Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.5, Oct. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.5.909

CIA-Level 기반 보안내재화 개발 프레임워크*

강 수 영,[1†] 김 승 주[2‡]
[1,2]고려대학교 정보보호대학원(대학원생, 교수)

CIA-Level Driven Secure SDLC Framework for Integrating Security into SDLC Process*

Sooyoung Kang,[1†] Seungjoo Kim[2‡]
[1,2]CIST(Center for Information Security and Technologies)
School of Cybersecurity, Korea University(Graduate student, Professor)

요 약

미국 정부는 1970년대 초반부터 모의해킹만으로는 제품의 보안 품질을 향상시킬 수 없다는 것을 인지하기 시작하였다. 모의해킹팀의 역량에 따라 찾을 수 있는 취약점이 달라지며, 취약점이 발견되지 않았다고 해서 해당 제품에 취약점이 없는 것은 아니기 때문이다. 제품의 보안 품질을 향상시키기 위해서는 결국 개발 프로세스 자체가 체계적이고 엄격하게 관리되어야 함을 깨달은 미국 정부는 1980년대부터 보안내재화(Security by Design) 개발 방법론 및 평가·조달 체계와 관련한 각종 표준을 발표하기 시작한다. 보안내재화란 제품의 요구사항 분석 및 설계 단계에서부터 일찍 보안을 고려함으로써 제품의 복잡도(complexity)를 감소시키고, 궁극적으로는 제품의 신뢰성(trustworthy)을 달성하는 것을 의미한다. 이후 이러한 보안내재화 철학은 Microsoft 및 IBM에 의해 Secure SDLC라는 이름으로 2002년부터 민간에 본격적으로 전파되기 시작하였으며, 현재는 자동차 및 첨단 무기 체계 등 다양한 분야에서 활용되고 있다. 하지만 문제는 현재 공개되어 있는 Secure SDLC 관련 표준이나 가이드라인들이 매우 일반적이고 선언적인 내용들만을 담고 있기 때문에 이를 실제 현장에서 구현하기란 쉽지 않다는 것이다. 따라서 본 논문에서 우리는 Secure SDLC를 기업체가 원하는 수준에 맞게 구체화시키는 방법론에 대해 제시한다. 우리가 제안하는 CIA(functional Correctness, safety Integrity, security Assurance)-Level 기반 보안내재화 프레임워크는 기존 Secure SDLC에 증거 기반 보안 방법론(evidence-based security approach)을 접목한 것으로, 우리의 방법론을 이용할 경우 첫째 경쟁사와 자사간의 Secure SDLC 프로세스의 수준 차이를 정량적으로 분석할 수 있으며, 둘째 원하는 수준의 Secure SDLC를 구축하는데 필요한 상세한 세부 활동 및 산출해야 할 문서 등을 쉽게 도출할 수 있으므로 실제 현장에서 Secure SDLC를 구축하고자 할 때 매우 유용하다.

ABSTRACT

From the early 1970s, the US government began to recognize that penetration testing could not assure the security quality of products. Results of penetration testing such as identified vulnerabilities and faults can be varied depending on the capabilities of the team. In other words none of penetration team can assure that “vulnerabilities are not found” is not equal to “product does not have any vulnerabilities”. So the U.S. government realized that in order to improve the security quality of products, the development process itself should be managed systematically and strictly. Therefore, the US government

Received(07. 07. 2020), Accepted(08. 14. 2020)
* 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00532,고등급(EAL6 이상) 보안 마이크로커널 개발)
† 주저자, bbang814@gmail.com
‡ 교신저자, skim71@korea.ac.kr(Corresponding author)

began to publish various standards related to the development methodology and evaluation·procurement system embedding "security-by-design" concept from the 1980s. Security-by-design means reducing product's complexity by considering security from the initial phase of development lifecycle such as the product requirements analysis and design phase to achieve trustworthiness of product ultimately. Since then, the security-by-design concept has been spread to the private sector since 2002 in the name of Secure SDLC by Microsoft and IBM, and is currently being used in various fields such as automotive and advanced weapon systems. However, the problem is that it is not easy to implement in the actual field because the standard or guidelines related to Secure SDLC contain only abstract and declarative contents. Therefore, in this paper, we present the new framework in order to specify the level of Secure SDLC desired by enterprises. Our proposed CIA (functional Correctness, safety Integrity, security Assurance)-level-based security-by-design framework combines the evidence-based security approach with the existing Secure SDLC. Using our methodology, first we can quantitatively show gap of Secure SDLC process level between competitor and the company. Second, it is very useful when you want to build Secure SDLC in the actual field because you can easily derive detailed activities and documents to build the desired level of Secure SDLC.

Keywords: Security by Design, Secure SDLC(Secure Software Development Life Cycle), SDLC(Software Development Life Cycle), SDL(Security Development Lifecycle), Security Engineering

I. 서 론

미국 정부는 이미 1970년대부터 모의해킹을 통한 제품의 개선(Penetrate and Patch approach)은 한계가 있음을 깨닫는다. 왜냐하면 모의해킹팀의 역량 및 수행 기간에 따라 취약점 유무가 결정되며, 취약점이 발견되지 않았다고 해서 대상 제품에 취약점이 없음을 보장하는 것은 아니기 때문이다[1-4].

이후 미국은 1980년대부터 제품의 보안 품질을 향상시키기 위해서는 취약점을 찾고 패치하는 것에만 의존할 것이 아니라 개발 프로세스 자체가 체계적이고 엄격하게 관리되어야 함을 인식하고, 보안내재화(Security by Design)와 관련된 각종 개발 방법론 및 평가·조달 체계 관련 표준들을 발표하기 시작한다[5-8].

여기서 보안내재화란 제품의 요구사항 분석 및 설계와 같이 개발 초기 단계부터 보안을 고려함으로써 제품의 복잡도를 감소시키고 결과적으로는 제품의 신뢰성(trustworthiness)[1]을 달성하는 것을 일컫는 것으로[10-14], 이러한 보안내재화 철학이 담긴 개발 프로세스를 "Secure SDLC(Secure S/W Development Life Cycle)" 또는 "보안공학 프로세스(Security Engineering Process)"라고 부른다[9,10,15-21].

산업계에서는 Microsoft와 IBM이 2002년부터 Secure SDLC에 관심을 갖고 본격적으로 민간에 전파하기 시작하였다[22,23]. 이 중 Microsoft는 자체 수립한 Secure SDLC, 일명, "MS-SDL(Microsoft Security Development Lifecycle)"을 2007년 Windows VISTA 개발부터 본격 적용하기 시작했으며, 현재 가장 앞선 기술력을 보유하고 있다[24,25].

