

.NET 코드에 대한 보안 대책

소프트웨어 복제 방지 대책은 어플리케이션 소프트웨어 개발자/벤더들이 매출 증대 및 이익 창출측면에서 가장 중요하게 생각하는 하나의 이슈입니다. 그런데, 마이크로소프트사가 얼마 전에 발표한 .NET 개발환경은, 어플리케이션 소프트웨어 개발자/벤더들한테 소프트웨어 불법 복제 방지 솔루션을 개발.판매하는 벤더들로 하여금 대단히 골치아픈 문제에 봉착하도록 만들었습니다.

.NET 개발환경하에서 생성된 코드들은 소스 레벨이건 컴파일된 코드이건 리버스 엔지니어링이 용이합니다. .NET 컴파일러는 소스코드를 본질적으로 데이터 레벨인 MSIL 코드로 컴파일하며, 그 컴파일된 코드는 다큐먼트가 잘된 마이크로프로세서에 의해 실행되는 것이 아니고 다큐먼트가 잘 안된 마이크로소프트 프로그램에 의해 실행이 됩니다. 또한, 그 컴파일된 .NET 코드는 쉽게 디컴파일이 되며 그 디컴파일된 코드는 원천 소스코드처럼 이해하기가 용이합니다.

한편, 복제 방지 기술로는 대표적으로 2가지가 있습니다. 하나는 코드를 해석하기 어렵게 만듦으로서 리버스 엔지니어링을 방지하는 Obfuscation 기술과 다른 하나는 사람에게나 디컴파일러거나 완벽히 해독이 불가능 하도록 함으로서 리버스 엔지니어링을 방지하는 Encryption 기술이 있습니다. 문제는 기존의 Obfuscation 기술은 .NET 코드상에 적용하는데 아무런 문제가 없지만 기존의 암호화 기술은 .NET 코드상에 적용하기가 매우 어렵다는 사실입니다.

Obfuscation 기술은 주로 주석들의 삭제, 정수들의 엔코딩, 소스내 identifiers의 개명등을 통해 사람에게는 유용하지만 컴퓨터에게는 소용없는 정보들을 삭제하는 방식으로, 이는 다음과 같이 3가지 문제를 안고있습니다. 첫째, 디컴파일된 기계언어를 해독하는 수준에 도달한 전문 해커들한테는 하나의 해쳐나갈 장벽일뿐이며, 둘째, 이 방식은 제 3자 소프트웨어에 의해 쉽게 쿨이 가능한 취약점을 갖고 있으며, 셋째, 프로그램이 실행되는 동안 디버거의 동작을 방지하지 못하며, 파일내 코드들 및 메모리내 코드들의 시험을 방지하지 못합니다.

Encryption 기술은 통상 컴파일러들에 의해 생성된 오브젝트 코드상에 적용을 하며, 하나의 파일내 코드의 암호화 또는 컴퓨터내 동작하는 과정중의 코드 또한 암호화가 가능합니다. 이는 전문 해커들로 하여금 프로그램 파일뿐만 아니라 그 프로그램이 컴퓨터상에서 실행되는 동안 시스템 메모리내 상주하는 코드들조차 리버스 엔지니어링을 할 수 없도록 해 줍니다.

결론적으로, .NET 환경으로 제작한 어플리케이션 소프트웨어들에 대한 불법복제 방지툴의 채택시 반드시 Obfuscation 기술이 아닌 암호화 기술을 제공하는 툴을 선택.구입할 것에 유의하시기 바라며, CrypKey야 말로 .NET 환경하에서 암호화 기술을 제공하는 전세계에서 몇 안되는 솔루션/툴중 하나임을 공언합니다.