소프트웨어 생명주기상에서의 공통평가기준기반 보안보증 적용 프로세스에 관한 연구

신호준*, 김행곤*, 김태훈**, 노병규**
*대구가톨릭대학교 컴퓨터정보통신공학부, **한국정보보호진흥원 평가인증사업단

A Study on the Process for Applying Security Assurance based CC on Software Lifecycle

Ho-Jun Shin*, Haeng-Kon Kim*, Tai-Hoon Kim**, Byong-Kyu No**
*School of Computer Information & Communication, Catholic University of Daegu,
**IT Security Evaluation & Certification Authority, Korea Information Security Agency

요 약

최근 정보보호에 대한 관심이 높아짐에 따라 그에 따른 기반 기술들이 요구되고 있다. 특히, 통신 인프라에 집중되었던 정보보호 산업이 이를 기반한 제품으로 관심이 부각되고 있다. 이러한 정보보호제품의 신뢰성은 매우 중요한 요소이며, 신뢰성 보장을 위한 보안 기능의 보증은 중요하다.

본 논문에서는 개발 단계에서 유지보수 단계에 이르는 생명주기 활동의 보증과 품질보증 위한 방법 또한 중요하다는 것을 인식하고 이를 위해 소프트웨어 개발에 정보보호시스템 공통평가기준(정보통신부 고시 제2002-40호, 이하 공통평가기준)을 적용하여 개발할 수 있도록 프로세스를 제시한다. 이를 통해 소프트웨어 개발자나 시스템 관리자들이 정보보호 인증을 보장하며, 안전한 소프트웨어를 개발하여 효율적으로 관리할 수 있도록 소프트웨어 개발 및 변경시 발생할 수 있는 위험들과 이에 대한 통제들을 제안한다. 향후 전산망 시스템에서 사용되는 정보보호 제품의 개발 및 관리에 도움 줄 것을 기대한다.

1. 서론

정보화 사회가 확산됨에 따라 개인이나 조직의 주요정보들이 컴퓨터 시스템을 통하여 처리, 저장, 관리되고 있다. 중요한 정보들이 정확하게 처리되고 안전하게 관리되기 위해서는 시스템에서 사용되는 프로그램 즉, 소프트웨어가 이러한 요구조건을 수용할 수 있도록 개발되어야 한다. 새로운 환경변화에 따라 사용자 요구사항이 더욱 다양해지고, 컴퓨터 시스템의 사용 편리성이 강조됨에 따라 새로운 기능을 갖춘 소프트웨어의 개발 경쟁이 치열해지고 있다. 또한 프로그램의 개발을 지원해줄 수 있는 개발 도구의 성능이 계속 발전함에 따라 새로운 소프트웨어의 개발속도는 더욱 빨라지고 있으며, 기술적으로 더욱 복잡해지고 있다. 이러한 소프트웨어의 양적인 발전과 함께 질적인 발전 또한 중요하다. 비록 사용자 요구사항에 맞게 개발된 소프트웨어라 할지라도 허가되지 않은 제3자에 의해 정보의 비밀성과 무결성이 침해될 수 있다면 결코 우수한 소프트웨어라고 말할 수 없다[1].

사용자 요구사항에 맞게 정보를 정확히 처리할 수 있을 뿐만 아니라 안전하게 관리할 수 있는 소프트웨어의 개발을 위해서는 소프트웨어 개발 단계에서부터 처분 단계에 이르기까지 체계적인 관리와 통제가 요구된다.

본 논문에서는 소프트웨어 개발자나 시스템 관리자들이 안전한 소프트웨어를 개발하여 효율적으로 관리할 수 있도록 소프트웨어 개발 및 변경시 발생할 수 있는 위험들과 이에 대한 통제들을 제안한다. 향후 전산망 시스템에 사용되는 소프트웨어의 보안관리에 많은 도움을 줄 수 있을 것으로 기대한다.

