정 보 보 호 학 회 지
제26권 제1호, 2016. 2

# BSIMM을 통해 살펴본 보안성숙도모델의 분석

서 동 수*

요 약

최근 들어 보안사고의 대응 방식이 네트워크 보안이나 운영체제 보안과 같은 전통적인 방법에서 벗어나 개발생명주기 보안이나 SW 공급망 보안으로 확대되고 있다. 이러한 흐름에서 주목받는 것이 BSIMM과 같은 소프트웨어 보안성숙도 모델이다. 보안성숙도 모델은 소프트웨어 시스템의 보안성 향상을 위해 관리자와 개발자가 중점을 두어야 할 부분을 스스로 평가할 수 있도록 하는 프레임워크이다. 본 고에서는 BSIMM을 통해 보안성숙도 모형이 갖는 특징을 소개하며, 이의 활용에 대해 살펴본다.

## Ⅰ. 서 론

최근 논란이 되고 있는 APT 공격과 같은 진화된 보안공격은 애플리케이션 소프트웨어의 보안에 대해 새로운 경각심을 환기시키고 있다[1]. 날로 치밀해지는 보안공격에 대해 견고하게 대응할 수 있는 정보시스템을 개발하기 위해 기업에서는 많은 비용을 투자하여왔다. 그럼에도 불구하고 만족할만한 효과를 얻지 못하는 것 또한 현실이다. 보안공격에 대해 기존의 접근방법이 갖는 한계는 대부분 사후대응 중심의 활동이라는 점이다. 안티바이러스 백신의 설치, 결함 코드의 제거, 보안패치의 제공 등은 전형적인 사후대응 활동이며, 사고 후 보안인력의 강화 역시 같은 맥락에서 생각할 수 있다.

최근 연구에 의하면 사후대응 중심의 접근보다는 선제적 대응이 비용 효과측면에서 우수하다고 보고되고 있다[2]. 선제적 대응이란 보안 취약점을 소프트웨어 개발단계에서 사전에 제거함으로써 이후 서비스 운영 시 발생할 보안 문제를 완화시키는 방법이다. 선제적 대응 방법으로는 시큐어코딩, 보안설계와 같이 개발생명주기 활동을 강화시키는 방법, 보안활동 수준을 강화시킬 수 있는 최고실무(best practices)를 도입하는 방법을 생각할 수 있다.

본 고에서는 침해공격에 대한 선제적 대응을 골자로 하는 보안활동과 이들 활동의 성숙도를 측정하는 모형인 BSIMM의 특징과 구성을 소개한다. 이를 통해 소프트웨어 개발 조직의 보안 대응 수준을 평가하고 높일 수 있는 방안에 대해 살펴본다.

## Ⅱ. 소프트웨어보증과 보안

### 2.1. 소프트웨어 보증(SwA)의 개념

1990년대에 등장한 안전한 소프트웨어 방법론(Trusted Software Methodology)[3]은 소프트웨어 개발단계의 품질을 강화시킴으로써 견고한 소프트웨어 개발할 수 있다는 구상을 역설한 최초의 시도이다. 이후 개발 프로세스의 품질이 소프트웨어 보안성에 큰 영향을 줄 수 있다는 점은 2003년 미국 국토안보부와 국가표준기술연구소(NIST)가 주도한 소프트웨어 보증(Software Assurance, SwA) 프로젝트를 통해 본격적으로 알려지기 시작했다[4].