최근 보안내재화 개발 방법론은 전산 시스템뿐만이 아니라 자동차에서 첨단 무기 체계에 이르기까지 다양한 응용 분야에서 널리 활용되고 있는 것이 사실이다[25-38]. 그러나 현재 공개되어 있는 각종 표준이나 가이드라인들은 추상적인 보안 활동 목록들만 나열하고 있기 때문에 실제 현장에서 Secure SDLC를 구축하는 것은 쉽지 않다[39-49].

따라서 본 논문에서 우리는 Secure SDLC를 기업체가 원하는 수준에 맞게 구체화하기 위한 방법론을 제안한다. 우리가 제안하는 방법론은 기존의 Secure SDLC 관련 표준에 증거 기반 보안 방법론(evidence-based security approach)[2]을 접목한 것으로[50-54], 우리의 방법론을 이용할 경우 첫째 경쟁사와 자사간의 Secure SDLC 프로세스의 수준 차이를 정량적으로 분석할 수 있으며, 둘째 원하는 수준의 Secure SDLC를 구축하는데 필요한 상세한 세부 활동 및 산출해야 할 문서 등을 쉽게 도출할 수 있으므로 실제 현장에서 Secure SDLC를 구축하고자 할 때 매우 유용하다.

1) security, availability, reliability, safety 등을 함께 일컫는 말. 임베디드 시스템(embedded system) 환경에서 이들은 상호 독립적이 아닌 상호 의존적 관계이므로, 초기 요구사항 분석 단계에서부터 이 4가지 요소들을 한꺼번에 고려해 복잡도(complexity)를 최소화하려는 노력이 무엇보다 중요함.

2) 신뢰된 결과를 도출하기 위하여, 검증된 과학적 증거들을 활용하는 접근 방법을 의미함. 수집된 증거들의 출처를 검증하여 증거들 간 추적성을 제공할 수 있음.

우리는 이를 위해 각 분야에서 대표적으로 활용되고 있는 Secure SDLC 표준 및 가이드라인 10종을 선정하였다. 이후 이 프로세스들을 10단계로 정규화·일반화 하였으며, 각 단계별로 수행해야 하는 보안 활동 66개를 도출하고, 끝으로 대표적인 증거 기반 보안 방법론 표준들을 활용해 상세한 세부 보안 활동 및 산출 문서 등을 이끌어 내었다.

지금껏 Secure SDLC를 구체화하기 위한 시도들이 많이 있었으나, 본 연구와 같이 상세한 방법론을 제공한 적은 없었다. 또한 Secure SDLC 프로세스의 수준을 정량적으로 분석하거나 다른 조직과의 갭을 분석해주는 시도는 최초이기 때문에 본 연구결과는 Secure SDLC를 원하는 실제 현장에서 매우 효과적으로 사용될 수 있을 것으로 기대된다.

II. 관련 연구

본 장에서는 Secure SDLC와 관련한 최근 연구동향을 살펴본다. 선정한 대상은 (i) Secure SDLC를 구체화 시키는 방법론과 관련된 학술 논문, (ii) Secure SDLC와 관련된 특허, (iii) Secure SDLC 표준, (iv) 증거 기반 보안 방법론이다.

이후 각 절에서는 이 4가지 항목들을 선정한 기준과 심층 분석한 내용을 설명하고, 분석된 결과는 〔Table 1〕에서 정리하여 보여준다.

2.1 관련 논문

우리는 Secure SDLC를 상세화 시키는 방법론과 관련된 학술 논문을 검색하기 위하여 대표적으로 알려진 (i) ACM〔55〕, (ii) Elsevier〔56〕, (iii) IEEE〔57〕, (iv) Scopus〔58〕, (v) Springer〔59〕의 5가지 출판사를 선정하였다. 또한 'SDL', 'SDLC', 'SSDLC', 'secure development lifecycle', 'secure SDLC' 키워드를 가지며, 'integrating', 'mapping'과 같이 통합과 관련된 활동을 주제로 하는 논문들을 검색하였고, 추가적인 정보가 필요한 경우 google scholar〔60〕를 활용하였다. 논문들을 검색한 결과, ACM 647건, Elsevier 24건, IEEE 8건, Scopus 1건, Springer 62건이 검색되었으나, 이 중 상세한 내용이 포함되지 않거나 중복된 논문들을 제거한 후 최종적으로 학술 논문 11편을 선별하여 심층 분석하였다〔39-49,61-72〕.

총 11건의 학술 논문을 분석한 결과, 6건의 논문들에서는 제품 개발 및 조직 운영에 있어 보안을 통합하기 위해 관련 표준들과의 매핑을 수행하였으며, 적용 가능한 보안 기법들을 조사하여 실무에 활용하고자 하였다. 하지만 이들은 모두 너무 추상적인 상위 수준의 매핑을 수행하였으며, 상세한 세부 활동이나 산출해야 하는 문서 등은 제시하고 있지 않았다. 또한 나머지 5건의 논문들에서는 요구사항 분석 단계 및 설계 단계 등과 같이 SDLC 프로세스 중 극히 일부 단계만을 다루고 있는 것으로 확인되었다. 이러한 결과로 이루어 볼 때 Secure SDLC의 세부 보안 활동 및 산출해야 할 문서 등을 고려하여 전체 프로세스를 상세화 시키는 방법론이 필요하다는 것을 확인할 수 있었다〔39-49〕.

2.2 관련 특허

Secure SDLC의 중요성이 증가함에 따라 관련된 특허들이 발표되었으며, 관련 특허를 조사하고 선별한 결과 7건을 선정하여 심층 분석하였다〔73-79〕.

세계적인 은행인 BoA(Bank of America)의 특허는 7건의 특허 중 유일하게 Secure SDLC 전체 프로세스를 다루고 있었다〔73〕. 해당 특허는 보안 설계를 통해 취약점을 최소화하고, 위험 수준 별 Secure SDLC 프로세스를 구축하는 내용을 담고 있다. 위험 수준은 3단계(High, Medium, Low)로 분류되며, High의 경우 아키텍처 및 설계 검토를 통해 내부 구조를 파악한 후 취약점을 모두 스캔하고 분석한다. Medium의 경우 내부 구조를 파악한 후 빈번하게 발생하거나 위험도가 낮은 취약점만을 분석한다. Low의 경우 내부 구조를 파악하지 않은 채 해킹을 시도한다. 하지만 해당 특허의 경우 위험 수준을 산정하는 기준이 명확하지 않고, 각 단계에서 수행하는 세부 보안 활동 및 산출해야 하는 문서 등에 대한 정보가 충분히 제공되지 않는다.

BoA 특허를 제외하고 남은 6건 중 4건의 특허는 요구사항 분석 단계 및 설계 단계 등과 같이 일부 단계에서만 활용할 수 있는 특허임을 확인하였고〔74,75,78,79〕, 남은 2건의 특허는 보안 모듈 및 보안 장치와 같이 특정 제품에서 활용할 수 있는 특허임을 확인하였다〔76,77〕.