2. 관련 연구

2.1 공통평가기준의 생명주기 지원 클래스

IT 시스템의 부분이나 공통평가기준에 기반하여 평가되어야 하는 생산품은 평가 목표(TOE : Target of Evaluation)라고 불리고 평가 권한에 의해 검증되는 다른 보안 요구사항을 수행해야 한다. 공통평가기준의 보안 요구사항은 보안 기능 요구사항(생산품상의 요구사항)과 보안 보증 요구사항(프로세스상의 요구사항)으로 분할되며, 클래스 내에 구조화된다. 기능적인 요구사항은 TOE의 보안 목표를 달성하기 위한 시스템의 기능에서 실체화되며, 보증 요구사항의 수와 엄격함에 따라 TOE를 위해 선택한 평가 보증 등급(EAL : Evaluation Assurance Level)에 의존하여 수행된다[2].

TOE에 대한 적절한 평가 보증등급을 부여하기 위해서는 제시된 요구사항의 고수준을 만족해야한다. 그림 1은 제시된 생명주기 지원 클래스에 대한 적절한 요구사항들의 클래스, 패밀리, 컴포넌트의 구조를 도식화하였다. 생명주기 지원 클래스는 결함 교정 절차 및 정책, 도구와 기법의 정확한 이용, 개발 환경을 보호하기 위해

사용되는 보안 대책 등을 포함한 TOE 개발의 모든 단

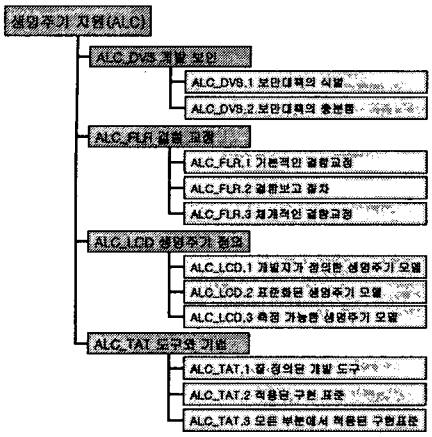

그림 1. 공통평가기준에서의 생명주기 지원 클래스의 구성

계에 대하여 잘 정의된 생명주기 모델을 채택함으로써 보증 요구사항을 정의한다. 또한, 개발 및 유지하는 동안 TOE의 상세화 과정에서 규칙 및 통제를 수립한다. 보안 분석 및 증거 생성이 개발 과정과 운영지원 활동의 한 부분으로서 정기적으로 수행될 때, TOE와 TOE 보안 요구사항 간의 일치성에 대한 신뢰는 증가한다.

2.2 표준 소프트웨어 생명주기 프로세스

ISO12207은 소프트웨어 생명주기 프로세스 표준이다. 기존의 방법론이나 표준과는 달리 소프트웨어 개발주기에 대한 기본공정이외에도 지원공정과 조직공정을 추가함으로써 정보시스템 전체를 어떻게 하면 포괄적으로 적용할 수 있을 지에 대한 전체적인 프레임워크를 제공한다[3].

그림 2는 ISO12207의 전체 구성도이며, 기본공정, 지원동정, 조직공정으로 구분되는데 기존의 소프트웨어 품질 및 개발주기 관련 표준과 달리 조직공정에 기반구조 공정을 포함함으로써 정보시스템 아키텍처가 정보시스템을 구축하는데 반드시 다루어야할 필수요소임을 제시하는 최초의 국제표준이다. 소프트웨어를 포함한 시스템, 단독형 소프트웨어 및 서비스의 획득 동안에, 그리고 소프트웨어 제품의 공급, 개발, 운영 및 유지보수 동안에 적용될 수 있는 공정(process), 활동(activity) 및 세부업무(task)를 포함한다.

3. 소프트웨어 생명주기상에서의 공통평가기준기반 보안보증 적용 프로세스

대부분의 소프트웨어는 소프트웨어 개발 생명주기(Software Development Life Cycle : SDLC)를 이용하여 개발된다. 일반적으로 소프트웨어 개발 생명주기는 시작, 분석, 설계, 구현, 운영과 같이 다섯 단계로 구성되어 있다. 소프트웨어 보안 계획은 소프트웨어 개발 생명주기 전단계에서 고려되어야 하나, 가장 효과적인 방법은 시작 단계에서 전체적인 보안 계획을 수립하는 것이다.

