Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.6, Dec. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.6.1541

BSIMM을 활용한 정보보호시스템 보안 설계 방안

박 정 섭†‡
한국인터넷진흥원

Security Design for Information protection System using BSIMM

Jung-Sup Park†‡
Korea Internet & Security Agency

요 약

최근 IT 산업에서 보안은 소프트웨어 개발 시 가장 중요하게 고려해야 할 요소로 자리 잡았다. 특히 자산을 보호하는 목적의 정보보호시스템의 보안성은 더욱 중요하다. 정보보호시스템의 보안성 평가시 공통평가기준(Common Criteria: CC)에서는 보안 구조(ADV_ARC) 패밀리 요구사항을 제공하여 평가대상의 보안성을 보장하도록 한다. 하지만, 체계적인 소프트웨어 보안 구조 설계 프로세스 없이 이러한 보증 요구사항을 일관되게 만족시키는 것이 다소 어려운 것이 현실이다. 본 논문에서는 BSIMM의 보안 프레임워크를 활용한 정보보호시스템의 보안 설계 방안을 제안한다.

ABSTRACT

In the recent IT industry, security has established itself as the factor to be considered the most in the software development. It goes without saying that security is the critical factor for the development of information security products. In the evaluation of the information security products, the security is assured by the security architecture requirement (ADV_ARC). However, the Absence of the systematic software security architecture process makes it difficult to guarantee the security quality consistently even though they are evaluated based on common criteria. In this paper, we propose a way to ensure a consistent security quality applying the software security framework in BSIMM.

Keywords: BSIMM, SSDL, Common Criteria

I. 서 론

안전한 소프트웨어를 개발하기 위해 요구사항, 설계, 구현, 시험 단계 등 소프트웨어 개발 생명주기(Software Development Life Cycle) 전반에 걸쳐 많은 노력이 수반되고 있다. 공통평가기준에서도 평가대상의 보안성을 보증하기 위해 보안 구조(ADV_ARC, Security Architecture) 요구사항을 제시하고 있다. 이는 소프트웨어 설계 시 자체보호, 영역분리, 우회불가성 등을 고려하여 소프트웨어의 보안 구조가 안전하게 설계되었다는 것을 보증하기 위함이다[3].

공통평가기준에서는 소프트웨어의 보안 구조를 보증하기 위한 요구사항으로 각 시스템의 보안구조를 보증하지만, 이러한 요구사항을 만족시키기 위한 보안 설계 프로세스는 보증하지 않고 있어, 소프트웨어의 보안구조 설계가 일회성으로 그칠 수도 있다. 일관된 보안설계의 품질 확보를 위해서는 소프트웨어 개발 시 안전한 보안구조를 설계하기 위한 보안 요구사항 뿐만 아니라 체계적인 보안설계 프로세스가 요구된다. 이에 본 논문에서는 소프트웨어 개발 시 일관된 보안성을 확보하기 위해 공통평가기준과

Received(05 .19. 2015), Modified(1st: 09. 04. 2015 2nd: 10. 19. 2015), Accepted(10. 19. 2015)
† 주저자, pjs@kisa.or.kr
‡ 교신저자, pjs@kisa.or.kr(Corresponding author)

BSIMM을 비교, 분석하여 체계적인 보안구조 설계 방안을 제시하고자 한다.

II. 관련 연구

2.1 성숙도 모델

성숙도 모델은 특정 목적을 위한 프로세스 내 구성된 요소의 능력, 절차 및 최적화 등의 수준을 산정하여 더 높은 수준으로 개선할 수 있도록 가이드 한다. 성숙도 모델은 CMM(Capability Maturity Model), CMMI(Capability Maturity Model Integration) 및 BSIMM(Build Security In Maturity Model) 등이 있다[1].

BSIMM은 소프트웨어 보안 계획을 구축 및 개선하기 위한 가이드이다. BSIMM은 주요 소프트웨어 보안 계획의 공통 활동을 문서화하여 위험 관리, 소스코드의 질 향상, 비용 감소 및 보안성을 명확히 하는 것을 목적으로 한다. BSIMM의 SSF(The Software Security Framework)는 4개의 도메인으로 구성되어 있으며, 각 도메인 당 3개의 Practice가 존재한다. Practice는 Level 1~3의 Objective와 Activity로 구성되어 있으며, 소프트웨어의 최종 성숙도는 각 Practice의 Level을 산정하여 도출된다.