2.3 Secure SDLC 표준

그동안 다양한 정부 조직과 기업체들이 보안 제품 개발 시 보안내재화 철학을 적용한 Secure SDLC

프로세스를 통해 제품의 보안 품질을 향상시켜왔다[80]. 산업계에서는 Microsoft와 IBM이 2002년부터 Secure SDLC에 관심을 갖고 민간에 전파하기 시작했고, 이 중 Microsoft는 자체 수립한 Secure SDLC인 MS-SDL을 2007년 Windows VISTA 개발부터 본격적으로 적용하기 시작했다[22-25].

이후 NIST는 Secure SDLC의 대상을 소프트웨어에서 하드웨어까지 확장하고, third-party가 개발한 제품을 구매하는 획득 단계 및 매체를 안전하게 소거하는 폐기 단계를 추가한 NIST SSDLC(National Institute of Standards and Technology Security considerations in the System Development Life Cycle) 표준을 2008년에 발표하였다[26]. 싱가포르 정부도 CSA SDF(Cyber Security Agency of singapore Security by Design Framework)를 자체 개발하고 2017년에 표준을 발표하였다[27]. 이와 같이 정부에서 개발한 Secure SDLC 표준에는 획득 및 폐기 단계를 포함하여 제품 구매와 프로세스 종료 이후의 보안도 고려하였다. 하지만 특정 기법이나 도구 등에 대한 구체적인 정보들은 제공하지 않는다.

Secure SDLC의 활용 사례가 증가하면서 모범사례를 기반으로 한 Secure SDLC도 개발되었다. McGraw Touchpoints는 모범사례를 기반으로 7개의 중요 보안 활동을 중점적으로 수행하며[29], OWASP CLASP(OWASP Comprehensive, Lightweight Application Security Process)는 5개의 중요 관점에서의 보안 활동을 수행한다[30]. 또한 비영리조직인 SAFECode는 모범사례, 참고자료, 각 단 계별 산출해야 하는 문서의 예시 템플릿 등과 같이 실용적인 자료들을 제공한다[28]. 하지만 모범사례를 활용할 경우, 모범사례를 보여준 조직과 특징이 유사할 경우에만 활용 가능하기 때문에 실제 활용 시 제약사항이 발생한다.

또한 이러한 Secure SDLC가 잘 구축되어있는지를 판별하는 척도인 보안 성숙도를 측정하는 모델도 있다. Cigital에서 개발한 BSIMM(Cigital Building Security In Maturity Model)은 이해 관계자들의 협력하는 정도에 따라 보안 성숙도를 측정하고[31], OWASP에서 개발한 SAMM(OWASP Software Assurance Maturity Model)은 보안 실무 목표를 달성하기에 적합한 정도에 따라 보안 성숙도를 측정한다[32].

이 외에도 미국 정부는 2013년에 발표된 RMF (Risk Management Framework) 표준을 통해 전산 시스템의 개발 및 평가·조달 체계를 관리하였으며, 2015년에 발표된 국방부 사이버 전략(The Department of Defense Cyber Strategy)에 따라 RMF 대상 범위가 전산 시스템에서 첨단 무기체계로 확대되었다[33,81].

또한 자동차 분야에서도 Secure SDLC 표준이 발표되었다. 유럽경제위원회인 UNECE는 사이버보안 규제를 제정하고 있으며, 2022년부터 이를 준수하지 않은 자동차를 유럽에 수출할 수 없음을 공표하였다[36]. 사이버보안 규제에서는 자동차를 개발하는 전체 프로세스 동안 사이버보안을 고려하도록 요구하고 있으며[35,37,38] 자동차에 대한 사이버보안 가이드라인은 SAE J3061[34]에 정의되어 있다.

앞서 설명한 10건의 Secure SDLC 표준 및 가이드라인을 분석한 결과, 대부분 세부 보안 활동 및 산출해야 할 문서 등에 대한 정보가 부족했고, SAFECode와 같이 산출해야 할 문서 템플릿이 공개된 경우에도 그것이 요구사항 분석 단계에 집중되어 있음을 확인할 수 있었다.

2.4 증거 기반 보안 방법론

앞서 설명한 바와 각 분야에서 대표적으로 활용되고 있는 Secure SDLC 표준 및 가이드라인들은 추상적인 보안 활동 목록들만을 나열하고 있기 때문에 실제 현장에서 Secure SDLC 구축 시 활용하기가 쉽지 않다. 본 논문에서는 이러한 문제들을 해결하기 위해 증거 기반 보안 방법론을 활용하여 Secure SDLC 표준 및 가이드라인들을 구체화 시키고자 한다. 증거 기반 보안 방법론을 활용할 경우 수집된 증거들의 출처를 검증하기 때문에 증거들 간 추적성을 확보할 수 있다. 이러한 특징을 가지는 대표적인 표준은 ISO/IEC 15408 CC(Common Criteria)로, 보안 기능이 탑재된 IT 제품에 대한 보안성(security) 및 보증(assurance) 수준을 평가하는 국제 표준이다. CC에는 EAL(Evaluation Assurance Level)이라는 보증 수준이 7단계로 정의되어 있으며, 보증 수준이 높을수록 더 엄격한 기준을 준수하고 많은 산출물이 요구된다[82].

Table 1. Related works

Classification	Year	Author/Title	Explanation	Ref.
Papers	2002	Younghwa Lee	A study that integrates SDLC standard (IEEE/EIA 12207) and security engineering	[39]
	2007	Daniel Mellado	A study that integrates CC standards (ISO/IEC 15408) and Security Requirements Engineering in the requirements analysis phase of Secure SDLC	[40]
	2007	Lynn Futcher	A study that integrates SDLC and security design standards (ISO 7498-2) that can be used at the design phase	[41]
	2012	Razieh Sheikhpour	A study that integrates the lifecycle of IT services and ISMS standards (ISO/IEC 27001)	[42]
	2013	Tahereh Nayerifard	A study that integrates the CC standard (ISO/IEC 15408) and the ISMS standard (ISO/IEC 27001)	[43]
	2015	Siwar Kriaa	A study that integrates the process evaluation standard (ISO/IEC 15504) and the information security management practice guideline (ISO/IEC 27002)	[44]
	2015	Antoni Lluı´s Mesquida	A study that integrates the process evaluation standard (ISO/IEC 15504) and the information security management practice guideline (ISO/IEC 27002)	[45]
	2017	Nabil M. Mohammed	A study that investigates all security techniques applicable to SDLC requirements analysis, design, and implementation stages	[46]
	2017	Mary-Luz Sánchez-Gordón	A study that incorporates Secure SDLC best practices and SDLC standards for small organizations (ISO/IEC 29110)	[47]
	2018	Patrick Morrison	A study that investigated all the security factors to be measured, such as the number of vulnerabilities and the number of attack surfaces in Secure SDLC	[48]
	2020	Valentina Casola	A study that integrates automated risk management measures into the design phase of Secure SDLC	[49]
Patents	2013	Bank of America	Patent for the entire risk-based Secure SDLC process	[73]
	2012	Microsoft	Patent for threat modeling method and threat modeling tool for requirements analysis and design phase of Secure SDLC	[74]
	2012	Lawrence Wilcock	Patent for development automation technique that can be used in the implementation phase of SDLC	[75]
	2018	Nokia	Patent for the life cycle of the Subscriber Identification Module (SIM)	[76]
	2018	Schilder Marius	Patent for security level setting and access control in SDLC of security device	[77]
	2019	Nishchal Bhalla	Patent for the risk identification method for requirements analysis and design phase of Secure SDLC	[78]
	2020	Vincent Benjamin	Patent for account and authority management method for the operation stage of Secure SDLC	[79]