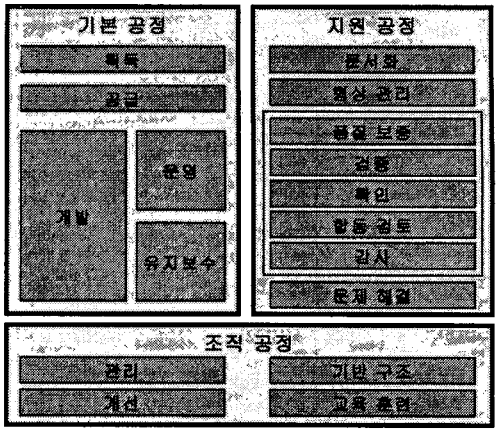

그림 2. ISO/IEC 12207 국제표준의 구성도

일반적으로 소프트웨어 개발이 완료된 후에 새로운 기능을 추가하는데 소요되는 비용은 시작 단계에서 소요되는 비용의 10배 이상이 요구된다. 따라서 소프트웨어 시작 단계에서부터 적절한 보안 계획을 수립하는 것이 비용 효과적인 방법이다. 소프트웨어나 정보보호 제품의 개발 생명주기상에서 이루어지는 보안 활동에서 공통평가기준의 내용을 기반으로 그림 3에 나타내었다. 정보보호제품의 개발을 위한 단계뿐만 아니라 관리 단계에서 요구되는 공통평가기준의 보안보증 요구사항들을 고려하였다.

시스템 주변환경이나 기술 등이 계속적으로 변화함에 따라 새로운 보안 기능의 추가가 필요할 수 있다. 따라서 소프트웨어 생명주기의 각 단계마다 보안 계획을 검토하여 필요에 따라 적절히 수정하여야 한다. 대부분의 경우, 환경이나 기술 변화에 맞게 보안 계획을 수정함으로서 더 큰 이득을 얻을 수 있다.

또한 보안과 관련된 모든 활동들을 문서화해 두어야 한다. 소프트웨어 개발 관리자는 문서화를 통해 소프트웨어 생명주기 전단계에 걸쳐 보안 기능이 적절히 구현되었는지 확인할 수 있으며, 미흡한 부분을 보완할 수 있다. 문서화는 소프트웨어 개발 관리자뿐만 아니라 외부 감사요원에게도 많은 도움을 줄 수 있다. 외부 감사요원은 소프트웨어 개발 문서를 바탕으로 개발 활동의 문제점과 보안이 미흡한 부분들을 확인할 수 있다.

3.1 시작 단계에서의 보안

소프트웨어 개발 시작 단계에서는 조직 수준에서의 위험분석, 조직 전체에 적용되는 보안 방침 수립, 전체적인 보안 계획 등이 작성된다.

조직 수준에서의 위험분석 : 시작 단계에서는 조직 수준에서의 위험분석이 이루어져야 한다. 조직 수준에서의 위험분석은 개발하고자 하는 소프트웨어나 기존 소프트웨어에 발생할 수 있는 위협을 식별하고 이에 대한 대응책을 수립하는 것이다. 먼저 조직의 시스템 환경에 적합한 위험관리 전략을 수립하여 위험을 분석하고 위험에 대한 보안 대책들을 시간 및 비용 효과적인 방법으로 수립하는 것이다.