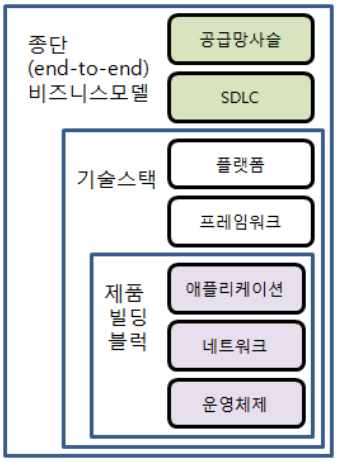
일반적으로 소프트웨어 보증은 소프트웨어 품질 확보를 위한 제반기술 및 관리활동을 말한다. 반면에, 국토안보부의 SwA는 사이버테러로부터 국가기간망과 주요정보자산을 보호하기 위한 목적의 품질보증 활동이라 볼 수 있다. 이러한 목적의 SwA는 시큐어코딩, 시큐어 SDLC, 최고 실무 등을 포함하는 전반적인 보안 보증 활동으로 지칭하는 개념으로 사용된다. 현재 SwA에 대한 연구는 소프트웨어 개발 및 상용 소프트웨어를 공급, 획득하는 공급망 사슬 영역까지 넓혀 진행되고 있

* 성신여자대학교 IT학부 (dseo@sungshin.ac.kr)

다. 보안 영역의 SwA는 다음 네 가지 영역을 포함한다.

- 시큐어 소프트웨어 컴포넌트: 상용 패키지 소프트웨어를 포함하는 컴포넌트의 보안품질을 다룬다.
- 시큐어 소프트웨어 생명주기: 소프트웨어 생명주기 내에서 정의된 분석, 설계, 코딩, 테스팅, 유지보수 단계의 보안 활동을 정의하고 관리하는 영역을 다룬다.
- 서비스의 소프트웨어 보안: SaaS와 같은 플랫폼에서 제공되는 소프트웨어 서비스를 대상으로 한다.
- 소프트웨어 공급망사슬: 비즈니스 모델 개념에서 제공되는 서비스를 사용하는 사용자와 이를 제공하는 공급자 간의 종단에서 발생하는 보안 문제를 다룬다.

SwA는 기존의 보안접근과는 틀리게 단계적 방어(defense in depth)시각에서 보안 문제에 접근한다. 단계적 방어란 악의적인 공격에 대응하는 절차를 네트워크와 운영체제와 같은 제품 블럭에서, 플랫폼, 프레임워크와 같은 상위 기술영역, 그리고 개발생명주기, 소프트웨어 공급망 수준의 종단 비즈니스 모델로 보안 방어망을 확대하는 개념의 방어이다[그림 1]. 단계적 방어가 주목받는 이유는 보안활동이 특정 단계에서만 수행되는 활동이 아닌 계층적인 활동을 강조한다는 점이다.

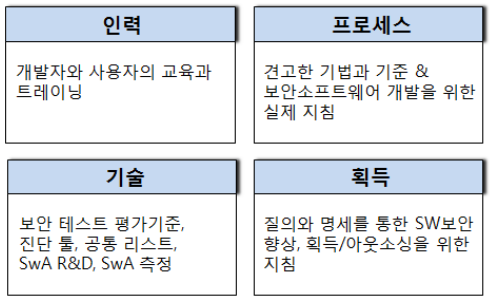



〔그림 1〕SwA의 수직구성

### 2.2. Build Security In

단계적 방어가 효과적으로 작동하기 위해서는 체계적으로 정리된 적용절차가 필요하다. 보안구축이라 불리는 Build Security-In (BSI)은 소프트웨어 개발자, 설계자, 보안담당자가 개발 단계에서 보안 기능을 구축하는 데 사용할 수 있는 방법, 도구, 지침, 규칙 및 제반 자원을 제공하는 노력이다[4]. BSI를 통해 설명되는 방법론은 특정 기술이나 생명주기 모형을 지원하기 위해 제공되는 것은 아니다. 오히려 BSI는 소프트웨어공학 측면과 관리 측면을 폭넓게 포함함으로써 모든 개발기법과 결합이 가능하도록 설계되었다. 이를 위해 BSI는 다음 네 영역에서 필요한 실무를 담고 있다[그림 2].