2.2 정보보호시스템 공통평가기준

정보보호시스템 공통평가기준(Common Criteria for Information Technology Security Evaluation)은 정보보호시스템의 보안성을 보증하기 위한 공통 요구사항을 제시하여 표준화된 보안성 평가를 수행할 수 있도록 한다. 또한, 정보보호시스템의 개발, 평가, 배포 등에 대한 가이드로 활용될 수 있다. 공통평가기준은 총 3부로 구성되어 있는데, 1부는 소개 및 일반모델, 2부는 보안기능요구사항, 3부는 보증요구사항을 설명하고 있다. CEM(Common Methodology for Information Technology Security Evaluation)은 공통평가기준에 정의된 기준과 평가 증거를 기반으로 평가자가 수행해야 하는 최소 평가 행동을 정의하고 있다[2][3].

정보보호시스템 공통평가기준은 보안 문제(위협, 정책, 취약성, 위험도, 운영환경에 대한 가정 사항)를 해결하기 위해 보안 문제에 대한 보안목적(정보보호시스템의 보안목적, 운영환경에 대한 보안목적)을 정의한다. 또한, 보안목적을 이루기 위해 정보보호시스템에 보안기능 요구사항(SFR: Security Function Requirements)이 구현되었으며, 보안보증 요구사항(SAR: Security Assurance Requirements)을 만족하여 개발되었는지 검증한다[2].

2.3 CWE, CVE, CAPEC

CWE (Common Weakness Enumeration)는 코드, 설계, 시스템 아키텍처에서 발견되는 소프트웨어 보안 취약성의 원인을 논의 및 관리하기 위한 공통 언어이다[4]. CVE (Common Vulnerabilities and Exposures)는 공개적으로 알려진 보안 취약성과 노출을 공유하기 위한 사전이다. 취약성과 노출은 보안정책을 위반하거나 공격을 허용한 것을 말한다[5]. CAPEC(Common Attack Pattern Enumeration and Classification)은 보안 시스템과 서비스의 약점을 악용한 예를 분석한 공격 패턴을 말한다[6]. CWE는 CVE의 특성 및 속성이 될 수 있으며, CAPEC은 CWE의 예시와 관련된 공격 패턴이 될 수 있다. CWE-CVE-CAPEC은 공통평가기준에 활용하여 정보보호시스템의 체계적인 취약성 분석할 수 있다. 또한, 정보보호시스템의 보안성 품질 측정 및 관리를 위한 효용성 있는 취약성 분석 절차로 활용할 수 있다[7]

III. BSIMM과 정보보호시스템 공통평가기준

BSIMM의 평가 대상은 소프트웨어 개발 정책과 프로세스이다 물론 정보보호시스템 공통평가기준에서도 개발 프로세스 및 다양한 절차들을 평가항목으로 포함하고 있지만 대부분의 평가요구사항은 정보보호시스템 집중되어 있다. BSIMM과 정보보호시스템 공통평가기준은 평가 항목이 매우 유사하지만, BSIMM이 보다 체계적인 소프트웨어 보안 정책 및 거버넌스를 제공한다. 본 장에서는 정보보호시스템의 일관된 보안 품질을 유지하기 위해 BSIMM SSF의 각 Practice와 정보보호시스템 공통평가기준의 3부 보증요구사항을 비교 및 분석하고 공통평가기준에 적용 가능한 방안을 모색하였다.

Table 1. BSIMM SSF

SSF			
Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

3.1 거버넌스

거버넌스는 〔Table 2〕와 같이 전략과 메트릭스, 컴플라이언스와 정책, 교육으로 구성된다. 전략과 메트릭스는 'Level 1: 방향과 전략의 이해', 'Level 2: 전략을 조율하고 책임을 명확화'와 'Level 3: 위험기반 포트폴리오 관리'를 요구한다. 이는 정보보호시스템을 개발하고 유지하기 위한 관리, 통제 절차를 평가하는 생명주기 지원 클래스의 생명주기 정의(ALC_LCD)와 유사하다.