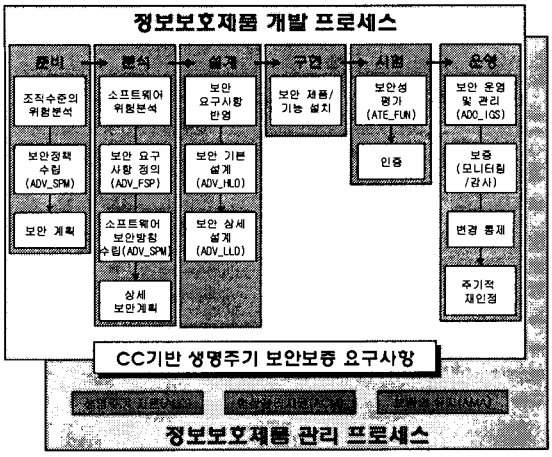

그림 2. 공통평가기준기반의 정보보호제품 개발 프로세스

보안 정책 수립 : 조직수준에서의 위험분석을 실시한 후에는 조직 전체에 공통적으로 적용될 수 있는 기본적인 보안 방침을 수립하여야 한다. ADV_SPM(Security Policy Model)의 보안 정책에 대한 구조적인 표현을 기반으로 기본적인 보안 방침을 보안 목적과 정보의 비밀성, 무결성, 가용성, 기록성, 신뢰성을 기초로 한 보안 요구사항, 조직의 하부구조와 책임, 소프트웨어 개발 및 구매시 보안 기능의 설치, 비상사태에 대한 대응책, 법적인 책임 등이 언급되어야 한다.

보안 계획 : 기본적인 보안 방침이 수립되면 보안 방침에 따라 적절한 보안 계획을 수립하여야 한다. 보안 계획은 보안 방침에 따라 취해져야 할 활동들을 정의하는 것이다. 보안 계획은 장기적으로 취해져야 될 활동과 단기적으로 취해져야 할 활동을 구분하여 수립되어야 하며, 소요되는 비용, 실행 일정 등이 정의되어야 한다.

3.2 분석 단계에서의 보안

소프트웨어 분석 단계에서는 소프트웨어 위험분석, 보안 요구사항 정의, 소프트웨어 보안 방침 수립, 상세 보안 계획 수립 등이 이루진다.

소프트웨어 위험분석 : 소프트웨어 위험분석은 모든 소프트웨어에 대해 상세한 위험분석을 실시하는 것이다. 소프트웨어 위험분석에는 자산과 자산의 가치, 위협, 취약성 등을 분석하고, 예상되는 위험에 대한 적절한 대응책을 수립하여야 한다. 또한 확인된 위험에 대한 적절한 대응책을 선택하여 관리자가 요구하는 수준까지 위험을 감소시키는 활동도 포함되어야 한다. 소프트웨어 위험분석에는 많은 비용과 자원 소모가 요구되므로 분석 범위를 신중히 고려하여야 하며, 관리자가 계속적으로 참여하여야 한다.

보안 요구사항 정의 : 사용자 요구사항 정의시 보안 요구사항도 함께 정의되어야 한다. ADV_FSP(Functional Specification)을 통해 TOE 보안기능 요구사항을 완전하고 정확한 예를 들어 설명해야한다. 일반적으로 상당히 많은 량의 보안 요구사항이 발생하므로 보안 요구사항을 기술적인 측면(예를 들면, 패스워드, 접근통제)과 관리적인 측면(예를 들면 교육과 훈련), 보증(예를 들면, 소프트웨어 개발자의 배경 검토) 등으로 분류하여 정의하는 것이 효과적이다.

소프트웨어 보안 방침 수립 : 소프트웨어 보안 방침은 소프트웨어와 제공되는 서비스, 정보 등의 보호를 위한 세부적인 규칙이며, 소프트웨어 보안 방침과 조직 전체의 보안 방침이 조화를 이루어야 한다. 소프트웨어 보안 방침 수립시에는 소프트웨어 개발 목적, 소프트웨어 의존도, 처리되는 정보의 가치, 위협, 취약성, 영향, 필요한 예산 등을 고려하여야 한다. 소프트웨어 보안 방침은 강제적인 사항이므로 고위 관리자의 승인을 받아 실행되어야 한다.

상세 보안 계획 : 상세 보안 계획은 개발 우선순위, 예산, 개발 일정 등을 고려하여 상세한 보안 계획을 수립하는 것이다. 상세 보안계획 수립시에는 우선순위에 따라 취해져야 할 활동을 정의하고, 자원과 책임의 할당, 개발 진행상태의 기록, 보안 교육 및 훈련 일정 수립 등을 고려하여야 한다.

3.3 설계 단계에서의 보안

소프트웨어 설계 단계에서는 보안 요구사항의 반영과 보안 제품의 구매여부에 대한 결정이 이루어진다.