- 인력 영역: 시스템 개발자와 관리자를 위한 보안 개념과 보안 활동의 세부 사항을 주기적인 교육을 통해 공고히 한다. 이 부분은 선택이 아닌 보안의 필수 활동으로 정의한다.
- 프로세스 영역: 소프트웨어 공학 개념에 기반을 두며, 소프트웨어 개발 생명주기에서 정의되는 계획, 요구분석, 설계, 구현, 테스트 활동에 대한 보안강화 요소를 정의한다.
- 기술 영역: 프로세스 영역에서 정의된 단계별 영역에서 사용되는 개별 기법의 내용, 특성, 그리고 이와 관련 도구를 설명을 한다.
- 획득 영역: 획득과 관련한 생명주기 모형은 획득, 계약, 구현, 지속관리에 관한 세부 활동 및 지침을 설명한다.




〔그림2〕 BSI의 영역 구성

## Ⅲ. BSIMM의 구성

소프트웨어 보안조치(Software Security Initiative, SSI)란 악의적인 공격으로부터 견고한 소프트웨어를 구축하기 위해 필요한 소프트웨어 보안의 계획, 개발, 관리, 측정하고 교육을 하기위한 실무 집합이다. 소프트웨

어 보안조치와 관련한 BSIMM (Building Security In Maturity Model)의 목적은 IT 기업에서 실제 수행된 소프트웨어 보안조치에 대해 자료를 수집하고 보안활동을 식별하거며, 계량화하는데 있다. 보안 활동을 식별하고 계량화하는 과정에서 발생하는 문제는 IT 기업이 각자 고유한 개발방법론을 적용해서 시스템을 개발하는 상황에서 발생하는 다양성에 관한 것이다. 즉, 단일 시각에서 보안활동을 파악하고 서술하는 것이 가능한가라는 점은 객관성 확보를 위해 필요한 사항이다. 이러한 이유로 BSIMM은 소프트웨어 보안프레임워크 (Software Security Framework)체계를 제안한다.

보안 프레임워크가 유용한 이유는 다음 두 가지이다. 첫째, 보안조치를 설명하기 위한 공통용어를 제공한다. 기존의 소프트웨어 보안조치들은 용어의 차이, 작동환경, 지원 규모, 프로덕트의 차이 등으로 인해 보안조치들을 비교 서술하는 것이 용이하지 않았다. 소프트웨어 보안프레임워크는 이러한 문제에 대해 공통 용어를 제공한다. 둘째, 보안조치를 구분하기 위해 BSIMM은 110개의 상세한 활동을 정의하고 있다. 나아가 BSIMM은 이들 상세활동은 4개 도메인, 12개 실무로 그룹화하여 계층적인 개념구조로 설명한다.

- 도메인(domain): 보안 프레임워크의 가장 상위요소이며, 거버넌스, 인텔리전스, 보안 소프트웨어 개발 생명주기 터치포인트, 그리고 배치 등 4개 도메인을 갖는다.
- 활동(activity): 소프트웨어 보안그룹에 의해 수행되는 보안액션이며, 활동은 3개 수준으로 구분된다.
- 실무(practice): 실무란 활동을 12개의 그룹으로 묶은 각각을 말한다.

### 3.1. BSIMM 도메인과 실무

소프트웨어 보안프레임워크에 정의된 4개 도메인을 소개하면 다음과 같다.

- 거버넌스: 소프트웨어 보안조치를 위해 필요한 구서, 관리, 그리고 측정과 관련된 실무 집합이다.
- 인텔리전스: 전사적인 소프트웨어 보안 활동의 수행과 관련된 조직의 지식을 수집하는데 관련된 실무집합이다.
- SSDL 터치포인트: 특정의 소프트웨어 개발 결과물에 대한 분석과 평가, 그리고 프로세스에 관련된 활동과 관련된 실무집합이다.
- 배치: 전통적인 네트워크 보안과 소프트웨어 유지보수 조직과의 인터페이스에 관련된 실무집합이다.

각 도메인은 3개씩의 실무를 포함한다. 각 도메인별로 정의된 실무를 소개하면 다음과 같다.