거버넌스 수립을 위해 정보보호시스템 개발을 위해 내부 '정보보호시스템 개발 프로세스'를 게시하고, 프로세스 관리를 위한 그룹과 관리자와 프로세스에 대한 위험 수용 범위와 책임을 명시해야 하며, 해당 프로세스 수행의 검토를 위한 도구가 적용되어야 한다.

컴플라이언스와 정책은 'Level 1: 규칙, 규제, 컴플라이언스 문서 통합'과 'Level 2: 컴플라이언스와 정책에 의한 내부 practice 조정', 'Level 3: 조직의 위협, 공격, 결함 및 운영 문제를 기반으로 정책 개선 및 벤더 요구사항 반영'을 말한다. 개발 조직의 보안정책, 규제 및 통제 등에 대한 내용으로 공통평가기준의 보안문제 중 조직의 보안 정책과 다르다.

Table 2. Governance of SSF and CC

SSF		Common Criteria
Governance	Strategy and Metrics	ALC_LCD
	Compliance and Policy	-
	Training	-

'정보보호시스템 개발 프로세스'에 대한 내부 정책, 컴플라이언스 정의 및 개선을 수행해야 한다.

교육은 'Level 1: 소프트웨어 보안인력 양성', 'Level 2: 요구사항 기반으로 최적화된 역할 기반 교육'과 'Level 3: Provide recognition for skills and career path progression'을 요구한다. 공통평가기준은 교육에 대해서는 명시하고 있지 않다. 교육을 위해서 '정보보호시스템 개발 프로세스'의 내부 관리자와 교육 담당자 정의 및 업무시간을 수립해야 한다. 또한, 교육 담당자는 개발 프로세스에 대한 교육을 받아야하며 보안 문화 정착 및 주기적인 개선을 수행해야 한다.

3.2 인텔리전스

인텔리전스는 〔Table 3〕과 같이 공격 모델, 보안기능과 설계, 규격과 요구사항으로 구성된다. 공격 모델은 'Level 1: 지식을 기반으로 한 공격 및 데이터 자산 정의', 'Level 2: 공격과 공격자에 대한 정보 제공' 및 'Level 3: 새로운 공격 패턴 연구'를 요구한다. 정보보호시스템의 보안 문제로 정의할 수 있는데, 공통평가기준에서는 잠재적 위협이나 결함, 공격 등에 대한 취약성 분석(AVA_VAN, Vulnerability aNalysis)의 위협원 조사 및 분석과 같다. 평가자 뿐 아니라 조직 내부에서 위협원에 대한 기존 정보 수집 및 목록화, 위협원에 대한 정보 분석 및 체계화를 해야 하며, 향후 위협원에 대한 연구 및 위협원에 대한 예측 및 대응을 수행하는 대응팀을 보유해야 한다.

보안기능과 설계는 'Level 1: 보안기능과 아키텍처 게시', 'Level 2: 보안 솔루션 정의 및 구축' 및 'Level 3: 승인된 보안기능과 보안 설계 프레임워크를 재사용'에 대해 확인한다. 공통평가기준에서는 보안기능 요구사항(APE/ASE_REQ)에 따라 보안기능 명세(ADV_FSP)와 TOE 설계(ADV_TDS)가되며,

Table 3. Intelligence of SSF and CC

SSF		Common Criteria
Intelligence	Attack Models	AVA_VAN
	Security Features and Design	ADV_FSP ADV_TDS
	Standards and Requirements	-

상호보완 관계를 갖는다. '정보보호시스템 개발 프로세스' 내에서 보안기능과 보안 설계 구조 분석을 수행해야 하고 이를 위한 라이브러리 구축 및 게시가 수반되어야 한다. 또한, 관리자의 보안 설계 검토와 승인된 보안 프레임워크의 효율적인 사용이 필요하다.

규격과 요구사항은 'Level 1: 보안 규격과 컴플라이언스 중심의 요구사항 제공', 'Level 2: 내부 및 벤더에게 공식적으로 승인된 규격' 과 'Level 3: 오픈소스 사용을 위한 위험 관리 결정'를 확인한다. 공통평가기준의 보안 기준(위협원, 보안기능 및 구조)과 평가 방법을 내부 프로세스에 적용해야 한다. 이러한 보안 프로세스 정형화 및 검토자가 필요하다.

