NPGE(nProtect Game Encrypter) Manual

-NPGE 매뉴얼-

■ (주)잉카인터넷 게임보안 사업본부 게임가드개발팀 개발파트

DV-111103

Ver. 1.0

업체 제공용

INCA
I.n.t.e.r.n.e.t

NPGE (nProtect Game Encrypter) 매뉴얼 V 2.00

1. 개요

- NPGE는 게임 클라이언트 실행 파일(EXE) 또는 모듈(DLL)을 암호화하는 툴입니다.
- 파일은 암호화된 상태로 저장되어 있으며, 실행하면 메모리 상에서 스스로 복호화하여 실행시킵니다.

2. 게임 클라이언트 암호화의 필요성

- 암호화되어 있지 않은 실행 파일은 역어셈블러(disassembler)로 쉽게 전체 내용을 분석할 수 있고, 코드나 데이터의 수정이 매우 쉽습니다.
- 비교적 간단한 코드 수정으로 게임가드를 호출하는 부분을 건너 뛰게 할 수도 있습니다.
- 파일 자체가 수정될 수 있으므로 게임가드에 의한 코드 영역 변조 여부 확인이 불가능합니다.
- 게임 서버에서 클라이언트 파일의 변조를 막기 위해 Checksum 검사를 수행하는 코드가 있더라도, 그 부분 역시 정상 Checksum으로 속이도록 간단히 수정할 수 있습니다.

3. 암호화의 효과

- 파일 자체로는 분석과 수정이 힘들어집니다. 파일은 암호화되어 있으므로 역어셈블이 되지 않고, 수정할 부분을 찾더라도 파일 자체를 수정할 수 없습니다. 그러므로 파일 자체의 무결성을 크게 향상시킬 수 있습니다.
- 해킹툴은 암호화가 풀린 후의 메모리 상의 코드나 데이터를 조작할 수 밖에 없습니다. 즉 조작의 대상이 메모리로 제한되는 것입니다. 게임가드는 해킹툴로부터 게임 프로세스의 메모리 접근을 차단하므로 해킹툴이 게임을 조작하기가 매우 힘들어집니다.
- 본 암호화 툴을 사용하여 클라이언트를 암호화할 경우, 게임가드에서 암호화전의 파일 Checksum과 실행 후 메모리상에 풀린 코드의 Checksum 비교가 가능하게 됩니다. 따라서 게임 클라이언트는 코드 영역의 변조 없이 메모리 상에서 실행될 것을 기대할 수 있습니다. (Protection Level 4는 검사 불가)

4. 암호화의 한계

- 해커가 클라이언트 실행 파일을 unpack하여 암호화되지 않은 상태의 파일로 만들고, 그것이 실행 가능하게까지 만든다면 결국 암호화를 하기 전과 동일한 상태가 됩니다. 실제로 대부분의 암호화 프로텍터들은 그것을 쉽게 풀어주는 unpacker들이 존재합니다.
- 이런 상황을 대비해 게임가드에서 클라이언트가 암호화되어 있지 않으면 GAMEHACK_DOUBT 메시지를 발생시키도록 옵션을 줄 수 있습니다. 이 옵션을 사용하길 원하면 게임가드 담당자에게 요청하시면 됩니다.
- 이 옵션의 단점은 개발 중에는 불편할 수 있다는 것입니다. 개발자가 빌드 후 바로 클라이언트를 실행시키면 역시 위 메시지가 나오면서 게임이 종료될 것입니다.
- 그럴 경우에는 게임가드를 Disable 해놓고 개발 작업을 할 수 있습니다. 혹은, NPGEClient를 로그인한 상태로 실행시켜놓으면 게임가드는 위 옵션을 무시하게 됩니다.