#### 가. 거버넌스 도메인

- 전략과 메트릭: 계획, 역할 할당, 책임, 소프트웨어 보안 목적의 식별, 예산 결정, 메트릭과 보안게이트를 식별한다.
- 순응성과 정책: COTS 소프트웨어 위험을 통제하기 위한 SLA와 같은 계약상의 통제를 개발하거나, 보안 정책의 수립, 그러한 정책을 감사하는 등의 순응성과 이와 관련된 정책을 수립한다.
- 교육: 개발자와 설계자는 종종 보안지식이 결여된 상태에서 시작하기 때문에 교육은 소프트웨어 보안에 관해서 항상 중요한 역할을 수행한다.

#### 나. 인텔리전스 도메인

- 공격 모델: 공격모델은 공격자 관점에서 그들처럼 생각할 수 있는 정보를 수집하는데 관심이 있다. 이러한 유형으로는 위협모델, 오용사례 개발과 정제, 자료의 분류, 특정 기술별 공격 패턴의 수집 등이 있다.
- 보안 속성과 설계: 주요 보안 통제를 위해 사용가능한 보안패턴을 생성하며, 이러한 통제를 위한 프레임워크를 개발하고, 적극적인 보안 가이드를 만들고 공지한다.
- 표준과 요구사항: 조직으로부터 명시적인 보안 요구사항을 추출하고, 어떤 COTS를 추천해야 할지를 결정한다. 또한, 주요 보안제어(인증, 유효성 검사 등)를 위한 표준을 개발하며, 현재 사용 중인 기술에 대한 보안 표준을 생성하고, 표준검토위원회를 설립한다.

#### 다. SSDL 터치포인트 도메인

- 아키텍처 분석: 소프트웨어 아키텍처를 간결한 다이어그램을 통해 보여주는 활동, 위험과 위협 목록의 활용, STRIDE나 아키텍처 위험 분석과 같은 리뷰 프로세스 적용, 그리고 보안 평가와 문제완화

계획의 수립 등을 포함한다.
- 코드 리뷰: 코드 리뷰 도구의 사용과 정적분석 규칙을 정제한 맞춤형 규칙개발, 도구 사용 자료를 수집하며, 분석하고, 결과를 측정하거나 추적하는 활동을 포함한다.
- 보안 테스팅: 퍼징 도구와 같은 블랙박스 보안 도구의 사용, 위험기반 화이트박스 테스팅, 공격 모델의 응용, 그리고 코드 커버리지 분석과 같은 활동을 포함한다.

라. 배치 도메인
- 침투 테스팅: 최종 형상항목에 대한 취약성에 초점이 맞추어져 있으며, 결함 관리와 완화 활동에 대해 직접적인 정보를 제공한다.
- 소프트웨어 환경: 운영체제, 플랫폼 패치, 웹 어플리케이션 방화벽 설치와 배치에 대한 문서, 어플리케이션 모니터링, 변경 관리 등과 관련이 있다
- 형상관리와 취약성 관리: 패칭, 어플리케이션 업데이트, 버전 관리, 결함 추적과 완화, 발생하는 사건을 다루는 일과 관련이 있다.

〔표1〕 소프트웨어 보안프레임워크의 구성

| 거버넌스 | 인텔리전스 | 터치포인트 | 배치 |
|---|---|---|---|
| 전략과 매트릭 | 공격 모델 | 아키텍처 분석 | 침투 테스팅 |
| 순응성과 정책 | 보안 속성과 설계 | 코드 리뷰 | 소프트웨어 환경 |
| 트레이닝 | 표준과 요구 | 보안 테스팅 | 형상관리와 취약성 관리 |