3.3 안전한 소프트웨어 개발 라이프 사이클

안전한 소프트웨어 개발 라이프 사이클은 〔표 4〕와 같이 아키텍처 분석과 코드 검토, 보안 시험으로 구성된다. 아키텍처 분석은 'Level 1: 소프트웨어 위험기반 구조 분석으로 설계 단계에서 증명', 'Level 2: 아키텍처 분석 프로세스 사용'과 'Level 3: 아키텍처 설계 그룹의 검토 및 개선 능력 구축'을 요구한다. 공통평가기준에서는 TOE 설계(ADV_TDS), 구현의 표현(ADV_IMP), 보안 아키텍처(ADV_ARC)를 통해 설계 및 구현에 대한 보안성을 보증하지만, 조직의 보안 설계 검토 및 분석 능력은 평가하지 않는다.

즉, '정보보호시스템 개발 프로세스'에는 보안기능에 대한 설계 검토 및 구조 분석을 위한 조직 내 프로세스 정의가 필요하다.

코드 검토는 'Level 1: 코드 검토 수행', 'Level 2 : 자동 코드 검토 및 중앙 집중화된 보고를 통해 표준을 시행' 및 'Level 3: 최적화된 규칙과 자동화된 평가 기술을 결합'을 확인한다. 공통평가기준에서는 취약성 분석(AVA_AVN)에서 모든 취약점의 식별 및 분석을 요구하고 있지만, 버그를 목록화하고 자동화된 도구 사용하여 코드 검토를 수행해야 하며, 코딩 표준을 사용하고 코드 검토 강제 수행하도록 중앙에서 제어가 필요하다. 또한, 평가기법을 토대로 한 최적화된 코드 규칙과 특정 버그 삭제를 위한 보안기능을 구현해야 한다. 보안 시험은 'Level 1: 보안을 고려한 QA(Quality Assurance)', 'Level 2: 공격자 관점의 시험 계획'과 'Level 3: 위험 기반 보안 시험'이 요구된다. 공통평가기준은 기능명세(ADV_FSP), TOE 설계(ADV_TDS)와 구현의 표현(ADV_IMP)과 같이 보안기능이 동작하는지 시험(ATE)해야 한다. 그러나 QA 프로세스 내에 보안기능 시험을 정의해야 하며 자동화된 수행이 필요하다. 경계 값 조건 시험을 수행해야 하며, 공격자 관점의 보안 시험과 블랙박스 보안 시험을 수행해야 한다.

3.4 배포

배포는 〔Table 5〕와 같이 침투시험과 소프트웨어 운영환경, 환경설정 및 취약성 관리로 구성된다. 침투 시험은 'Level 1:내외부 침투 시험 절차 정의 및 결과의 결점이나 문제 수정 및 개선', 'Level 2: 내부 침투 테스터에 의해 정기적으로 침투 시험 수행' 과 'Level 3: 공격의 발전을 고려한 심도 있는 침투 시험'이 요구된다. 공통평가기준에 의해 TOE의 침투 시험(AVA_VAN)을 수행하고 있다. 그러나 개발자에게 요구하고 있지 않으며, 프로세스 및 침투 방법에 대해 상세히 기술되어 있지 않다. 이를 위해 개발자는 소스코드, 설계문서, 구조 분석 결과, 코드 검토를 참조하여 정기적으로 내부 침투 시험 절차를 수립 및 수행해야 한다. 또한, 외부 전문가를 활용한 모의 해킹 절차도 병행해야 한다.