5. NPGE Client 사용법

- 발급 받은 ID와 Password를 입력하여 로그인 합니다.
- 암호화할 게임 클라이언트 실행 파일을 선택합니다. (Drag&Drop 지원)
- Protection Level을 선택합니다. (마지막으로 사용했던 Level이 저장됨)
- Encrypt 버튼을 클릭합니다.
- 암호화가 완료되면 선택한 파일 이름으로 암호화된 파일이 생성됩니다.
- 암호화하기 전의 파일은 .bak 파일로 저장됩니다.
- 암호화된 실행 파일로 정상적으로 실행되는지 테스트합니다.
- 만약 비정상 종료가 되거나 다른 문제가 발생하면 .bak 파일을 사용해서 복원한 뒤, Protection Level을 조정하여 다시 암호화하고 테스트합니다.
- 커맨드라인 작업을 원하면 /ID:Password:ProtectionLevel:Filename[;Parameter][;Working Directory] 형식으로 파라미터를 전달합니다.
- NPGEClient 1.90부터는 ProtectionLevel을 반드시 지정해야 합니다.
- Filename에는 전체 경로가 포함될 수 있습니다. 만약 파일 경로에 공백이 포함될 수 있다면 " "로 묶습니다.
- Parameter 앞에는 콜론(:)이 아니라 세미콜론(;)으로 구분하여야 하는 점에 주의 바랍니다. Parameter는 생략될 수 있습니다. 만약 Parameter가 지정되면 Filename 파일을 암호화한 후 자동으로 지정된 Parameter를 사용하여 Filename 파일을 실행시킵니다. 일반적으로 암호화 후 게임을 한번 실행시켜야 하는 경우(L2+, L5, L6)에 사용할 수 있습니다. 지정된 Parameter가 전달되면 게임은 게임가드 초기화까지 완료한 후 자동으로 종료시키면 NPGE 패킹 작업을 일괄 처리하는데 도움이 될 것입니다. 물론 게임 실행에 필요한 나머지 파일들은 Filename에 지정된 경로에 적절히 위치하고 있어야 합니다.
- 만약 패킹한 파일이 DLL이라면 Parameter에 실행할 EXE 이름과 파라미터를 " "로 묶어 지정합니다. 하지만 EXE를 패킹하는 경우라면 " "로 묶지 말아야 합니다.

- **예제 (EXE 패킹)**

EX) NPGEClient.exe /inca:inca:client.exe - X (반드시 ProtectionLevel을 지정해야 합니다)
EX) NPGEClient.exe /inca:inca:5:client.exe - Level 5 사용
EX) NPGEClient.exe /inca:inca:25:client.exe - Level 2.5 사용
EX) NPGEClient.exe /inca:inca:22:client.exe - Level 2 + DLL & rdata protection 사용
EX) NPGEClient.exe /inca:inca:5:"client 2.exe" - 파일 경로에 공백이 포함될 경우
EX) NPGEClient.exe /inca:inca:5:"client 2.exe";run
- 암호화 후 run이라는 파라미터로 자동 실행
EX) NPGEClient.exe /inca:inca:5:"client 2.exe";-arch d:₩work₩game run
- 파라미터가 복잡한 경우에도 EXE 패킹시에는 파라미터를 " "로 묶지 말아야 합니다.

- **예제 (DLL 패킹)**

  EX) NPGEClient.exe /inca:inca:6:engine.dll;"client.exe run"
  - dll 암호화 후 client.exe를 run이라는 파라미터로 자동 실행

  EX) NPGEClient.exe /inca:inca:6:"d:₩work₩game₩engine.dll";"d:₩work₩game₩client.exe run"
  - 전체 경로를 지정하여 dll 암호화 후 client.exe를 run이라는 파라미터로 자동 실행

  EX) NPGEClient /inca:inca:6:"d:₩game₩bin₩engine.dll";"d:₩game₩bin₩cli.bin run";"d:₩game"
  - 실행할 파일이 있는 폴더와 Working Directory가 다른 경우
  - 파라미터에 경로가 포함되는 등 복잡한 경우에는 Working Directory를 지정해주어야 함