보안 요구사항 반영 : 소프트웨어 설계 단계에서 정의된 보안 요구사항을 반영하여야 한다. 먼저 보안 요구사항들을 검토하여 설계 가능성 여부를 판단하여야 한다. 보안 요구사항은 보안 방침이나 표준 등에 적합하여야 한다. 소프트웨어 개발자는 보안 요구사항을 분석하여 설계에 적절히 반영하여야 한다.

보안 기본 설계 : ADV_HLD(High Level Design)에서 정의한 것처럼 TSF(TOE Security Function) 기능 명세를 TSF의 주요 구성부분으로 세분화한 최상의 수준의 설계 명세이다. 기본 설계는 기본 구조와 주요 하드웨어, 펌웨어, 소프트웨어 요소들을 식별한다.

보안 상세 설계 : ADV_LLD(Low Level Design)에서 정의한 내용을 기반으로 TSF의 각 모듈에 대한 목적, 기능 인터페이스, 종속관계, 수행기능의 구현을 설명하고 명세한다.

3.4 구현 단계에서의 보안

소프트웨어 구현 단계는 구매한 보안 제품을 설치하거나 자체 개발한 보안소프트웨어를 프로그래밍하는 단계이다. 구현 단계에서는 소프트웨어에 영향을 미치는 모든 기술적, 비기술적 보안 기능들의 타당성을 검토하고 잠정적인 위험들을 분석하여야 한다. 보안 기능의 타당성 검토는 소프트웨어 시험 및 평가 절차의 일부로서 매우 중요하다. 보안 기능이 부적합할 경우, 안전한 소프트웨어 구현이 불가능하므로 구현 단계에서는 보안 기능의 타당성 검토가 우선적으로 이루어져야 하며, 또한 잠정적인 위험에 대한 분석이 실시되어야 한다.

일반적으로 보안 기능의 구현은 보안관리자의 책임이

다. 보안 기능 구현시 보안관리자는 보안 대책에 소요되는 비용과 보안 요구사항의 수용 여부, 보안 기능의 정확한 실행 여부 등을 고려하여야 한다.

3.5 시험 단계에서의 보안

시험 단계에서는 보안성 평가와 인증 두 가지 보안 활동이 이루어진다.

보안성 평가 : 시험 단계는 소프트웨어 개발 생명주기에서 매우 중요한 단계이다. 개발된 소프트웨어는 운영 단계로 이동하기 전에 반드시 시험 단계를 거쳐야 한다. 시험 단계에서는 보안 기능의 평가가 ATE_FUN (Functional Tests)에 요구사항에 따라 이루어져야 한다. 보안성 평가는 보안 기능이 보안 요구사항을 어느 정도 만족시키는지 기술적으로 평가하는 것이다. 보안성 평가는 소프트웨어 개발 단계에 참여하지 않은 관리자와 기술요원에 의해 실행되어야 한다. 보안성 평가가 완료되면 보안 관리자는 평가결과를 검토하여 보안 기능의 적절성 여부를 판단하여야 한다.

인증 : 보안성 평가가 완료되면 평가 작업에 참여하지 않은 관리자와 기술요원에 의해 보안성 평가에 대한 인증이 이루어져야 한다. 인증은 보안성 평가가 평가 절차에 따라 적절히 계획되고 실행되었는지, 평가 결과가 신뢰할 수 있는 것인지 검증하는 것이다. 보안성 평가 결과와 인증은 고위 관리자가 이에 대해 승인을 하므로서 효력을 발생하게 된다. 보안성 평가 결과와 인증은 모두 문서화하여야 한다.

3.6 운영 단계에서의 보안

소프트웨어 운영 단계에서는 많은 보안 활동들이 이루어 진다. 일반적으로 보안 소프트웨어의 운영 및 관리, 보증(모니터링, 감사 등), 소프트웨어 변경 통제, 소프트웨어의 주기적인 재인증 등이 운영 단계에서 이루어진다.

보안 소프트웨어 운영 및 관리 : 보안 소프트웨어의 운영 및 관리는 자체 개발하였거나 구매한 보안 기능을 실직적으로 운영하고 유지보수하는 것이다. 보안 제품 설치, 백업, 사용자 보안 교육, 암호화 키관리, 사용자 접근 권한 통제, 보안 소프트웨어이 업그레이드 등 전반적인 보안 활동들이 포함된다. 또한, ADO_IGS(Installation, Generation and Start-up)에서 제공되는 설치, 생성, 시동 요구사항 및 절차를 구성, 작동되도록 해야한다.