소프트웨어 운영환경은 'Level 1: 소프트웨어를 모니터링하고 호스트/네트워크 보안 환경 제공', 'Level 2: 소프트웨어 개발 환경에 애플리케이션 설치가이드를 제공하고 보안을 위해 모니터링, 코드사

Table 4. SSDL Touchpoints of SSF and CC

SSF		Common Criteria
SSDL Touchpoints	Architecture Analysis	ADV_TDS ADV_IMP ADV_ARC
	Code Review	-
	Security Testing	ATE

Table 5. Deployments of SSF and CC

SSF		Common Criteria
Deployments	Penetration Testing	AVA_VAN
	Software Environment	ALC_DVS
	Configuration Management and Vulnerability Management	ALC_CMC ALC_CMS ALC_FLR

이닝' 및 'Level 3: 코드 보호'를 수행하는 것을 요구한다. 공통평가기준의 개발보안(ALC_DVS)에서 개발 환경의 물리적, 절차적, 인적 보안대책에 대해 기밀성, 무결성이 보장되고 있음을 요구하지만, 세부사항은 언급하지 않는다. 보안 대책을 증명하기 위해 소프트웨어 운영환경의 요소를 활용할 수 있을 것이다. 환경설정 및 취약성 관리는 운영 모니터링 및 긴급대응에 대한 요구사항으로 'Level 1: 운영 모니터링과 사건 및 버그 식별', 'Level 2: 공격을 받는 동안 긴급대응 가능함을 보장'과 'Level 3: 운영기반의 소프트웨어 버그 수정 및 개발 프로세스 향상'이 있다. 공통평가기준에서는 정보보호시스템의 변경을 추적하기 위한 형상관리 능력(ALC_CMC)과 형상관리 범위(ALC_CMS), 보안결함을 추적하고 교정하는 결함교정(ALC_FLR)이 존재한다. 그러나 운영 및 긴급 대응, 개발 프로세스 향상에 대해 검증하지 않으므로 이 부분에 대한 개선이 필요하다.

IV. 정보보호시스템의 보안 설계 방법

자체보호는 자산을 안전하기 보호하기 위한 정보보호시스템에 매우 중요한 요소이다. 그러나 보안성 평가를 통해 보증 받은 시스템도 잠재적인 취약성 및 공격의 가능성을 완전히 배제할 수 없다. 이에 대응하기 위해 잠재 취약성의 연구 및 발생을 방지하기 위한 보안기능 구현 및 모의 침투시험 등이 필요하다. 그러나 공통평가기준은 보안 설계 프로세스 평가에 중점을 두고 있지 않기 때문에 정보보호시스템에 대한 보안구조 설계가 일관되지 않을 수 있다. 본 장에서는 보안 요구사항을 만족하며, 일관된 보안 수준을 유지하기 위한 [Fig 1]의 정보보호시스템 보안 설계 방법을 제안한다. 정보보호시스템(Products) 보안 설계 방법은 취약성과 공격분석(CWE-CVE-CAPEC)을 기반으로 보안 아키텍처 수립, 보안기능(Security Functions) 설계, 구현 및 검증(Tool)하는 프로세스이다. 단, TOE(Target of Evaluation)는 보안성 평가대상을 의미하며 정보보호시스템의 일부분이나 조합일 수 있다[3]. 이러한 프로세스를 수행하는 데는 관리자, 전문가, 분석가 및 교육 담당자 등이 필요하다.

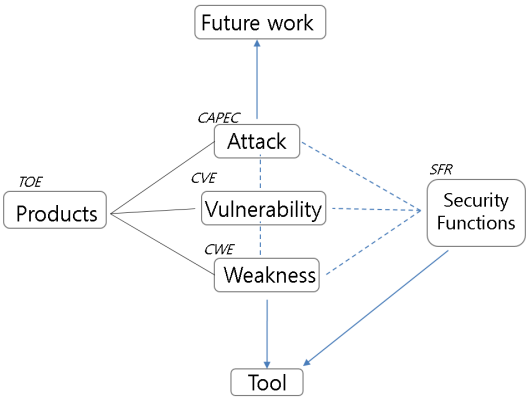

Fig. 1. Design of Information security protection system

4.1 취약성과 공격 분석

공통평가기준의 보안문제정의(ASE_SPD)에 의해 정보보호시스템의 위협원, 자산, 악의적 행동의 관점에서 위협이 식별되어야 하며, 보안정책과 운영환경을 고려해야한다[3]. 시스템은 약점을 내재하고 있으며, 공격자에 의해 악용되어 취약성으로 발현될 수 있다. 정보보호시스템에 대한 취약성의 존재 가능성과 이용 가능성 및 위험을 감소시키기 위해 취약성의 식별, 제거 또는 완화 등의 대응을 위한 수준별 분석 프로세스가 필요하다. [Fig 2]는 BSIMM의 공격모델을 공통평가기준의 침투시험(AVA_VAN)에 적용한 단계별 취약성 분석 방법이다.

