGameName);

- 게임가드를 초기화합니다.
- C++의 경우 CNPGameLib의 생성자에서 자동 호출됩니다.
- 리턴값: NPGAMEMON_SUCCESS - 초기화 성공, 그 외 값 - 에러 코드
- 적용 예 혹은 5장의 에러 코드 참조

DWORD InitNPGameMon();

- PreInitNPGameMon()의 초기화 결과를 알려줍니다.
- CNPGameLib의 Init()과 동일합니다.
- 리턴값: NPGAMEMON_SUCCESS - 초기화 성공, 그 외 값 - 에러 코드
- 적용 예 혹은 5장의 에러 코드 참조

void SetHwndToGameMon(HWND hWnd);

- 게임 클라이언트의 메인 윈도우 핸들을 GameMon으로 전달합니다.
- CNPGameLib의 SetHwnd()와 동일합니다.
- 리턴값: 없음

DWORD CheckNPGameMon();

- GameMon이 실행 중인지 확인합니다.
- CNPGameMon의 Check()와 동일합니다.
- 리턴값: NPGAMEMON_SUCCESS - 실행 중, 그 외 값 - GameMon 없음

BOOL SendUserIDToGameMon(LPCSTR szUserID);

- 사용자 ID를 GameMon으로 전달합니다.
- 리턴값: 0 - 실패, 1 - 성공

LPCSTR GetInfoFromGameMon();

- GameMon으로부터 발견된 해킹툴의 이름이나 정보를 얻어옵니다.
- CNPGameLib의 GetInfo()와 동일합니다.
- 리턴값: 해킹툴 정보 문자열 혹은 NULL

BOOL CloseNPGameMon();

- 게임가드를 종료합니다.
- C++의 경우 CNPGameLib의 소멸자에서 자동 호출됩니다.
- 리턴값: 0 - 실패, 1 - 성공

LPBYTE GetHackInfoFromGameMon(DWORD* dwSize);

- NPGAMEMON_GAMEHACK_REPORT 메시지가 왔을 경우 상세 데이터를 가져옵니다.
- 암호화된 데이터(바이너리)가 리턴되며 전달인자에 데이터 사이즈가 저장됩니다.
- 전달 받은 데이터가 없거나 에러는 0 이 리턴됩니다.
- return 값은 최대 1024byte 입니다.

DWORD GGGetLastError();

- 114 에러코드 발생시 원인을 파악할 수 있도록 세부적인 에러코드를 리턴합니다.
- '114에러 세부코드' 문서 참조

BOOL IsAdminPrivilege();

- 윈도우 계정이 Admin권한인지 알아보는 함수입니다.
- 제한된 계정일 경우 FALSE를 리턴합니다.