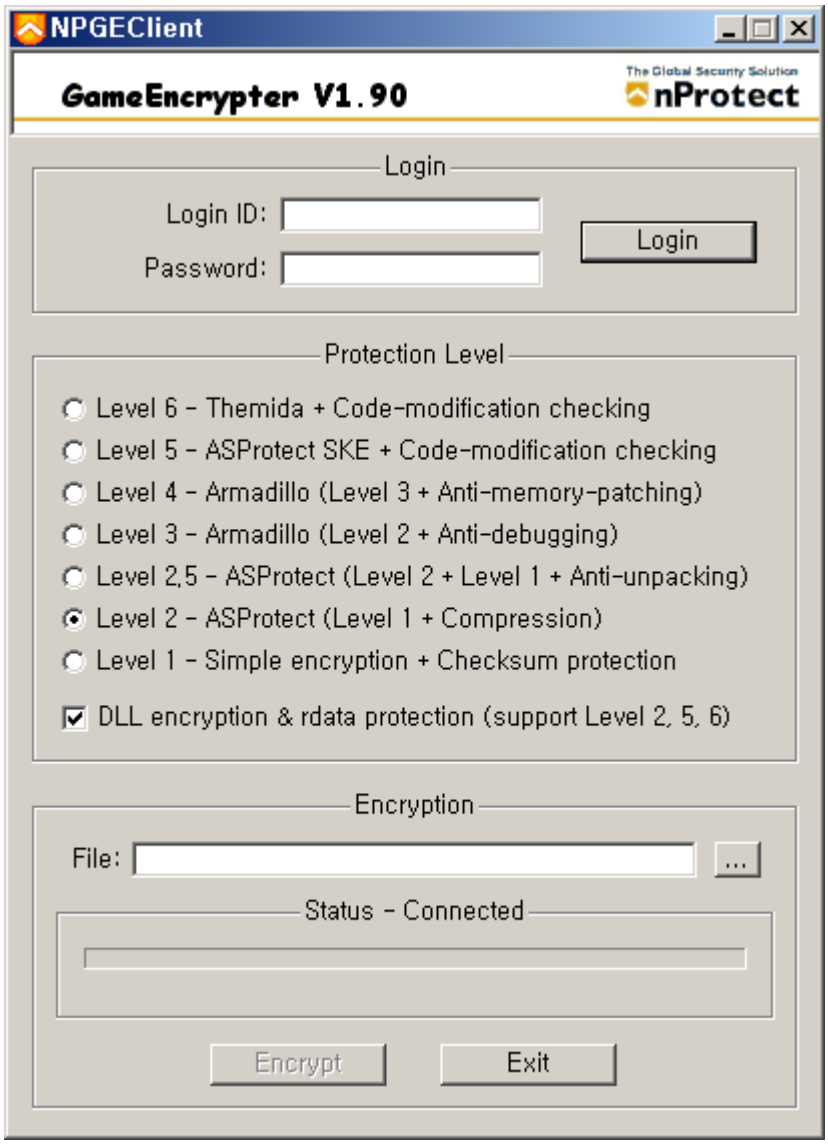
- 커맨드라인 실행시에는 일반적으로 메시지 박스가 표시되지 않습니다. 대신 batch 파일을 만들어 NPGEClient.exe의 종료코드로 암호화 작업의 성공 여부를 확인할 수 있습니다. 0은 성공, 1은 실패입니다.