〔표2〕 공격모델 실무(AM)의 구성예

| 공격모델 (Attack model, AM) | | |
|---|---|---|
| 레벨 1 | | |
| 활동 명세 | 활동# | 참여자 |
| 상위N 수준의잠재 공격목록은 구축하고 관리함 | AM1.1 | 22% |
| 자료 분류 스키마와 목록을 생성함 | AM1.2 | 65% |
| 잠재적 공격자를 식별함 | AM1.3 | 40% |
| 공격 스토리를 수집하고 알려줌 | AM1.4 | 10% |
| 공격정보를 수집하고 사용함 | AM1.5 | 59% |
| 공격에 대한 토론을 할 수 있는 내부 포럼을 구성함 | AM1.6 | 14% |
| 레벨 2 | | |
| 잠재적 공격에 대한 공격 패컨과 오용 유스케이스를 작성함 | AM2.1 | 8% |
| 기술종속적인 공격 패턴을 생성함 | AM2.2 | 10% |
| 레벨 3 | | |
| 새로운 곤약 패턴 발굴을 위한 과학적 팀을 구성함 | AM3.1 | 5% |
| 공격자가 할 행위를 생성하고 자동화하여 사용함 | AM3.2 | 3% |

이들 4개의 도메인에서 정의되는 실무목록은 [표 1]과 같다. 여기서 주목할 것은 실무와 활동의 관계이다. 하나의 실무는 여러 활동을 포함하며, 실제 측정은 각 활동에서 일어난다는 점이다. 예로서, [표 2]는 SSDL 터치포인트 도메인에서 정의된 공격모델의 실무 구성을 보여준다. 공격모델 실무는 총 10개의 활동(AM1.1~AM3.2)을 가지며, 이들 활동은 수준별로 레벨 1~3 수준으로 구분이 된다. [표 2]의 참여자 항목은 인터뷰에 참여한 기업이 배정한 인력 비율 분포를 나타낸 것이다. 이를 통해 해당 기관이 보는 특정 보안활동의 상대적인 중요성을 알 수 있다.

### 3.2. BSIMM의 진화

BSIMM의 다른 특징은 보안 문제에 대해 처방목적의 조치보다는 보안성숙도에 대한 서술적 측정 기준을 제공하는 것이다. 이를 위해 2008년도에 제작된 BSIMM의 최초 버전이 다음의 과정을 통해 자료를 수집하고 이를 바탕으로 프레임워크를 제작하였다.
- 소프트웨어 보안프레임워크를 생성하기 위해 참여자 자신의 소프트웨어 보안 실무 경험을 활용함
- 9개 IT기업의 보안책임자를 대상으로 개별인터뷰를 통해 공통 보안활동을 식별함
- 각 기업에서 수행한 보안조치에 대해 활동별 스코어카드를 생성하고 이를 반영한 보안프레임워크 구축함. 구축된 결과는 실무자들에게 보여주어 확인을 받음
- 결과적으로 수집된 보안활동은 110개에 달하며, 이들은 다시 3개 수준의 성숙도 단계로 분류하여 보안 프로임워크에 반영함

이후 개정된 BSIMM2는 30개 기업으로부터 자료를 수집한 결과를 반영하였으며, BSIMM-V에서는 67개 기업의 보안활동 자료를 반영하여 해외 IT 기업의 보안활동 수준 성숙도평가 근거를 지속적으로 업데이트 하

였다. 가장 최신 버전으로 2015년 발표된 BSIMM 6는 78개의 기업을 대상으로 202개의 개별 활동을 추출한 것으로 보고되었다[6].

## Ⅳ. BSIMM의 활용

기존의 CMMi나 SSE-CMM과 같은 성숙도모형과 비교할 때 BSIMM이 갖는 장점은 보안활동의 최고실무정보를 지속적인 축적하고, 모델에 반영한다는 점이다. 이를 통해 공통적으로 적용이 가능한 핵심보안 활동을 식별하고, 이를 통해 새로운 보안위협에 대처하는 IT 기업의 트랜드를 반영할 수 있다,

| GOVERNANCE | | INTELLIGENCE | |
|---|---|---|---|
| ACTIVITY | OBSERVED | ACTIVITY | OBSERVED |
| [SM1.1] | 41 | [AM1.1] | 17 |
| [SM1.2] | 40 | [AM1.2] | 51 |
| [SM1.3] | 36 | [AM1.3] | 31 |
| [SM1.4] | 66 | [AM1.4] | 8 |
| [SM2.1] | 36 | [AM1.5] | 46 |
| [SM2.2] | 29 | [AM1.6] | 11 |
| [SM2.3] | 30 | [AM2.1] | 6 |
| [SM2.5] | 17 | [AM2.2] | 8 |
| [SM2.6] | 29 | [AM3.1] | 4 |
| [SM3.1] | 15 | [AM3.2] | 2 |