6. 적용 예

```
// 헤더 파일 인클루드
// #define NO_GAMEGUARD
#include "NPGGameLib.h"

.....

// 전역 변수로 객체 생성
CNPGGameLib npgl("Cube");

.....

// 게임 초기화 루틴(WinMain 등)에서 Init() 호출, 리턴값 처리
// 가능한 디스플레이 모드 전환이 있기 전에 최대한 일찍 처리
DWORD dwResult = npgl.Init();
if (dwResult != NPGAMEMON_SUCCESS)
{
    TCHAR msg[256];
    LPCSTR lpszMsg;

    // '6. 주요에러코드'를 참조하여 상황에 맞는 메시지를 출력해줍니다.
    switch (dwResult)
    {
        case NPGAMEMON_ERROR_EXIST:
            lpszMsg = TEXT("게임가드가 실행 중 입니다. 잠시 후나 재부팅 후에 다시 실행해보시기 바랍니다.");
            break;
        case NPGAMEMON_ERROR_GAME_EXIST:
            lpszMsg = TEXT("게임이 중복 실행되었거나 게임가드가 이미 실행 중 입니다. 게임 종료 후 다시 실행해보시기 바랍니다.");
            break;
        case NPGAMEMON_ERROR_INIT:
            lpszMsg = TEXT("게임가드 초기화 에러입니다. 재부팅 후 다시 실행해보거나 충돌할 수 있는 다른 프로그램들을 종료한 후 실행해 보시기 바랍니다.");
            break;
        case NPGAMEMON_ERROR_AUTH_GAMEGUARD:
        case NPGAMEMON_ERROR_NFOUND_GG:
```

```
case NPGAMEMON_ERROR_AUTH_INI:
case NPGAMEMON_ERROR_NFOUND_INI:
    lpszMsg = TEXT("게임가드 파일이 없거나 변조되었습니다. 게임가드 셋업 파일을 설치해
    보시기 바랍니다.");
    break;
case NPGAMEMON_ERROR_CRYPTAPI:
    lpszMsg = TEXT("윈도우의 일부 시스템 파일이 손상되었습니다. 인터넷 익스플로러(IE)를
    다시 설치해보시기 바랍니다.");
    break;
case NPGAMEMON_ERROR_EXECUTE:
    lpszMsg = TEXT("게임가드 실행에 실패했습니다. 게임가드 셋업 파일을 다시 설치해보시
    기 바랍니다.");
    break;
case NPGAMEMON_ERROR_ILLEGAL_PRG:
    lpszMsg = TEXT("불법 프로그램이 발견되었습니다. 불필요한 프로그램을 종료한 후 다시
    실행해보시기 바랍니다.");
    break;
case NPGMUP_ERROR_ABORT:
    lpszMsg = TEXT("게임가드 업데이트를 취소하셨습니다. 접속이 계속 되지 않을 경우 인터
    넷 및 개인 방화벽 설정을 조정해 보시기 바랍니다.");
    break;
case NPGMUP_ERROR_CONNECT:
    lpszMsg = TEXT("게임가드 업데이트 서버 접속에 실패하였습니다. 잠시 후 다시 접속하거
    나, 네트워크 상태를 점검해봅니다.");
    break;
case NPGAMEMON_ERROR_GAMEGUARD:
    lpszMsg = TEXT("게임가드 초기화 에러 또는 구버전의 게임가드 파일입니다. 게임가드 셋
    업파일을 다시 설치하고 게임을 실행해봅니다.");
    break;
case NPGMUP_ERROR_PARAM:
    lpszMsg = TEXT("ini 파일이 없거나 변조되었습니다. 게임가드 셋업 파일을 설치하면 해결
    할 수 있습니다.");
    break;
case NPGMUP_ERROR_INIT:
    lpszMsg = TEXT("npgmup.des 초기화 에러입니다. 게임가드폴더를 삭제후 다시 게임실행
    을 해봅니다.");
    break;
case NPGMUP_ERROR_DOWNCFG:
```



```
    lpszMsg = TEXT("게임가드 업데이트 서버 접속에 실패하였습니다. 잠시 후 재시도 해보거나, 개인 방화벽이 있다면 설정을 조정해 보시기 바랍니다.");
    break;
case NPGMUP_ERROR_AUTH:
    lpszMsg = TEXT("게임가드 업데이트를 완료하지 못 했습니다. 바이러스 백신을 일시 중지시킨 후 재시도 해보시거나, PC 관리 프로그램을 사용하시면 설정을 조정해 보시기 바랍니다.");
    break;
case NPGAMEMON_ERROR_NPSCAN:
    lpszMsg = TEXT("바이러스 및 해킹툴 검사 모듈 로딩에 실패 했습니다. 메모리 부족이거나 바이러스에 의한 감염일 수 있습니다.");
    break;
case NPGG_ERROR_COLLISION:
    lpszMsg = TEXT("게임가드와 충돌 프로그램이 발견되었습니다.");
    break;
default:
    // 적절한 종료 메시지 출력
    lpszMsg = TEXT("게임가드 실행 중 에러가 발생하였습니다. 게임 폴더 안의 GameGuard 폴더에 있는 *.erl 파일들을 Game1@inca.co.kr로 첨부하여 메일 보내주시기 바랍니다.");
    break;
}
wsprintf(msg, TEXT("게임가드 에러 : %lu"), dwResult);
MessageBox(NULL, lpszMsg, msg, MB_OK);

// 게임에 맞게 종료 코드
bAppExit = true;

return FALSE;
}

.....

// 윈도우 핸들이 생성된 후, 그리고 모든 그래픽, UI 관련 초기화가 끝난 후 SetHwnd() 호출
// 이 함수 호출 이후부터, CallBack 함수가 호출되기 시작함
nppl.SetHwnd(hWnd);

.....
```

```
// 사용자 로그인시 ID를 GameMon에 통보
npogl.Send(szUserID); // 반드시 문자열 끝에 종료(NULL) 를 포함시켜서 전달해야 합니다.
```

```
.....
// 게임 곳곳에서 또는 타이머에서 게임가드가 실행 중인지 확인
if (npogl.Check() != NPGAMEMON_SUCCESS)
    bAppExit = TRUE; // 게임에 맞게 적절히 종료시킴
.....
```

```
LPCTSTR g_szHackMsg [256] = { 0 };
// 메시지 처리 Callback 함수
// 게임 종료시에는 false를 return 해주고, 종료하지 않는 경우는 true를 return 합니다.
BOOL CALLBACK NPGameMonCallback(DWORD dwMsg, DWORD dwArg)
{
    switch (dwMsg)
    {
    case NPGAMEMON_COMM_ERROR:
    case NPGAMEMON_COMM_CLOSE:
        bAppExit = true; // 종료 코드
        return false;
    case NPGAMEMON_INIT_ERROR:
        wsprintf(g_szHackMsg, TEXT("게임가드 초기화 에러 : %lu"), dwArg);
        bAppExit = true; // 종료 코드
        return false;
    case NPGAMEMON_SPEEDHACK:
        wsprintf(g_szHackMsg, TEXT("스피드해킹이 감지되었습니다.));
        bAppExit = true; // 종료 코드
        return false;
    case NPGAMEMON_GAMEHACK_KILLED:
        wsprintf(g_szHackMsg, TEXT("게임해킹이 발견되었습니다.));
        bAppExit = true; // 종료 코드
        return false;
    case NPGAMEMON_GAMEHACK_DETECT:
        wsprintf(g_szHackMsg, TEXT("게임해킹이 발견되었습니다.));
        bAppExit = true; // 종료 코드
        return false;
    case NPGAMEMON_GAMEHACK_DOUBT:
        wsprintf(g_szHackMsg, TEXT("게임이나 게임가드가 변조되었습니다.));
        bAppExit = true; // 종료 코드
```

```
        return false;
case NPGAMEMON_GAMEHACK_REPORT:
    {

        DWORD dwHackInfoSize = 0;
        LPBYTE pHackInfo = NULL;
        pHackInfo = GetHackInfoFromGameMon(&dwHackInfoSize);
        // pHackInfo = npgl.GetHackInfo(&dwHackInfoSize); // C++ 일 경우.
        if (pHackInfo && dwHackInfoSize > 0)
        {
            // 아래 함수는 게임가드에서 제공하는 함수가 아닙니다.
            SendToHackLog(pHackInfo, dwHackInfoSize); // 서버로 데이터 전송.
        }
    }
    return true;
}

return true; // 계속 진행
}
```

```
.....

// 게임 종료 직전
// 게임가드 관련 메시지가 있으면 출력
if (g_szHackMsg[0])
{
    MessageBox(hWnd, g_szHackMsg, TEXT("nProtect GameGuard"), MB_OK);
}
```