  EX) EXE 패킹 배치 파일

```
@echo off
NPGEClient.exe /inca:inca:6:"d:₩work₩game₩client.exe";run
IF ERRORLEVEL 1 GOTO END
:SUCCESS
ECHO SUCCESS
:END
ECHO END
```

  EX) DLL 패킹 패치 파일

```
@echo off
NPGEClient.exe /inca:inca:6:"d:₩work₩game₩engine.dll";"d:₩work₩game₩client.exe run"
IF ERRORLEVEL 1 GOTO END
:SUCCESS
ECHO SUCCESS
:END
ECHO END
```



6. Protection Level 설명

- **Level 1** – 가장 기본적인 암호화만 합니다. 실행 파일 전체를 암호화하며, 간단한 디버거 체크와 무결성 검사도 합니다. 그러나 복호화 루틴 자체가 단순하여 큰 효과를 보기는 힘듭니다. 하지만 암호화를 전혀 하지 않는 것 보다는 훨씬 낫습니다. 암호화되고 나면 실행파일의 크기가 약 4KB 정도 늘어납니다. Level 1로 암호화하면 윈도우 2003에서는 실행되지 않게 됩니다.

- **Level 2** – 실행 파일을 암호화하며, 압축까지 합니다. 그리고 상당히 복잡한 Anti-debugging 코드가 포함되므로, 분석하기가 쉽지 않습니다. 실행 후에 복호화된 코드를 덤프하는 것을 막는 코드도 포함되어 있지만, 큰 효과는 없습니다. 200KB 미만의 파일은 용량이 조금 늘어날 수 있으나, 게임 클라이언트처럼 용량이 1MB 이상 되는 파일은 암호화 후에 용량이 많이 줄어듭니다.

- **Level 2.5** – Level 2로 암호화한 실행 파일은 특정 복호화 툴로 쉽게 unpack되는 문제가 있었습

니다. Level 2.5는 Level 2와 동일하나 그 후에 Level 1로 한 차례 더 암호화 하여 unpack이 더 어렵습니다. 역시 이 경우에도 윈도우 2003에서는 실행되지 않게 됩니다.

- **Level 3** – 암호화하고 압축하는 것은 Level 2와 거의 같으나, 실제 복호화하는 방법은 차이가 납니다. 훨씬 복잡한 복호화 루틴을 가지고 있으므로, 분석하기가 힘듭니다. SoftIce와 같은 커널 디버거를 체크하는 기능이 강화되었습니다. Level 2 방식보다 용량이 많이 증가됩니다. 그러나 1MB 이상의 파일은 암호화 후 용량이 줄어듭니다. Level 3으로 패킹한 후에 실행하면 똑 같은 프로세스 2개가 실행되어서 하나가 다른 하나를 디버거 형태로 잡은 채 실행됩니다. 따라서 일반 디버거로 Attach하기가 까다롭게 됩니다. 하지만 이 경우에는 게임가드의 디버깅 검출 기능을 OFF 시켜야만 컴퓨터가 리부팅되는 것을 방지할 수 있습니다. Level 3를 테스트하시려면 게임가드 담당자에게 문의 바랍니다.

- **Level 4** – 가장 강력한 보호 기능을 제공합니다. 다른 암호화 방법과는 달리 실행될 때 전체 코드를 복호화하지 않고, 페이지 단위(4KBytes)로 복호화합니다. 즉, 실행에 필요할 때에만 그 페이지를 복호화하는 방식입니다. 이 방식은 해킹툴이 메모리 패치를 하는 것을 매우 어렵게 합니다. 그러나 게임 클라이언트에 스스로 코드를 변경(Self-modify)하는 루틴이 있으면 문제가 발생할 소지가 높습니다. 또, 실행 속도가 많이 느려질 수 있으므로 테스트를 충분해 볼 필요가 있습니다. 그리고 이 옵션일 때는, 게임가드에서 클라이언트 코드의 변조 유무를 체크할 수 없으므로 다른 부분에서 보안 홀이 있을 수 있습니다. 역시 Level 3처럼 게임가드의 옵션 수정이 필요하므로 문의 바랍니다.