- S 1.CWE-CVE-CAPEC을 기준으로 한 시스템 대상 위협원 정보 수집 및 목록화
- S 2. CWE-CVE-CAPEC을 기준으로 한 위협원에 대한 정보 분석 및 연결
- S 3. 발생 가능한 위협원에 대한 연구 및 대응체계 마련 (운영환경 및 연관 시스템에 의해 발

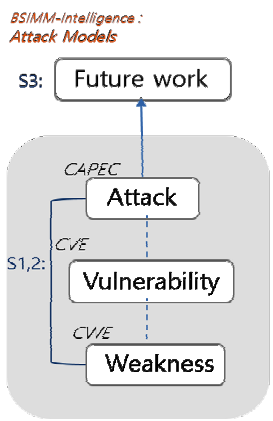

Fig. 2. Analysis of vulnerabilities and attacks

생할 있는 위험이나 융합 및 미래 기술의 취약성에 대한 분석 등)

S 1. S 2.를 통해 시스템군의 취약성을 식별하고 제거해야 하며, S 3.로 취약성의 제거 및 완화 등의 대응이 가능해야 한다.

4.2 보안기능과 설계

정보보호시스템은 취약성 및 공격을 보호하기 위한 보안기능이 구현 및 설계 되어야한다. SFR을 구현한 보안기능은 시스템의 취약성을 보완하는 기능을 말한다. 그러므로 단계별 보안기능 구현과 설계는 〔Fig 2〕와 같이 취약성 및 공격의 수준을 고려하여 구현되어야 한다. 〔Fig 3〕은 BSIMM의 보안기능구현과 설계를 공통평가기준의 기능 명세(ADV_FSP)와 TOE 설계(ADV_TDS)에 적용한 수준 별 방법이다.

- S 1. 보안기능 식별 및 설계
- S 2. 보안 프레임워크 구축
- S 3. 보안기능과 설계 검토 및 재사용

S 1. S 2.를 통해 해당 시스템군의 취약성에 대응하기 위한 보안기능을 명세하고 설계해야 하며, S 3.를 통해 보안기능과 설계관점에서 시스템 간 호환성 및 연관성을 검토하고 재사용할 수 있어야 한다.

Fig. 3. Security features and designs

4.3 보안 아키텍처 분석

〔Fig 4〕의 보안 아키텍처 분석에서 보안기능은 공통평가기준의 보안 아키텍처(ADV_ARC)의 자체보호, 영역분리, 우회불가성을 말한다. 〔그림 3〕의 보안기능과 보안 아키텍처 분석은 분리되어 검증하여야 한다〔3〕.

Fig. 4. Security Archtecture Analysis

S 1. 아키텍처 기반 보안기능 분석
S 2. 아키텍처 분석에 대한 프로세스 정의
S 3. 보안 아키텍처 설계의 개선 및 검토

정보보호시스템의 보안을 보장하기 위해 S 1. 아키텍처 기반으로 보안기능이 분석되어야 하며, S 2. 프로세스가 수립되어야 한다. 또한, S 3. 전문가에 의해 프로세스 전체를 검토하고 개선할 수 있어야 한다.

4.4 보안성 검증

정보보호시스템의 보안성 검증을 위한 시험 방법으로 〔Fig 3〕의 보안기능과 설계에 대한 코드 분석과 위험 기반 시험이 수행되어야 하며, 〔그림 2〕에 대한 모의 침투 시험을 해야 한다. BSIMM의 코드 검토는 구현의 표현 (ADV_IMP), 위험 기반 시험은 BSIMM의 보안 시험을 공통평가기준의 시험(ATE)에 BSIMM의 침투 시험은 공통평가기준의 취약성 분석(AVA_VAN)에 적용하였다.