Table 1. Continued

Classification	Year	Author/Title	Explanation	Ref.
Secure SDLC standards	2004~	Microsoft SDL	Secure SDLC for software developed by Microsoft	[25]
	2008~	NIST SSDLC	Secure SDLC for systems developed by NIST	[26]
	2017~	CSA SDF	Secure SDLC for systems developed by the Singaporean government	[27]
	2008~	SAFECode	Secure SDLC based on security development best practices and reference materials	[28]
	2004~	McGraw Touchpoints	Secure SDLC centered on 7 important security activities developed by McGraw	[29]
	2006~	OWASP CLASP	Lightweight Secure SDLC that performs verification of five key points developed by OWASP	[30]
	2009~	Cigital BSIMM	Four-level security maturity model according to the degree of stakeholder collaboration	[31]
	2009~	OWASP SAMM	Three-level security maturity model according to the degree suitable for security practice	[32]
	2013~	NIST RMF	Secure SDLC applying the concept of risk management to the development and evaluation and procurement system of the US computer system and advanced weapons	[33]
	2016~	SAE J3061	Automotive cybersecurity guidelines considering both functional safety and security of automobiles	[34]
Evidence-based security approach	1999~	ISO/IEC 15408	CC(Common Criteria)	[82]
	1995~	ISO/IEC 27001	ISMS(Information Security Management System)	[83]
	2013~	ISO/IEC 27701	PIMS(Privacy Information Management System)	[84]

또한 대표적인 증거 기반 보안 방법론 표준은 ISO/IEC 27001 ISMS(Information Security Management System)로, 정보보호 관리체계에 대한 국제 표준이다[83]. 이와 더불어 ISO/IEC 27701 PIMS(Privacy Information Management System)는 개인정보보호를 다루는 국제 표준으로, 두 표준은 조직과 개인정보보호를 점검하기 위한 세부 항목들로 구성되어 있다[84].

앞서 설명한 3건의 증거 기반 보안 방법론 표준들을 분석한 결과, 제품, 조직, 개인정보보호 관점의 보안 활동들을 수행하기 위한 세부 활동들이 상세하게 정의되어 있었으며, 특히 CC에는 산출되어야 할 문서의 요구조건들이 구체적으로 명시되어 있어 실제 현장에서 Secure SDLC를 구축하고자 할 때 매우 유용하다.

III. CIA-Level 기반 보안내재화 개발 프레임워크

앞서 설명한 바와 같이 각 분야에서 대표적으로 활용되고 있는 Secure SDLC 표준 및 가이드라인들은 매우 일반적이고 선언적인 내용들만을 담고 있기 때문에 이를 실제 현장에서 활용하기 어렵다.

따라서 본 장에서는 기업체가 원하는 수준에 맞게 Secure SDLC를 구체화하기 위한 방법론을 제안한다. 우리가 제안하는 방법론은 공개되어 있는 대표적인 Secure SDLC 표준 및 가이드라인에 증거 기반 보안 방법론을 접목한 것으로, CIA-Level 기반 보안내재화 개발 프레임워크라 명명한다. 여기서 CIA-Level이란 정확성(functional Correctness), 안전성(safety Integrity), 보안성(security Assurance) 수준을 의미하며, 수행하고 있는 Secure SDLC 프로세스 수준에 따라 CIA-Level을 산정할 수 있다.

우리는 가장 먼저 각 분야에서 대표적으로 활용되고 있는 Secure SDLC 표준 및 가이드라인 10종을 선정한 후 비교·분석하여 10개의 단계로 정규화·일반화 하였다. 그리고 10개의 단계별로 수행해야 하는 보안 활동 66개를 도출하고, 각 보안 활동들은 증거 기반 보안 방법론 표준들을 활용하여 상세한 세부 보안 활동 및 산출해야 하는 문서 등을 도출하였다.

이러한 방법론을 통해 동작하는 CIA-Level 기반 보안내재화 개발 프레임워크는 두 가지 중요 기능을 제공한다. 첫째, 자사와 경쟁사 간 갭분석을 통해 Secure SDLC 프로세스의 수준 차이를 정량적으로 분석할 수 있다.

둘째, 기업체가 원하는 수준의 Secure SDLC를 구축하는데 필요한 상세한 세부 보안 활동 및 산출해야 하는 문서 등을 쉽게 도출하여, 실제 현장에서의 활용이 용이하도록 한다.

앞서 설명한 CIA-Level 기반 보안내재화 개발 프레임워크는 [Fig 1]과 같이 4가지 구성요소로 구성되며, 각 구성요소의 역할은 [Table 2]와 같다. 이후 각 절에서는 구성요소 별 수행하는 기능에 대해 구체적으로 설명한다.

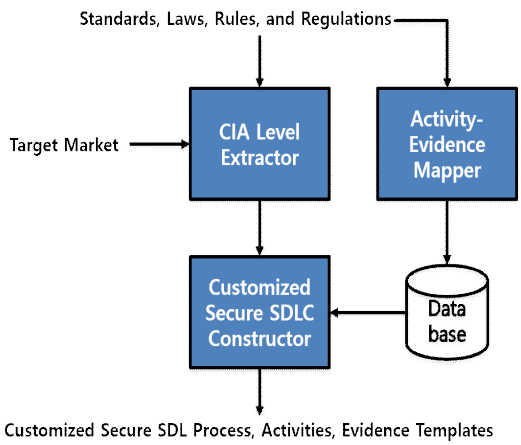

Fig. 1. CIA-Level Driven Secure SDLC Framework

Table 2. Components

Component	Role
Activity-Evidence Mapper	Mapping of Secure SDLC standards and guidelines to evidence-based security methodology standards
Database	Store details such as mapping results and artifacts
CIA-Level Extractor	CIA-level extraction and gap analysis of companies and competitors
Customized Secure SDLC Constructor	Provide Secure SDLC process, security activity, detailed security activity, artifact template, etc.