보증(모니터링, 감사) : 보증은 소프트웨어나 정보를 보호하기 위해 보안 기능들이 어느 정도 적절히 수행되는 지에 대한 신뢰도이다. 운영 단계에서는 일반적으로 보안에 대한 사용자나 관리자의 인식 부족으로 보안 침해 가능성이 높다. 따라서 보안 기능의 적절한 실행을 확인할 수 있어야 한다.

보안 기능을 보증하기 위한 방법으로는 일반적으로 모니터링과 감사 두 가지 방법이 사용되고 있다. 모니터링은 시스템과 주변환경, 시스템 사용자의 활동 등을 계속적으로 관찰하는 것이며, 감사는 보안성의 재평가를 위해 주기적으로 실시되거나, 보안 침해 사고가 발생하였을 경우, 이를 조사하기 위해 실시된다. 감사는 시스템 전체에 대해 광범위하게 실시된다.

소프트웨어 변경 통제 : 컴퓨터 시스템과 시스템 주변 환경은 계속적으로 변화한다. 사용자의 불만사항, 새로운 기능 및 서비스 요구, 새로운 위협이나 취약성 등에 대응하여 시스템 관리자나 사용자는 소프트웨어에 새로운 기능이나 절차를 추가하거나, 기존 소프트웨어를 갱신하여야 한다.

주기적인 재인증 : 컴퓨터 시스템과 주변환경이 계속적으로 변화함에 따라 보안 기능의 적절성도 주기적으로 평가되어야 한다. 보안 기능의 적절성과 변경의 필요성을 재인증하므로서 소프트웨어 변경으로 인해 발생할 수 있는 위험이나 취약성을 방지할 수 있다.

5. 결론 및 향후연구

정보통신 제품이나 시스템을 개발할 경우 보안에 대한 평가를 위해서 표준화된 요구사항들의 목록으로 공통평가기준이 정의되어 있다. 공통평가기준을 사용하여, 시스템 자체와 시스템 개발에 많은 보안 요구사항을 정의 가능하다. 사용자 요구사항에 맞게 정보를 정확히 처리할 수 있을 뿐만 아니라 안전하게 관리할 수 있는 소프트웨어의 개발을 위해 전체 생명주기에 보안의 체계적인 관리와 통제가 요구된다.

본 논문에서는 소프트웨어공학 생명주기동안의 프로세스에서 보안측면을 고려하여, 공통평가기준에서의 행위와 문서 등의 자원과 생명주기 지원을 위한 클래스의 적용 및 개발에 보안 요소를 적용시켰다. 이를 통해 소프트웨어 개발자나 시스템 관리자들이 안전한 소프트웨어를 개발하여 효율적으로 관리할 수 있도록 소프트웨어 개발 및 변경시 발생할 수 있는 위험들과 이에 대한 통제가 가능하다. 또한, 시스템 개발로 밀접하게 연관되어 시스템의 신뢰성을 증가시킬 수 있다. 향후 공통평가기준 기반의 정보보호제품의 개발과 평가를 위한 세부적인 프로세스와 이를 통한 사례연구가 수반되어야 하며, 이를 지원 가능한 도구에 관한 연구가 요구된다.

[참고문헌]

1. 김세헌, 정보보호 관리 및 정책, 생능출판사, 2002.
2. Common Criteria Project/ISO, "Common Criteria for Information Technology Security Evaluation Version 2.1 (ISO/IEC 15408)," http://www.commoncriteria.org/ CC /, 1999.
3. "Information Technology-Software Life cycle Process, (ISO/IEC 12207)," http://standards.ieee.org/reading/ieee/std/, 1998.
4. Boehm, B. W., Software Engineering Economics, Prentice-Hall, NJ, 1988.
5. Ruben Prieto-Diaz, "The Common Criteria Evaluation Process," Commonwealth Information Security Center Technical Report, 2002.
6. "정보보호시스템 공통평가기준," 정보통신부 한국정보보호진흥원, 2002.