[C에서의 적용 예]

```
// NPGameLib.h를 include 하기 전에 NPGAMELIB_C를 define해서 C linkage임을 알려줌
#define NPGAMELIB_C
#include "NPGameLib.h"

.....

// 게임 초기화 루틴(WinMain 등)에서 PreInitNPGameMon() 호출
PreInitNPGameMon("Cube");

// InitNPGameMon() 호출, 리턴값 처리
DWORD dwResult = InitNPGameMon();
if (dwResult != NPGAMEMON_SUCCESS)
{
    //// 각 리턴값 처리 방법은 C++ 적용 예 참조
}

.....

// 윈도우 핸들이 생성된 후, 그리고 모든 그래픽, UI 관련 초기화가 끝난 후
// SetHwndToGameMon() 호출. 이 함수 호출 이후부터, CallBack 함수가 호출되기 시작함
SetHwndToGameMon(hWnd);

.....

// 사용자 로그인시 ID를 GameMon에 통보
SendUserIDToGameMon(szUserID);

.....

// 게임 곳곳에서 게임가드가 실행 중인지 확인하는 CheckNPGameMon() 호출
if (CheckNPGameMon() != NPGAMEMON_SUCCESS)
    bAppExit = TRUE; // 게임에 맞게 적절히 종료시킴
```

```
//// Callback 함수 구현은 동일하므로 '적용 예' 참조  
// npgl.GetInfo() 대신 GetInfoFromGameMon() 사용
```

.....

```
// 게임 종료 부  
// 게임가드도 함께 종료시켜 줌  
CloseNPGameMon();
```

[C에서의 적용 예] 끝

7. 기타 1) 패킷 암호화 함수 (Peer to Peer 용)

P2P 구조로 클라이언트끼리 주고 받는 패킷을 간단히 암호화할 수 있는 함수들입니다.

패킷이 완전히 노출되는 것보다는 아주 간단한 암호화라도 하게 되면 분석과 조작이 많이 힘들어집니다.