- **Level 5** – Level 4 보다는 안정적이면서도 unpack이 힘들며 코드 및 파일 변조 유무 체크가 가능합니다. 단 이 옵션으로 암호화할 경우 다음 과정을 정확히 거쳐야 암호화가 완료됩니다. Encrypt 버튼을 눌러 파일 전송/암호화/수신이 완료되면 NPGE는 게임이 실행되길 기다리게 되는데, 이 때 방금 일부 암호화가 된 이 실행 파일로 게임을 실행시킵니다. (.bak 파일이 같은 폴더에 존재해야만 함) 그러면 게임가드는 실행 시점의 정상적인 Checksum을 클라이언트에 기록하게 됩니다. GAMEHACK_DOUBT 메시지가 발생하는 것은 정상이므로 그냥 종료합니다. DLL을 암호화할 경우는 GAMEHACK_DOUBT 메시지가 EXE에 비해 상대적으로 늦게 출력될 수 있습니다. 따라서 이 메시지가 발생할 때까지 기다려야 합니다. 게임을 종료하면 NPGE에서 암호화가 완료되었다는 메시지 박스("Encryption Done!")가 뜰 것입니다. 이제 이 실행 파일로 게임이 잘 실행되는지 테스트하면 됩니다.

- **Level 5**를 사용할 경우에는 게임가드 라이브러리(NPGameLib)에서 포함된 USER_POLYBUFFER 매크로를 사용할 수 있습니다. 게임 내의 중요한 함수들 맨 위에 USER_POLYBUFFER 매크로를 넣어주면 NPGE 패킹시 그 함수 전체를 특별히 처리하게 됩니다. 즉 게임 실행 후에도 메모리에 풀리지 않으며 오직 그 함수가 실행될 때에만 다른 메모리 영역에 일시적으로 풀려서 복잡한 Code obfuscation과 함께 실행됩니다. 따라서 unpack은 물론 그 함수의 분석 및 조작이 극히 힘들어집니다. 하지만 너무 자주 호출되는 함수에 사용하면 퍼포먼스가 떨어질 수 있습니다. 또한 이 함수 내에는 switch/case 문과 break문이 없어야 합니다. 그러한 조건이 맞지 않으면 매크로

를 사용하더라도 실제로는 특별한 처리가 되지 않을 수도 있습니다. 자세한 점은 게임보안센터로 문의 바랍니다.

- **Level 6** – 다른 Level들과는 달리 프로텍터(Themida)를 직접 구매하여 패킹하셔야 합니다. NPGE를 하는 이유는 실시간 변조 체크를 가능하게 하기 위함입니다. 사용법은 Level 5와 동일합니다. Themida로 패킹한 후에 그 파일을 NPGE에 넣고 Encrypt 버튼을 클릭한 뒤, 게임을 한번 실행시켜주시면 됩니다.

※ Level 6 사용 시 주의사항

Level 6로 패킹은 다른 레벨의 패킹과 달리 Themida를 사용하므로 주의를 하셔야 합니다. Themida가 다른 프로텍터들에 비해 보안성이 높은 반면, NPGE와의 호환 및 안정성 부분에서는 다른 레벨에 비해 다소 떨어질 수 있습니다.

또한, 더미다 적용 시 옵션 중 아래 부분의 항목을 해제 하신 뒤 테스트를 진행 해주시길 부탁 드리겠습니다.

Protetion Options -> Monitor Blokers -> Files MonitorsProtections -> Monitor Blokers -> Registry Monitors

Level 6로 패킹은 Level 5와 동일합니다만 다음에 알려드리는 과정을 다시 한 번 살펴봐 주십시오. **과정에 사소한 부분이 틀려도 정상적인 NPGE 패킹이 이루어지지 않는 경우가 발생할 수 있습니다.** 애초 더미다로 패킹된 실행파일(즉, 더미다 패킹된 상태)을 가지고 Level 6로 패킹을 시작합니다.