〔그림 3〕 BSIMM 스코어카드의 예

〔표 3〕 코어 활동 목록

| 모든 조직이 수행하는 12개 코어 활동목록 | |
|---|---|
| 활동 | 설명 |
| SM1.4 | 보안게이트 식별과 필요자원의 수집 |
| CP1.2 | PII 의무를 식별한다 |
| T1.1 | 보안인식 트레이닝을 실시한다 |
| AM1.2 | 자료 분류 스키마와 목록을 생성한다 |
| SFD1.1 | 보안 특성을 구축하고 공지한다 |
| SR1.1 | 보안 표준을 생성한다 |
| AA1.1 | 보안 특성을 리뷰한다 |
| CR1.4 | 수동리뷰와 함께 자동도구를 사용한다 |
| ST1.3 | 보안요구기반 보안 테스팅을 수행한다 |
| PT1.1 | 문제파악을 위해 침투테스트를 수행함 |
| SE1.2 | 네트워크와 호스트의 보안기본을 확인한다 |
| CMVM1.2 | 운영 모니터링을 통해 확인되 버그를 개발자에게 통지한다 |

### 4.1. 스코어 카드

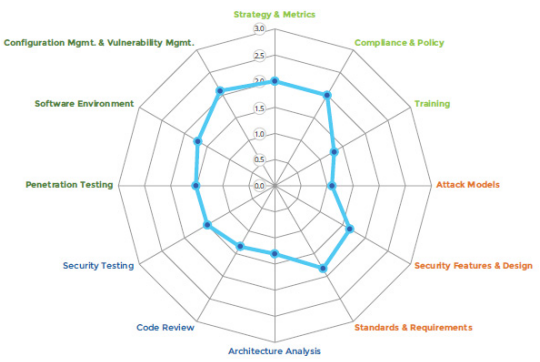
앞서 설명했듯이 BSIMM은 SSF를 구성하는 과정에서 방대한 보안활동에 관한 정보를 활용하였다. 이 때 사용된 것이 스코어카드로 이를 통해 사용자는 식별된 보안활동이 기업별로 수행된 빈도를 알 수 있으며, 또한 각 기업이 공통적으로 수행하는 보안 활동이 무엇인지를 각 수준별, 실무별로 식별할 수 있다.

예로서 [그림 3]의 스코어카드로부터 거버넌스와 인텔리전스 도메인에서 가장 빈번히 발생하는 보안 활동은 '보안게이트 식별과 필요자원의 수집함 [SM1.4]'과 '자료 분류 스키마와 목록을 생성함[AM1.2]' 임을 알 수 있다. BSIMM은 이러한 과정을 통해 파악된 보안활동을 대상으로 12개의 코어 보안활동 목록을 확정G나다[표 3].

### 4.2. 방사형 차트

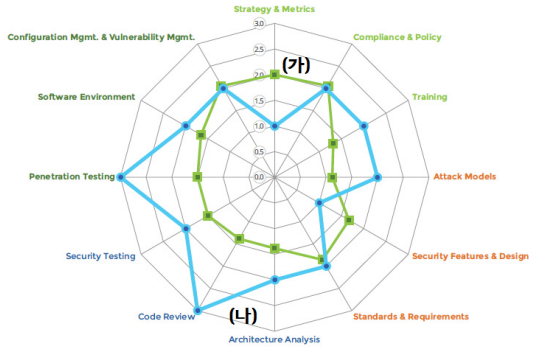
BSIMM의 유용성은 개발 조직이 갖는 보안 성숙도 수준을 비교를 통해 개선되어야 할 부분을 명확히 식별할 수 있다는 점이다. 이를 위해 BSIMM은 코어 활동으로 구성된 방사형 차트를 기준차트로 제공한다 [그림 4]. 핵심활동 기준차트는 각 12개 실무 중 최고점을 가진 활동의 점수를 방사형 기준선에 마킹하여 연결한 차트로서 점수의 분포를 한 눈에 볼 수 있게 한다.