1) 게임가드 패킷 암호화 적용 방법

- 초기화 중에 InitPacketProtect(“임의의 키 스트링”) 형식으로 키를 초기화 합니다. 이 스트링은 모든 클라이언트에서 동일해야 합니다. 만약 다르다면 상대 클라이언트는 올바른 패킷이 아닌 것으로 판단하고 접속을 해제할 것입니다. 오래된 클라이언트를 사용하지 못 하게 하기 위해 일부러 구 버전 클라이언트와 새 버전 클라이언트의 키 스트링을 다르게 변경할 수도 있습니다.
- 이제 클라이언트 사이에 주고 받는 패킷에 패킷 암/복호화 함수를 호출합니다.
- 패킷을 상대 클라이언트로 보내기 직전 - 즉 send 명령이 호출되기 직전에 EncryptPeerPacket() 함수로 패킷을 암호화 하고, 이 함수의 리턴값으로 패킷의 길이를 다시 설정해줍니다. 가능하면 모든 send 함수 앞에서 작업하기 보다는, 최종적으로 send 함수 호출하는 Wrapping 함수를 하나 만들어 그 함수 안에서 한번 작업하는 것이 좋습니다.
- 서버로부터 패킷을 받은 직후 - 즉 recv 명령이 호출된 후나 그 세션의 Queue에서 패킷을 읽었거나, 온전한 패킷 하나를 읽어 들였을 때, DecryptPeerPacket() 함수로 패킷을 복호화 하고, 이 함수의 리턴값인 원래 패킷의 길이로 나머지 처리를 계속합니다. 단, 이때 이 함수의 리턴값이 0 인지 반드시 확인하여 0일 경우 잘못된 패킷이므로 적절한 메시지를 보여주거나 종료하도록 합니다.
- 만약 게임 서버-클라이언트 사이의 패킷을 암복호화하려면 EncryptPacket()/DecryptPacket()을 사용하면 됩니다. EncryptPeerPacket()/DecryptPeerPacket()은 클라이언트 사이의 패킷을 암복호화 할 때만 사용해야 합니다.

2) 패킷 암호화 적용 예

```
// 클라이언트 초기화 중
InitPacketProtect(“패킷게임가드”);

// 상대 클라이언트로 송신 함수
void SendToPeer(char *lpData)
{
    // lpData의 첫 4 bytes는 길이 필드로서, 실제 패킷의 사이즈가 들어 있다고 가정
    // dwPacketLength에는 길이 필드의 사이즈는 포함되지 않음.
    DWORD dwPacketLength = *((LPDWORD) lpData);
    DWORD dwTotoalLength = EncryptPeerPacket(lpData + 4, dwPacketLength);

    // 암호화 후에는 패킷 크기가 조금 늘어나므로 늘어난 크기로 패킷 크기 정보를 다시 설정
```

```
memcpy(lpData, &dwTotalLength, sizeof(DWORD));
...
// dwTotalLength에는 길이 필드는 포함되지 않으므로 길이 필드의 크기를 더해줌.
send(socket, lpData, dwTotalLength + sizeof(DWORD), 0);
...
}

// 상대 클라이언트로부터의 수신 함수
void RecvFromPeer(char *lpData, DWORD dwLength)
{
    // lpData에는 이미 하나의 온전한 패킷이 들어 있음.
    DWORD dwRealLength = DecryptPeerPacket(lpData, dwLength);
    If (dwRealLength == 0)
    {
        // 잘못된 패킷. PPGetLastError() 값을 포함한 적절한 로그를 남긴 후 접속을 해제함.
        return;
    }
    // 암호화 전의 원래 길이는 dwRealLength에 들어 있음. 나머지 처리를 계속함.
    ...
}
```

3) 패킷 암호화 함수 프로토타입

DWORD InitPacketProtect(LPCSTR lpszUserKey);

- PakcetProtect의 클라이언트키를 초기화 합니다.
- lpszUserKey에 따라 암/복호화에 사용되는 키 값이 달라지게 됩니다. 그러므로 모든 클라이언트의 lpszUserKey는 반드시 동일하게 설정하셔야 합니다.
- 이전 버전의 클라이언트를 사용하지 못하게 할 목적으로, 새 버전 클라이언트의 UserKey를 바꾸어 주는 것도 좋은 방법입니다.
- 리턴값: 0 - 실패, 1 - 성공

DWORD EncryptPeerPacket(LPVOID lpData, DWORD dwLength);

- 패킷을 암호화 합니다.
- lpData는 패킷의 길이를 제외한, 실제 내용이 들어있는 버퍼의 포인터를 넘겨주시면 됩니다.
- dwLength는 패킷의 길이 필드를 제외한, 실제 패킷 내용의 길이를 넘겨주시면 됩니다.
- 암호화 후에는 패킷의 길이가 조금 늘어나게 됩니다. 시퀀스번호, CRC, Key ID, Padding Bytes 등의 정보 때문이며, 늘어나는 길이는 32Bytes 이하입니다.
- 이 함수를 호출한 후에는 패킷의 길이 필드를 이 함수의 리턴값으로 재설정 해주셔야 합니다.