3.1 Activity-Evidence Mapper

Activity-Evidence Mapper는 기업체가 원하는 수준에 맞게 Secure SDLC 프로세스를 구체화하기 위하여 Secure SDLC 표준 및 가이드라인과 증거 기반 보안 방법론 표준과 매핑하는 역할을 수행한다. 현재 Activity-Evidence Mapper는 2장에 서술한 바와 같이 각 분야에서 대표적으로 활용되고 있는 Secure SDLC 표준 및 가이드라인 10종을 대상으로 한다. Activity-Evidence Mapper에서 활용하는 10종의 Secure SDLC는 앞서 언급하였듯이, Microsoft SDL, NIST SSDLC, CSA SDF, OWASP CLASP, McGraw Touchpoints, SAFECode, OWASP BSIMM, OWASP SAMM, NIST RMF, SAE J3061이며, 향후 공신력 있는 Secure SDLC가 학술 및 산업계에 소개될 경우 관련 데이터를 추가하여 프레임워크를 확장할 수 있다.

Activity-Evidence Mapper는 10종의 각 단계들을 비교·분석하여 10개의 단계로 정규화·일반화하였고, 각 단계에서 수행해야 하는 보안 활동을 모두 합한 후 중복되는 보안 활동은 제거하여 66개의 보안 활동을 도출함으로써 하나의 Secure SDLC로 통합하였다. 정규화·일반화를 수행하여 도출된 10개의 단계와 66개의 보안 활동 및 산출물 목록은 [Table 3]과 같다.

산출물들은 기존 소프트웨어 공학에서 산출하는 문서를 기반으로 하였으나, 보안이 고려됨에 따라 기존 산출물에서 보안과 관련된 항목들이 추가되었다. 첫째, 보안 교육 단계에서는 교육 일정, 교육을 받아야 하는 대상, 교육 커리큘럼 등에 대한 계획과 해당 교육을 이수한 사람의 명단을 포함하였으나, 보안이 고려됨에 따라 기존 교육에서 보안과 관련된 교육 내용이 추가되어야 한다. 둘째, 착수 및 계획 단계에서는 현행 프로세스 분석서 및 현행 시스템 분석서는 기존과 동일하게 산출되고, 프로젝트 일정, 프로젝트 범위, 역할 등에 대한 계획을 다루는 프로젝트 계획서와 사용자의 요구사항을 명세한 요구사항 정의서는 보안을 고려해야 하는 범위를 산정하고, 보안 기능 요구사항을 추가적으로

Table 3. Normalized and generalized Secure SDLC stage and security activities

Phase	No.	Security activity	Artifact
1.Security Training	1	Basic security training	-Security training plan -Training attendee list
	2	Advanced security training	
	3	Plan training schedules	
2.Initiation	4	Project categorization	-Current process analysis -Current system analysis -Project plan -Software Requirements Specification
	5	Role identification	
	6	Project tools selection	
	7	Security requirements source identification	
	8	Minimum quality level definition	
	9	Prepare compensation system for handling security issues	
	10	Plan project schedule	
	11	Security goals setting by field	
	12	Verifying consistency & completeness of goals	
3.Requirements Analysis	13	Estimating scope of project security analysis	-Impact assessment -Interface definition
	14	Impact assessment for privacy	
	15	Impact assessment for business	
	16	Impact assessment for safety	
	17	Existing software assessment	
	18	Functional requirements elicitation	
	19	Security requirements elicitation	
	20	Conformity & conflict check on requirements by field	
	21	Verifying requirements based on security goals	
4.Acquisition	22	Plan third-party components acquisition	-Acquisition confirmation document
	23	Requirements definition for third-party components	
	24	Assessment & test for third-party components	
5.Design	25	Functions & design specification	-Software design specification -Software architecture design and System architecture design specification -Integrated test plan and integrated test scenario
	26	Compliance with design best practices and principles	
	27	Structural design for the integration process	
	28	Asset identification	
	29	Create data flow diagram	
	30	Threat elicitation	
	31	Attack Library Collection	
	32	Risk analysis by field	
	33	Mitigation elicitation by field	
	34	Privacy analysis	
	35	Usecase and misuse case identification	
	36	Verifying design based on requirements	
6.Implemen-tation	37	Compliance with secure coding guidelines	-Source code -Unit test plan and test scenario -Unit test results
	38	Creation for deployment guide document and tools	
	39	Implementation verification according to design	
7.Verification	40	Static analysis	-Integrated/system/acquisition test plan and test scenario -Integrated/system/acquisition test results -Vulnerability analysis
	41	Automated static analysis	
	42	Dynamic analysis	
	43	Integration & acceptance test	
	44	Penetration test	
	45	Review and update for threat model	
	46	Review for minimum quality and security level	
	47	Review and update for security documents	

Table 3. Continude

Phase	No	Security activity	Artifact
8.Release	48	Final security review	-Rehearsal plan and rehearsal result -Release request -Emergency incident response plan -Emergency accident response result
	49	Final privacy review	
	50	Requirements elicitation for production	
	51	Production procedure determination	
	52	Verification of production	
	53	Accident response planning	
	54	Security review for deployment procedure	
9.Operation	55	Monitoring planning	-Preparation result -Operator instructions -User guide -Vulnerability Response Plan -Vulnerability patch result
	56	Continuous monitoring	
	57	Vulnerability report	
	58	Vulnerabilities assessment	
	59	Solution establishment	
	60	Vulnerability disclosure & patch/update	
	61	Configuration management after release	
10.Disposal	62	Transfer & disposal procedure planning	-System implementation plan and system implementation result
	63	Important information disposal	
	64	Media Erase	
	65	Hardware and software disposal	
	66	System shutdown	