〔코드 분석〕

- S 1. 자동화 툴을 통한 코드 분석
- S 2. 강제적인 코드 검토
- S 3. 지식기반의 규칙을 적용한 자동화 툴을 통한 코드 분석

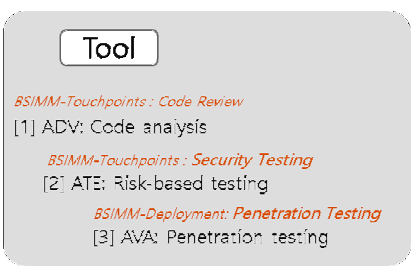

Fig. 5. Level security verification

〔위험 기반 시험〕
- S 1. 경계 값 기반 보안성 품질 시험
- S 2. 공격자 관점의 블랙박스 시험
- S 3. 퍼지 값 기반 시험 및 보안 시험 자동화

〔침투 시험〕
- S 1. 외부 침투 시험
- S 2. 내부 침투 시험
- S 3. 전문가를 통한 외부 침투 시험 및 시스템에 최적화된 침투 시험

정보보호시스템의 보안성을 검증하기 위해 단계별 S 1~3 코드분석이 필요하다. 또한, 시스템에 대한 단계별 위험 기반 시험과 침투 시험이 요구된다. 이와 같이 보안성을 검증하기 위한 방법의 S 1.은 상용 도구를 통해 가능하지만, S 2.는 내부 정보를 인지하고 수행해야 하며, S 3. 은 전문가가 필요하다.

V. 결 론

본 논문에서는 안전한 정보보호시스템의 보안설계를 위해 공통평가기준과 BSIMM을 비교 분석하였으며, 공통평가기준의 보증요구사항을 바탕으로 BSIMM의 보안 프레임워크를 활용한 정보보호시스템의 보안 설계 방안을 제안하였다. 정보보호시스템 보안 설계 방안은 단발성 보안성 검증의 한계를 극복하고, 개발 보안 성숙도를 개선 및 발전시킴으로서 정보보호시스템의 지속적인 보안성 강화의 큰 발판을 마련할 수 있을 것이다. 향후에는 본 논문에서 제안하는 정보보호시스템의 보안 설계방안이 보안성 평가에 중요한 역할을 할 수 있는지 증명하기 위해 평가자 및 업체를 대상으로 설문을 수행할 것이며, 실제 제품에 적용하여 평가를 수행할 계획이다.

References

〔1〕 Gary McGraw, Ph.D., Brian Chess, Ph.D., & Sammy Migues, "Building Security In Maturity Model," the Creative Commons Attribution-Share Alike 3.0 License, May 2010

〔2〕 ISO(International Organization for Standardization), "Evaluation criteria for IT security-ISO/IEC 15408-1/2/3:2009," Dec 2009

〔3〕 ISO(International Organization for Standardization), "Methodology for IT Security Evaluation-ISO/IEC 18045:2009," Dec 2009

〔4〕 CWE, "https://cwe.mitre.org"

〔5〕 CVE, "http://cve.mitre.org"

〔6〕 CAPEC, "http://capec.mitre.org"

〔7〕 Jae-Woo Im, "Refining software vulnerbility Analysis under ISO/IEC 15408 and 18045," Journal of The Korea Institute of Information Secyrity & Cryptology, 24(5), pp 969-974, Oct. 2014

〔8〕 Yeon-Hee Kang,young-Hwan Bang, Gang-Soo Lee, "Development of Security Evaluation Management System Based on Common Criteria," The Journal of Society for e-Business Studies, 10(3), pp 67-83, Aug 2005

〔9〕 Jinseok Park, Heesoo Kang, Seungjoo Kim, "How to Combine Secure Software Development Lifecycle into Common Criteria," Journal of The Korea Institute of Information Secyrity & Cryptology, 24(1), pp 171-182, Feb 2014

〈저 자 소 개〉

박 정 섭(Jung-Sup Park) 정회원
1999년 2월: 한국외국어대학교 정치외교학과 학사
2001년 2월: 한국외국어대학교 경영정보학과 석사
2008년 3월~현재: 서울과학기술대학교 IT정책대학원 박사과정
〈관심분야〉 정보보호