- 리턴값: 0 - 실패, 그 외 - 암호화된 패킷의 길이

DWORD DecryptPeerPacket(LPVOID lpData, DWORD dwLength);

- 패킷을 복호화 합니다.
- lpData에는 길이 필드를 제외한, 실제 내용이 들어있는 버퍼의 포인터를 넘겨주시면 됩니다.
- dwLength는 실제 패킷의 길이를 넘겨주시면 됩니다.
- 성공시에는 패킷 뒤에 붙어있던 추가적인 정보는 제거되며 원래 패킷의 길이가 리턴됩니다.
- 실패시는 올바른 패킷이 아니기 때문이므로 그 사용자의 접속을 해제하도록 하시면 됩니다. PPGetLastError() 함수를 통해 에러코드를 확인할 수 있습니다.
- 리턴값: 0 - 실패, 그 외 - 암호화된 패킷의 길이

DWORD PPGetLastError();

- DecryptPacket 실패시, 실패한 에러코드를 알려줍니다.
- 에러코드 1 - 키가 일치하지 않습니다. 암호화되지 않은 패킷이거나, UserKey가 다른 클라이언트로부터 온 패킷일 수 있습니다.
- 에러코드 2 - 시퀀스번호가 최근의 패킷과 같습니다. 패킷 리플라이 공격이 원인일 수 있습니다.
- 에러코드 3 - 시퀀스번호가 비정상입니다. 서버와 클라이언트의 약속된 시퀀스번호가 일치하지 않습니다. 네트워크 문제로 패킷 몇 개가 유실 되었을 수 있습니다. TCP 프로토콜에서는 거의 발생하지 않는 에러입니다.
- 에러코드 4 - 패킷 CRC 체크에 실패했습니다. 패킷 에디터 등으로 고의로 패킷을 조작했을 수 있습니다.
- 리턴값: 에러코드

8. 기타 2) D3DDevice의 VirtualTable을 후킹검사(NPGameLib 860이하 버전에서 지원)

// D3DDevice의 VirtualTable을 후킹중인지 확인하는 CheckD3DDevice() 호출

```
if (CheckD3DDevice((LPVOID)m_pd3dDevice, "d3d8.dll"))
```

```
    bAppExit = true; // 게임에 맞게 적절히 종료시킴
```

1) VirtualTable 후킹검사 적용 예

```
// Direct3DCreate9 객체 초기화 함수
```

```
// (다른 DirectX 버전도 해당 버전에 맞게 인자 및 함수명을 수정하셔서 사용하시면 됩니다.)
```

```
if(NULL == (m_pD3D = Direct3DCreate9(D3D_SDK_VERSION)))
```

```
...
```

```
if (FAILED(m_pD3D->GetAdapterDisplayMode(D3DADAPTER_DEFAULT,&CurrentMode)))
```

```
...
```

```
if( FAILED( m_pD3D->CreateDevice( D3DADAPTER_DEFAULT, D3DDEVTYPE_HAL, hWnd,
```

```
    D3DCREATE_HARDWARE_VERTEXPROCESSING,
```

```
    &m_PresentParameters, &m_pD3DDevice ) ) )
```

```
...
```

```
// D3DDevice의 VirtualTable을 후킹검사
```

```
if (CheckD3DDevice((LPVOID)m_pD3DDevice, "d3d9.dll"))
```

```
{
```

```
    // 변조되었음. GetLastError() 값을 포함한 적절한 로그를 남긴 후 접속을 해제함.
```

```
    bAppExit = true;
```

```
}
```

2) VirtualTable 후킹검사 함수 프로토타입

```
BOOL CheckD3DDevice(LPVOID lpD3DDevice, LPSTR lpszDll);
```

- D3DDevice의 VirtualTable을 후킹하는 방식을 검출 합니다.
- lpD3DDevice : m_pD3D->CreateDevice로 얻은 LPDIRECT3DDEVICE 포인터
- lpszDll : 사용하는 d3d DLL의 이름
- 리턴값: 0 - 변조안됨, 1 - 변조됨

9. 기타 3) D3DDevice의 VirtualTable을 후킹검사 (NPGameLib 87이상 버전에서 지원)

```
//direct3d의 정보를 세팅하기 위해 호출 해줌
if(!SetD3DDeviceInfo(9, &g_pD3DDevice))
    bAppExit = true; //게임에 맞게 적절히 종료시킴
```