포함되어야 한다. 셋째, 요구사항 분석 단계에서는 기존에 사업 영향도만 분석했던 반면에 보안 및 프라이버시 영향도를 추가적으로 분석한 영향평가서가 산출되어야 하며, 기존 인터페이스에 대한 식별자, 목적, 사용 방법 등을 명세한 인터페이스 정의서는 보안 기능을 수행하는 정도(직접 수행, 간접 수행 등)에 따라 인터페이스를 분류하여 명세되어야 한다. 넷째, 획득 단계에서는 third-party가 개발한 구성요소에 대해 검수하는 획득 검수서가 산출되어야 한다. 다섯째, 설계 단계에서는 클래스, 컴포넌트, 데이터베이스 등에 대한 설계를 명세하는 설계 명세서가 보안 기능을 수행하는 정도에 따라 모듈을 분류하여 명세되어야 하며, 소프트웨어 아키텍처 설계서 및 시스템 아키텍처 설계서는 보안 정책에 따라 보안 기능이 정확하게 동작하며 우회 경로가 없음을 입증하도록 명세되어야 한다. 또한 모듈을 설계하며 모듈 간 통합을 검증하기 위한 통합 테스트 계획서 및 테스트 시나리오는 보안 모듈 간 통합 내용을 추가적으로 포함해야 한다. 여섯째, 구현 단계에서는 보안을 고려한 보안 코딩 가이드라인을 준수하여 개발한 소스코드가 산출되어야 하며, 보안 모듈을 테스트하기 위한 단위 테스트 계획서 및 테스트 시나리오가 산출되어야 한다. 일곱째, 검증 단계에서는 보안을 고려한 통합, 시스템, 인수 테스트에 대한 테스트 계획서 및 테스트 시나리오가 산출되어야 하고, 기존 개발 프로세스에서는 수행하지 않았던 취약성 분석을 수행함에 따라 취약성 분석서가 산출되어야 한다. 여덟째, 배포 단계에서는 품질을 최종 검토했던 반면 보안 및 프라이버시에 대한 최종 검토 내용이 추가된 리허설 계획서 및 리허설 결과서가 산출되어야 하며, 기존 배포 과정에 물리적 보안 대책과 논리적 보안 대책이 모두 고려된 배포 요청서가 산출되어야 한다. 또한 배포 이후 비상 사고를 대비하여 비상 사고 대응 계획서와 비상 사고 대응 결과서가 산출되어야 한다. 아홉째, 운영 단계에서는 보안상 안전한 설치 및 운영 지침과 공격자에게 정보를 제한하기 위한 오류 메시지, 운영환경에서 충족되어야 하는 보안 대책 등이 포함된 설치 결과서, 운영자 지침서, 사용자 지침서가 산출되어야 한다. 또한 운영 단계에서 발생한 취약점에 대해 대응 계획서 및 패치 결과가 산출되어야 한다. 마지막 폐기 단계에서는 프로젝트 수행에 있어 시스템 사용에 대해 계획하고 이행하며, 시스템을 이관하거나 폐기하는 것에 대해 계획하고 이행하는 시스템 이행계획서와 이행결과서가 산출되어야 한다.

앞서 설명한 바와 같이 66개의 보안 활동들이 대표적인 증거 기반 보안 방법론 표준인 CC와 매핑되어 산출물까지 모두 도출되면, 조직이나 개인정보에 대한 보안 활동은 ISMS 및 PIMS로 보완된다.

CC는 보안성을 평가하기 위한 2부의 보안요구사

항과 보증 수준을 평가하기 위한 3부 및 CEM의 보증요구사항을 포함한다. 보증요구사항은 요구사항 분석, 개발, 설치 및 운영, 생명주기 지원, 테스트, 취약점 분석 단계로 분류되며, Secure SDLC의 단계와 유사하기 때문에 대부분의 보안 활동들을 커버할 수 있다. 또한 보증요구사항들은 각 단계별로 수행해야 하는 세부 항목들을 포함하고 있으며, 산출해야 할 문서들이 충족해야하는 상세한 요구 조건들이 포함되어 있어, 이를 활용하여 상세 세부 보안 활동들을 정의하고 산출해야 할 문서들의 템플릿을 생성한다. 이에 따라 Activity-Evidence Mapper는 3건의 증거 기반 보안 방법론 표준 중 가장 먼저 CC와의 매핑을 시작하며, 매핑을 하기 위해 CC 3부 및 CEM에 정의되어 있는 63개의 보증요구사항 컴포넌트를 활용한다.

하지만 66개의 보안 활동 중 CC에서는 다루지 않고, 조직에서 반드시 수행해야 하는 항목들이 존재한다. 예를 들어 보안 및 개인정보보호 관련 교육, 보안 교육과 관련된 일정 계획, 보안 교육 로드맵 수립 등은 조직 차원에서 수행되고 관리되어야 한다. 이와 같이 조직 차원에서 수행되어야 하는 보안 활동들은 ISMS에 정의되어 있는 104개의 세부 점검 항목을 활용하여 매핑된다.

또한 66개의 보안 활동 중 CC와 ISMS에서는 다루지 않는 항목들도 존재한다. 예를 들어 개인정보보호 영향평가의 경우 개인정보 침해가 우려되는 위험 요인을 분석 및 개선함으로써 사고를 방지 할 수 있다. 따라서 이와 같이 개인정보보호 관점에서 수행되어야 하는 보안 활동들은 PIMS에 정의되어 있는 54개의 세부 평가 항목을 활용하여 매핑된다.

결과적으로 Activity-Evidence Mapper는 CC

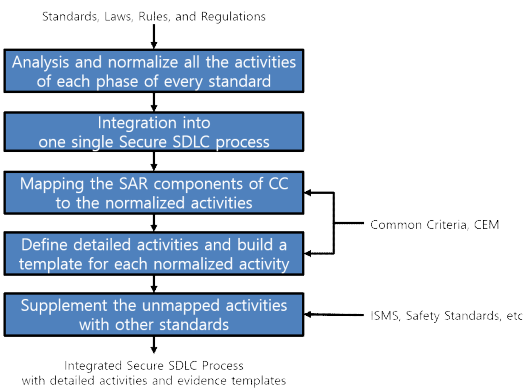

Fig. 2. Activity-Evidence Mapper

Table 4. Number of details security activities

Standard	Classification	CT.
ISO/IEC 15408 - CC	1. Security Target	10
	2. Development	19
	3. Guidance	2
	4. Life cycle support	18
	5. Tests	9
	6. Vulnerability assessment	5
CC total	63 assurance requirements components	
ISO 27001 - ISMS	1. Security policy	6
	2. Organization of information security	4
	3. Outsider security	3
	4. Information Asset Classification	3
	5. Security training	4
	6. Human resource security	5
	7. Physical security	9
	8. System development security	10
	9. Encryption control	2
	10. Access control	14
	11. Operation security	22
	12. Information security incident management	7
	13. IT incident recovery	3
	14. Information security management	12
ISMS total	104 detailed check items	
ISO/IEC 27701 - PIMS	1. PIMS of the target organization	8
	2. PIMS of the target system	6
	3. Step-by-step protection method of the privacy	12
	4. Technical protection method of the target system	19
	5. Privacy protection when using specific IT technology	9
PIMS total	54 detailed assessment items	
Total	221 detailed security activities	

의 보증요구사항 컴포넌트 63개, ISMS의 세부 점검 항목 104개, PIMS의 세부 평가 항목 54개를 합하여 총 221개의 세부 항목들과 66개의 보안 활동 간 매핑을 수행함으로써, 상세한 세부 보안 활동

들을 정의하고 산출해야 할 문서 등을 생성한다. 이와 같이 Activity-Evidence Mapper에서 수행하는 기능은 〔Fig 2〕와 같으며, 활용하는 증거 기반 보안 방법론들의 세부 항목 수는 〔Table 4〕와 같다.