기준차트는 IT조직이 어떤 수준의 보안 성숙도를 갖는지 평가할 때 유용하게 사용된다. 예를 들어 현재 어떤 조직에서 획득한 보안활동의 지표가 [그림 5]의 (나)와 같은 수준으로 파악되었다고 가정하자. 이 차트는 [그림 5]의 (가)와 같은 핵심활동 기준차트를 병치시킴




〔그림 4〕 핵심활동의 기준 차트

으로써 현 조직이 갖는 보안활동의 성숙도 수준을 비교할 수 있다. 이를 통해 부족한 영역이 어디인지 파악하고, 보안활동별 달성 수준을 확인함으로써 보안활동 측면의 강점과 약점을 파악할 수 있다.




〔그림 5〕 기준차트와 실측값의 비교예

## V. 결론 및 시사점

나날이 지능화되는 보안 침해사고에 대해 사후대응 방식의 보안활동에는 한계가 있음을 많은 개발자들이 인식하고 있다. 이를 개선하기 위해서는 보안계획 수립에서부터 이행에 이르는 과정의 많은 보안활동이 개선될 필요가 있다.

국내의 경우 2012년부터 소프트웨어 개발보안 제도를 통해 시큐어코딩 활동을 공공기관 정보시스템 구축에 적용하고 있다. 시큐어코딩은 코딩 표준과 가이드라인을 강제함으로써 보안성이 강화된 소스코드를 확보하기 위한 방법이며, 따라서 개발과정의 보안성을 높이기 위한 적극적인 선제적 대응 방법이다. 이 부분은 이미 BSIMM에서는 SSDL 터치포인트를 통해 핵심활동으로 정의하고 있다.

시큐어코딩을 통해 개발 단계의 소프트웨어 보증활동이 강화되고 있음에 비해 IT 기업이 갖는 보안조직의 역량과 활동은 어떤 수준에 있는지 등에 관한 평가는 상대적으로 미진하다. BSIMM은 이러한 문제에 대해 효과적인 정보를 제공해 줄 수 있는 시도로 판단한다.

마지막으로, BSIMM이 국외 IT 기업의 보안활동 자료를 토대로 구축된 모델이라는 점에서 국내 IT 기업을 대상으로도 적용될 경우 적합한 모델이 될 지에 대해서는 검토가 필요한 사항이다. 이 부분에 대해서는 향후 지속적인 연구가 필요할 것으로 본다.

## 참 고 문 헌

[1] Symantec, Internet Security Threat report, 2011 Trends, Vol. 17, April 2012
[2] Microsoft, Benefits of SDL, https://www.microsoft.com/en-us/SDL/about/benefits.aspx,
[3] Chisholm, G.H, et al, Peer Review of the Trusted Software Methodology, Argonne National Laboratory, 1994
[4] Build Security In, buildsecurityin.us- cert.gov, US-CERT
[5] Software Assurance Landscape. Dept of Homeland Security, August 2006
[6] BSIMM6, http://www.BSIMM.com, 2015
[7] 안전행정부, 소프트웨어 개발보안 가이드, 2013
[8] G. Mcgraw, Software Security: Building Security In, Addison-Wesley, 2006
[9] M Karen, et al. A Twenty-five Year Perspective, CrossTalk, July pp8-15,2013

## 〈저 자 소 개〉

**서 동 수 (Dongsu Seo)**
정회원
1987년 8월 : 중앙대학교 컴퓨터공학 학사
1990년 8월 : 맨체스터대학교, Computation 공학석사
1994년 8월 : 맨체스터대학교, Computation 공학박사
1998～현재 : 성신여자대학교 IT학부 교수
관심분야 : 정보보호, 소프트웨어공학