1) SetD3DDeviceInfo 적용 예(direct3d 9사용시)

```
// Direct3DCreate9 객체 초기화 함수
// (다른 DirectX 버전도 해당 버전에 맞게 인자 및 함수명을 수정하셔서 사용하시면 됩니다.)
if(NULL == (m_pD3D = Direct3DCreate9(D3D_SDK_VERSION)))
    ...

if (FAILED(m_pD3D->GetAdapterDisplayMode(D3DADAPTER_DEFAULT,&CurrentMode)))
    ...

if( FAILED( m_pD3D->CreateDevice( D3DADAPTER_DEFAULT, D3DDEVTYPE_HAL, hWnd,
    D3DCREATE_HARDWARE_VERTEXPROCESSING,
    &m_PresentParameters, &g_pD3DDevice ) ) )
    ...

// D3DDevice의 pointer정보 세팅(최초 1회만 호출해주면 됨)
if (SetD3DDeviceInfo(9, &g_pD3DDevice))
{
    //D3DDevice정보 세팅을 실패함. 정보 세팅을 방해받았을 수 있음. 적절한 종료 처리
    bAppExit = true;
}
}
```

2) VirtualTable 후킹검사 함수 프로토타입

```
BOOL SetD3DDeviceInfo(DWORD dwD3DVer, LPVOID *lppD3DDevicePtr);
```

- D3DDevice의 VirtualTable을 후킹하는 방식을 검출 검출하기 위해 정보를 세팅합니다.
- lppD3DDevicePtr : m_pD3D->CreateDevice로 얻은 LPDIRECT3DDEVICE 포인터변수의 주소

리턴값: 0 - 세팅 실패(실패), 1 - 세팅이 정상적으로완료됨(정상), 2 - 모듈버전으로 인한 실패(정상)

5장 게임가드 서비스

1. 게임가드 FAQ

- Q: 게임가드 사용 후 게임이 느려졌습니다.
A: 컴퓨터의 사양을 확인해 보십시오. 만약 게임사양에 부합하는 경우에도 느리다면 현재 실행 중인 백신 프로그램의 설정을 확인해 보십시오. 백신 프로그램의 실시간 바이러스 검사나 다른 불필요한 기능을 최소한으로 설정하시고 다시 실행해 보시길 권합니다.
일부의 경우, 백신 프로그램과의 충돌이 일어나기도 하니, CPU 점유율이 높은 백신의 경우, 기능을 최소한으로 설정해 주시면 보다 원활한 게임실행에 도움이 됩니다.
- Q: 게임가드가 해킹툴이 발견되었다면서 게임을 종료시킵니다.
A: 해킹툴은 물론, 바이러스나 스파이웨어에 의해 실행이 제대로 되지 않는 경우가 많습니다. 사용자들에게 최신 백신과 Ad-aware와 같은 스파이웨어 검사 프로그램을 사용하여 PC를 검사해보도록 권합니다.
- Q: 게임가드 업데이트에 실패했다고 합니다. 어떻게 해야 할지요? (380번 에러코드)
A: 게임가드 설치 파일을 다운 받으시고 설치를 시도해 보십시오. 윈도우 XP 서비스팩2를 사용하신다면 방화벽 설정을 확인해 보시기 바랍니다. 일부의 경우, 컴퓨터를 재시작하면 해결되기도 합니다.
A2: Sygate Firewall 등 방화벽 프로그램이 게임가드를 차단하고 있을 가능성이 있습니다. 해당 방화벽 프로그램을 가동시킨 후, 셋팅 부분에서 Block목록에 GameGuard.des가 들어있지 않은지 확인을 바랍니다.
- Q: 로딩하는데 시간이 매우 오래 걸리고 결국에는 114번 에러코드가 발생하며 게임실행이 되지 않습니다.
A: CPU를 굉장히 많이 점유하는 프로그램이 있을 수 있습니다. 또는 비디오 드라이버나 사운드 드라이버가 구 버전일 가능성이 있습니다. 바이러스와 스파이웨어 치료를 해주신 후, 드라이버를 최신 버전으로 업데이트 해주시기 바랍니다.
A2: 사용자의 시스템 사양으로 사용하기가 버거운 백신을 사용하고 있을 수 있습니다(Norton 2005, McAfee, VirusBuster 등등). 해당 백신의 설정부분으로 들어가신 후, 인터넷 감시, 아웃룩 체크, 스크립트 감시 등 불필요한 옵션을 체크해제 해 주시기 바랍니다.
- Q: 게임가드 업데이트가 되지 않으며 124번, 150번, 153번 에러코드가 발생합니다.
A: 게임가드를 수동으로 패치해 주시기 바랍니다.
- Q: 게임실행이 되지 않으며 100번 에러코드가 발생합니다.
A: 컴퓨터가 바이러스에 감염되어 있습니다. 백신을 최신 버전으로 업데이트 하신 후, 전체 검사를 해 주시기 바랍니다. 검사를 하셔도 여전히 100번 에러코드가 발생한다면, 현재 백신이 감지하지 못하는 신종 바이러스일 가능성이 있으니, 다른 백신을 이용해보시기 바랍니다(각 백신사 홈페이지에서 서비스하고 있는 무료검사를 이용)
- Q: 게임가드 실행이 되지 않고 360번 에러코드가 발생합니다.
A: 하드디스크 드라이브 마우스 오른쪽 버튼 - 도구 - 디스크 검사를 하신 후, 게임가드 폴더를