3.2 Database

Database는 앞서 설명한 Activity-Evidence Mapper가 수행한 매핑 결과 및 산출해야 할 문서와 같은 상세한 세부 정보들을 저장하는 저장소이다. Database 스키마는 4개의 테이블로 구성되어 있다. 첫 번째 테이블에는 정규화·일반화된 10개의 Secure SDLC 단계들이 저장되어 있으며, 두 번째 테이블에는 각 단계 별로 수행해야 하는 66개의 보안 활동들이 저장되어 있다. 세 번째 테이블에는 상세 세부 보안 활동 목록 및 설명들이 저장되어 있으며, CC 3부 및 CEM에 정의되어 있는 보증요구사항 컴포넌트 63개, ISMS에 정의되어 있는 세부 점검 항목 104개, PIMS에 정의되어 있는 세부 평가 항목 54개를 합하여 총 221개의 세부 보안 활동들이 저장되어 있다. 네 번째 테이블에는 CC 및 CEM의 보증요구사항 컴포넌트와 매핑한 결과인 산출해야 할 문서들의 템플릿이 저장되어 있다. 산출해야 할 문서들에는 Secure SDLC 프로세스 중 추적성을 확보하기 위해 증거들의 출처를 포함하는 매핑표가 포함되어 있다. 예를 들어 Secure SDLC의 '요구사항 분석 단계'의 '보안 요구사항 도출' 보안 활동의 경우 CC의 'ASE_REQ.1 보안요구사항' 보증요구사항 컴포넌트 및 ISMS의 '8.1.1 보안요구사항 정의' 세부 점검 항목과 매핑된다. 이 경우 요구사항 분석 단계에서 산출해야 하는 문서는 '보안이 고려된 SRS(Software Requirement Specification)'로, 해당 템플릿에는 인터페이스와 보안기능요구사항 간 매핑표가 포함되어 있어, 두 요소 간 추적성을 확보할 수 있다. 실제 현장에서 보안 문제가 발생하면 매핑표를 통해 원인을 파악하고 신속하게 보안 대책을 강구할 수 있다.

3.3 CIA-Level Extractor

CIA-Level Extractor는 기업체가 수행하고 있는 Secure SDLC 프로세스 수준을 정량적으로 분석한 후 CIA-Level을 추출한다. CIA-Level이란 정확성, 안전성, 보안성 측면을 고려한 지표로 Level1~Level7까지 7단계로 구성되며, 수준이 올라갈수록 Secure SDLC 프로세스가 체계적이고 엄격하게 관리된다는 것을 의미한다. 정확성, 안전성, 보안성 중 하나의 속성만 매핑될 경우 해당 속성의 CIA-Level로 산정되고, 2개 이상의 속성이 매핑될 경우 보안 활동 수행 현황을 통해 CIA-Level이 조율된다. 이러한 과정을 통해 산출되는 CIA-Level은 보안 활동 별 도출되며, 그 결과는 그래프 형식으로 출력되어 Secure SDLC 프로세스의 수준을 한눈에 파악할 수 있도록 한다. 이후 기업체는 동종 기업체 중 경쟁사를 선정하고, CIA-Level Extractor를 통해 산출된 경쟁사와 자사의 CIA-Level을 GAP Analyzer에 입력하여 두 조직 간 Secure SDLC 프로세스의 수준 차이를 정량적으로 분석하고 그 결과를 그래프 형식으로 확인할 수 있다. 이처럼 각 보안 활동 별 수준 차이를 그래프로 보여줌으로써, Secure SDLC 프로세스 중 미흡한 단계 및 보안 활동을 한눈에 파악할 수 있으며, 자사의 Secure SDLC 프로세스의 문제점을 파악하는데 용이하다. 잘 수립된 Secure SDLC 프로세스를 적용하고 있는 기업체의 경우 보안 활동 별 편차가 없이 일정한 그래프가 출력된다. 앞서 설명한 바와 같이 경쟁사와의 수준 차이를 정량적으로 분석해주는 GAP Analyzer가 수행하는 기능은 〔Fig 3〕과 같다.

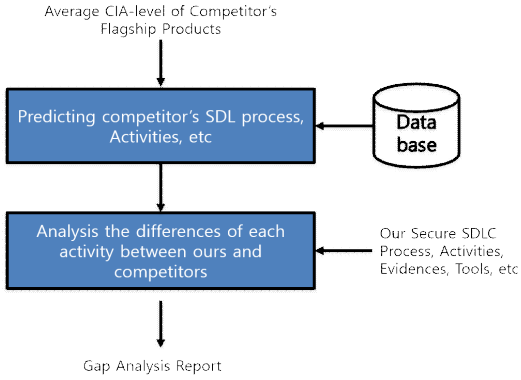

Fig. 3. GAP Analyzer

3.4 Customized Secure SDLC Constructor

Customized Secure SDLC Constructor는 GAP Analyzer의 결과를 기반으로 기업체가 원하는 수준의 Secure SDLC 프로세스, 보안 활동, 세부 보안 활동, 산출해야 할 문서 등을 도출한다.

만약 기업체와 경쟁사 간의 Secure SDLC 프로세스 수준 차이가 너무 큰 경우 한 번에 경쟁사 수준에 도달하기 어렵기 때문에, 기업체의 현재 상황을 고려하여 기업체가 원하는 수준의 Secure SDLC를 구축할 수 있도록 제시해준다.

앞서 선정한 경쟁사와 자사의 CIA-Level을 Customized Secure SDLC Constructor에 입력하면 두 조직의 분야 및 특성 등을 고려하여 전체 66개의 보안 활동 중 관련된 보안 활동들만 선택된다. 이후 선택된 보안 활동을 기반으로 기업체가 원하는 CIA-Level을 달성하기 위한 Secure SDLC 프로세스의 세부 보안 활동 및 산출해야 할 문서 등을 제공한다. 특히 산출해야 할 문서를 쉽게 도출함으로써 Secure SDLC 전체 프로세스 중 추적성을 확보할 수 있으며, 현장에서 보안 문제가 발생했을 경우 원인을 파악하고 신속하게 대응할 수 있다. 앞서 설명한 Customized Secure SDLC Constructor가 수행하는 기능은 [Fig 4]와 같다.