1. NPGE 실행 후 로그인
2. Level 6 선택 및 NPGE 적용할 파일을 불러오신 후 Encrypt 버튼을 누름
   - **Themida 적용이 된 실행파일(.exe)과 Themida 적용 전 실행파일 원본(.bak) 둘 다 필요하며, 파일명은 동일 폴더 상에 존재해야 합니다.**
   - 게임 실행파일이 Client.exe라면, Themida 패킹시 원본 파일이 Client.bak로 남습니다. **만약 같은 폴더에 Client.exe.bak도 있다면 NPGE가 실패할 수 있으므로 미리 삭제 바랍니다.**
3. NPGE 화면에 Status - Waiting for game executing 으로 바뀌면
4. 게임 실행 후 게임 접속 후에 "게임변조메세지"가 나오며 확인을 하신 후 정상 게임 종료해서 나오시면 Encrytion Done 메세지가 출력됩니다.
   - Encryption Done 메세지가 출력되면 정상적으로 NPGE 적용이 완료된 것입니다.
   - Encryption Done 메시지가 출력되지 않았다면 정상적으로 NPGE 적용이 완료되지 않은 것입니다. 이 경우 잘못된 .bak 파일을 참조하여 NPGE 패킹이 이루어진 것으로 볼 수 있습니다. NPGE는 .exe 파일과 .bak 파일이 동일한 TimeStamp를 갖고 있지 않으면 Encryption Done 메시지를 띄우지 않도록 설계되어 있습니다.
5. NPGE 패킹까지 적용된 게임실행파일로 게임가드에서 변조 메세지가 뜨지 않고 정상적으로 실행되는지 확인