통째로 지우시고 게임가드 수동 패치 파일을 다운로드 하시고 게임을 실행해보시기 바랍니다.

- **Q:** 게임가드 실행이 되지 않고 **361번** 에러코드가 발생합니다.

A: 게임가드를 제대로 업데이트 받고 있지 못하고 있습니다. 고객님의 인터넷 설정을 살펴봐주시길 바라며, 회선은 이상 없는지 그리고 itop등의 보안프로그램을 사용하고 있지는 않은지 확인 부탁드립니다. 인터넷 연결하기 전 인증 제도가 필요한 것은 아닌지 체크해보시기 바랍니다.

- **Q:** 게임가드 업데이트가 진행이 되지 않고 **350번** 에러코드가 발생합니다.

A: 업데이트 취소 버튼을 누르셨습니다. 인터넷 회선에 따라 업데이트가 빨리 진행이 되지 않는 것일 수도 있습니다. 또는 ftp나 http접속이 차단되어 있는 경우도 있으니, 게임가드 창 왼쪽 하단에 Retry 버튼이 생성된다면 눌러서 업데이트 받아보시기 바랍니다.

- **Q:** 게임클라이언트 실행파일에 대한 압축암호화를 통해서 보안을 강화하고 싶습니다.

A: 실행파일 압축암호화는 게임가드 NPGE 에 대한 사용설명서를 참고하시기 바랍니다.

2. 주요 에러 코드 - Init()의 리턴값

- NPGAMEMON_ERROR_EXIST (110) - GameMon이 이미 실행되어 있습니다. GameMon 프로세스를 종료하거나 재부팅 후 다시 실행합니다.
- NPGAMEMON_ERROR_GAME_EXIST (115) - 게임이 중복 실행되었습니다. 혹은 게임가드가 이미 실행되어 있습니다. 게임을 종료 한 후 다시 실행합니다.
- NPGAMEMON_ERROR_NPSCAN (112) - 바이러스 및 해킹툴 검사 모듈 로딩에 실패 했습니다. 메모리 부족이거나 바이러스에 의한 감염일 수 있습니다.
- NPGAMEMON_ERROR_INIT (114) - GameMon 초기화에 실패했습니다. 재부팅 후 다시 실행해보거나, 충돌할 만한 다른 프로그램들을 종료한 후 다시 실행합니다. 혹은 바이러스나 스파이웨어 검사를 해봅니다.
- NPGAMEMON_ERROR_NFOUND_GG (153) - 아래 참고
- NPGAMEMON_ERROR_AUTH_GAMEGUARD (124) - GameGuard.des 파일이 없거나 변조되었습니다. 게임가드 셋업 파일을 설치하면 해결할 수 있습니다.
- NPGAMEMON_ERROR_AUTH_INI (120) - 아래 참고
- NPGAMEMON_ERROR_CORRUPT_INI (141) - 아래 참고
- NPGAMEMON_ERROR_CORRUPT_INI2 (142) - 아래 참고
- NPGAMEMON_ERROR_NFOUND_INI (150) - 게임가드 설정 파일이 없거나 변조되었습니다. 게임가드 셋업 파일을 다운 받아 게임가드 폴더에 다시 설치합니다.
- NPGAMEMON_ERROR_CRYPTAPI (155) - 윈도우의 시스템 파일이 손상되었습니다. 바이러스 검사를 해보기를 권하며, 인터넷 익스플로러를 업그레이드 하거나 재설치 하면 해결할 수 있습니다.
- NPGAMEMON_ERROR_EXECUTE (170) - 게임가드 프로세스 실행에 실패했습니다. 게임가드 셋업 파일을 다시 설치하고 게임을 실행해봅니다.
- NPGAMEMON_ERROR_ILLEGAL_PRG (200) - 불법 프로그램이 발견되었습니다. 불필요한 프로그램을 종료한 후에 다시 게임을 실행해봅니다.
- NPGAMEMON_ERROR_GAMEGUARD (230) - 게임가드 초기화 에러 또는 구버전의 게임가드 파일

입니다. 게임가드 셋업파일을 다시 설치하고 게임을 실행해봅니다.

- NPGMUP_ERROR_PARAM (320) - ini 파일이 없거나 변조되었습니다. 게임가드 셋업 파일을 설치하면 해결할 수 있습니다.
- NPGMUP_ERROR_INIT (330) - npgmup.des 초기화 에러입니다. 게임가드폴더를 삭제후 다시 게임실행을 해봅니다.
- NPGMUP_ERROR_DOWNCFG (340) - 다운로드에 실패하였습니다. 네트워크 상태가 좋지 않거나 인터넷 설정에 문제가 있을 수 있습니다. 혹은 개인방화벽 설정을 살펴봅니다.
- NPGMUP_ERROR_ABORT (350) - 업데이트 중 취소 버튼을 눌렀습니다. 접속이 되지 않아 취소한 것이라면, 다시 시도해보거나 네트워크 상태를 점검해봅니다. 혹은 개인방화벽 설정을 살펴봅니다.
- NPGMUP_ERROR_AUTH (360) - 업데이트가 제대로 이루어지지 않았습니다. 바이러스 백신을 일시 중지 시킨 후 재시도 해보시거나, PC 관리 프로그램을 사용하면 설정을 조정해 보시기 바랍니다.
- NPGMUP_ERROR_CONNECT (380) - 게임가드 업데이트 서버 접속에 실패하였습니다. 잠시 후 다시 접속하거나, 네트워크 상태를 점검해봅니다.
- 그 외의 에러는 “게임가드 실행 중 에러가 발생하였습니다. 게임 폴더 안의 GameGuard 폴더에 있는 *.erl 파일들을 (국내: game1@inca.co.kr 해외: game2@inca.co.kr) 로 첨부하여 메일 보내 주시기 바랍니다.” 형태로 메시지를 출력해 주시면 됩니다.

3. 고객지원

㈜잉카인터넷에서는 nProtect GameGuard 사용자에게 E-Mail, 전화를 통해 고객 지원을 하고 있습니다.

- 게임가드와 관련된 에러는 메시지와 에러코드를 사용자에게 보여주면 고객 지원시 도움이 됩니다.
- 자주 발생하고 해결 방법이 명확한 문제는 게임에서 직접 해결법을 메시지로 보여주면 좋습니다.
- 바이러스나 스파이웨어에 의해 실행이 제대로 되지 않는 경우가 많습니다. 사용자들에게 최신 백신과 스파이웨어 검사 프로그램을 사용하여 PC를 검사해보도록 권합니다.
- 그래도 해결되지 않는 문제는 문제가 발생한 직후에 게임 폴더 안의 GameGuard 폴더에 있는 *.erl 파일들을 (국내: game1@inca.co.kr 해외: game2@inca.co.kr) 로 첨부하여 메일 주시면 분석 후 해결하도록 하겠습니다.

게임폴더 위치 찾기: 게임 아이콘 바로가기->오른쪽 버튼 클릭->속성(등록정보) 클릭->대상 찾기 클릭

- nProtect GameGuard 사용시 문의 사항이나 문제점이 있으신 경우 언제든지 아래와 같은 방법으로 ㈜잉카인터넷으로 연락 주시기 바랍니다.

- 1) 게임실행이 안 되거나 오류 발생 시점에서 게임을 종료합니다.
- 2) 게임실행이 안되는 상황 설명과 함께 게임 폴더 안의 GameGuard 폴더에 있는 *.erl 파일들을 첨부해서 메일 주시면 됩니다.

▪ 접수 방법

1) 이메일 접수

일반유저 : (국내: game1@inca.co.kr 해외: game2@inca.co.kr) 을 통해 문의사항 접수

고객사 : GameService@inca.co.kr 을 통해 문의사항 접수

2) 전화 접수

고객지원 : 02-6220-8114



■ (주)잉카인터넷 게임보안 사업본부 글로벌사업부 Feedback

메일: gameservice@inca.co.kr

www.gameguard.kr

서울시 구로구 구로3동 235-2 에이스 하이엔드타워 12층 1204호 (우 152-848)

Security, For a More Joyful Gameplay.

Copyright ©INCA Internet Corp. All rights reserved.

MEMO