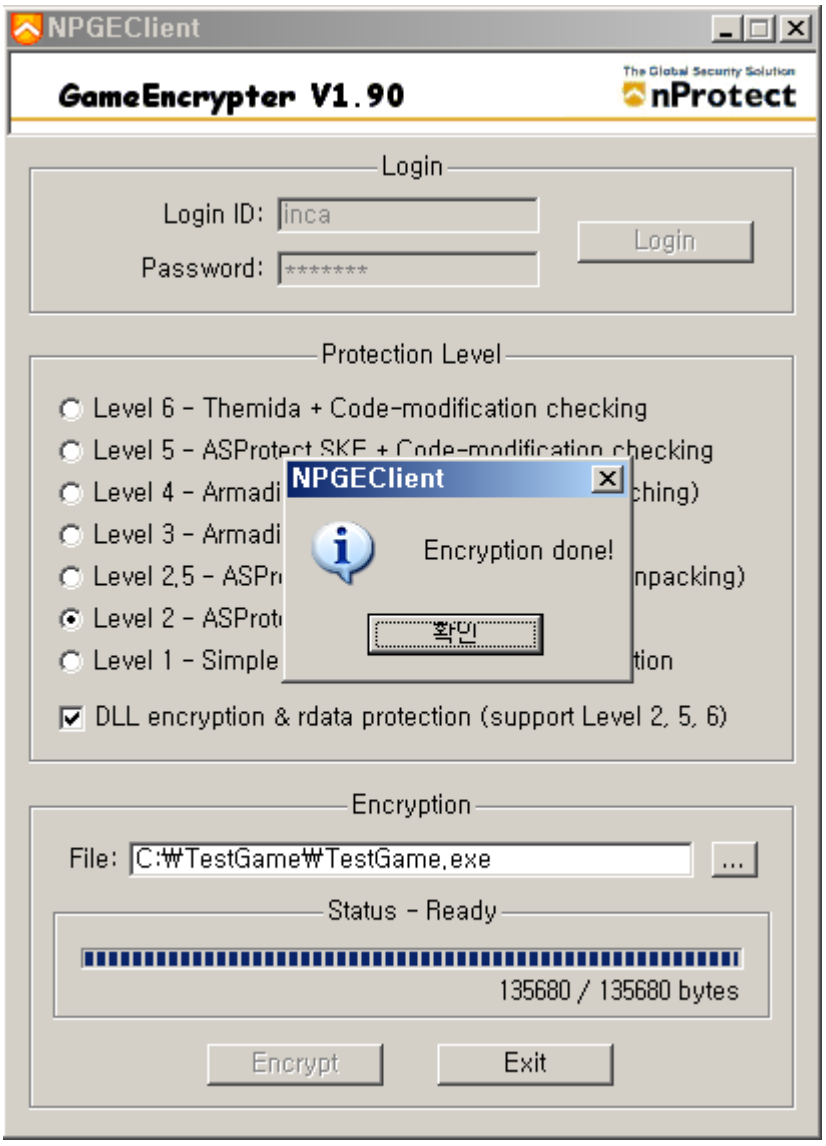
7. DLL encryption & rdata protection

- DLL을 암호화 하거나, EXE/DLL 파일의 rdata 영역 변조 체크를 이용하려면 이 옵션을 켭니다. 이 옵션은 Level 2, 5, 6만 지원합니다. Level 5, 6은 이 기능이 디폴트로 내장됩니다. 아래의 사항들에 유의하시기 바랍니다.

1) Level 2는 unpacker가 존재합니다. 그러나 이 옵션을 사용하면 unpacking 여부를 알아낼 수 있으므로 Level 2를 사용할 때는 EXE든 DLL이든 항상 이 옵션과 함께 사용하는 것을 권장합니다. (L2+)
2) 이 옵션을 사용하면 파일 전송/수신이 완료된 후에 게임을 1회 실행시켜야만 암호화가 완료됩니다. 위에 Level 5의 설명과 동일한 과정을 따르면 됩니다.
3) DLL일 경우 게임가드 초기화 전에 Load 되어 있어야 합니다. 또한 게임 도중에 Free 되어서도 안 됩니다. 만약 게임가드에 의한 변조 체크를 하지 않고, 단지 Encryption만 할 것이라면 Load/Free 타이밍은 무관합니다.
4) DLL의 경우 DLL 특성과 Protection Level 특성에 따라 DLL을 relocation 가능하게 할지 fixed로 할지 테스트 후 결정해야 합니다. Fixed로 생성하는 법은 아래 예제에 나와 있습니다. linker 옵션을 주지 않으면 일반적으로 DLL은 relocation 가능하게 생성됩니다. Fixed일 경우 Load 될 base 번지를 적절히 지정해야 합니다. 가능하다면 Fixed를 권장합니다.

   ```
   // fixed로 생성하기 위한 linker 옵션
   #pragma comment(linker, "/base:0x41000000 /fixed" )
   ```

5) DLL Encryption이 완료된 DLL을 게임에 패치한 후, 실시간 변조 체크를 원하면 그 DLL 이름을 잉카에 통보합니다. 게임가드가 업데이트되면 그 때부터 DLL 변조 체크가 가능하게 됩니다.



암호화가 완료되었다는 메시지 발생 화면

8. 유의 사항

- 암호화 후의 실행 파일은 사이즈가 변하므로 .bak 파일(원본 파일)과 비교, 확인해야 합니다.
- 암호화된 실행 파일은 반드시 다양한 운영체제에서 실행 여부를 확인해 보아야 합니다.
- 만약 문제가 발생된다면 Protection Level 을 바꾸어서 암호화한 후, 다시 테스트 해보아야 합니다. NPGE Tool 의 경우 직접 패킹을 수행하는 것이 아니라 패커를 선택할 수 있게 하는 유틸입니다. 패커의 호환성이 낮으면 개발코드의 작성 문법에 따라 호환되지 않는 경우가 생깁니다. 소스의 수정이 많은 작업의 경우 중간 중간 패킹을 시도하여, 호환성 여부를 수시로 검사하셔야 합니다.
- 로그인이나 암호화가 정상적으로 되지 않으면 ㈜잉카인터넷 게임보안사업본부로 문의해주시기 바랍니다.

■ (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.

MEMO