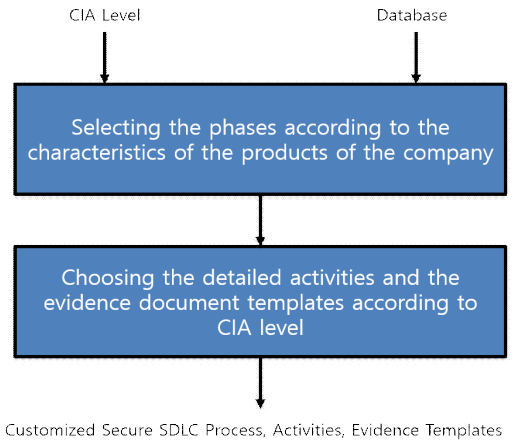

Fig. 4. Customized Secure SDLC Constructor

IV. 사례 연구

우리는 CIA-Level 기반 보안내재화 개발 프레임워크의 실효성을 입증하기 위하여 국내의 대표적인 소프트웨어 개발사에 실제 적용해보았다. 적용한 과정은 총 10단계이며, [Fig 5]와 같고, 이에 대한 과정에 대해 상세히 설명하고자 한다.

우리는 해당 기업체와 Secure SDLC를 비교할 경쟁사를 선정하기 위하여 기업체의 특성 및 현황을 파악하였다. 해당 기업체는 개발 초기 단계부터 배포

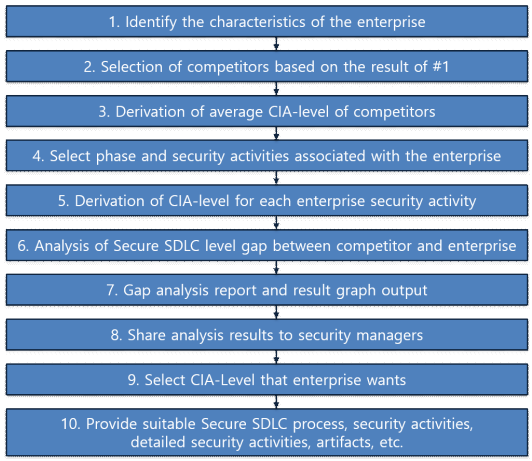

Fig. 5. CIA-Level Driven Secure SDLC Framework applied to enterprise

및 운영단계까지 모두 수행하고 있기 때문에, 이와 유사한 특징을 가진 대표적인 소프트웨어 개발사인 Microsoft를 경쟁사로 선정하였다.

우리는 두 조직의 Secure SDLC 프로세스의 수준 간 차이를 분석하기 위하여 우선적으로 Microsoft의 Secure SDLC 프로세스의 평균 CIA-Level을 추출하였다. 평균 CIA-Level을 추출한 방법은 Microsoft가 기존에 CC 인증을 획득한 사례를 기반으로 산정하였으며, 최근 5년간 CC 인증 사례를 조사한 결과 [Table 5]와 같이 데이터베이스 제품 5종과 운영체제 7종에 대한 사례가 있었

Table 5. CC Certification Case of Microsoft

	Product name	EAL	Year
DB	Microsoft SQL Server 2014	EAL2+	2015
	Microsoft SQL Server 2014	EAL4+	2015
	Microsoft SQL Server 2016	EAL4+	2017
	Microsoft SQL Server 2016 Database Engine Enterprise Edition	EAL2+	2017
	Microsoft SQL Server 2017	EAL4+	2020
OS	Microsoft Windows 10	EAL1	2016
	Windows 10 Anniversary Update and Microsoft Windows Server 2016	EAL1	2017
	Microsoft Windows 10	EAL1	2018
	Windows 10 and Windows Server	EAL1	2018
	Windows 10 and Windows Server	EAL1	2019
	Windows 10 and Windows Server 2019 version 1809	EAL1	2019
	Windows 10 and Server version 1903	EAL1	2019

음을 확인하였고, 이를 통해 평균 CIA-Level이 Level4를 달성한다고 판단하였다.

이후 우리는 기업체의 보안팀, 개발팀, 이해관계자를 대상으로 설문조사 및 토론 등을 통해 Secure SDLC 프로세스 수준을 각 보안 활동 별로 산정하였다. 이에 대한 결과를 CIA-Level Extractor에 입력하면, Database에 저장되어 있는 상세 세부 정보들 중 기업체와 연관된 단계 및 보안 활동 목록들이 선택된다. 이러한 과정을 수행한 결과, 10개의 단계 중 보안 교육, 착수 및 계획, 요구사항 분석, 설계, 구현, 검증, 배포, 운영 단계 총 8개의 단계가 선택되었으며, 66개의 보안 활동 중 58개의 보안 활동이 선택되었다. 58개의 보안 활동들은 CIA-Level Extractor에 의해 보안 활동 별 수준이 산정되었고, 그 결과는 〔Table 6〕과 같다.

Table 6. CIA-Level of enterprise

Phase	No.	Security activity	Level
1.Security Training	1	Basic security training	2
	2	Advanced security training	0
	3	Plan training schedules	1
2.Initiation	4	Project categorization	1
	5	Role identification	0
	6	Project tools selection	2
	7	Security requirements source identification	0
	8	Minimum quality level definition	2
	9	Prepare compensation system for handling security issues	0
	10	Plan project schedule	1
	11	Security goals setting by field	2
	12	Verifying consistency & completeness of goals	0
3.Requirements Analysis	13	Estimating scope of project security analysis	2
	14	Impact assessment for privacy	0
	15	Impact assessment for business	0
	16	Impact assessment for safety	0
	17	Existing software assessment	0
	18	Functional requirements elicitation	2
	19	Security Requirements elicitation	2
	20	Conformity & conflict check on requirements by field	0
	21	Verifying requirements based on security goals	0
5.Design	22	Functions & design specification	2
	23	Compliance with design best practices and principles	0
	24	Structural design for the integration process	0
	25	Asset identification	0
	26	Create data flow diagram	1
	27	Threat elicitation	1
	28	Attack Library Collection	1
	29	Risk analysis by field	0
	30	Mitigation elicitation by field	1
	31	Privacy analysis	0
	32	Usecase and misuse case identification	0
	33	Verifying design based on requirements	0
6.Implemen-tation	34	Compliance with secure coding guidelines	4
	35	Creation for deployment guide document and tools	0
	36	Implementation verification according to design	0
7.Verification	37	Static analysis	4
	38	Automated static analysis	4
	39	Dynamic analysis	4
	40	Integration & acceptance test	4
	41	Penetration test	4
	42	Review and update for threat model	0
	43	Review for minimum quality and security level	0
	44	Review and update for security documents	0

Phase	No.	Security activity	Level
8.Release	45	Final security review	1
	46	Final privacy review	0
	47	Requirements elicitation for production	1
	48	Production procedure determination	1
	49	Verification of production	1
	50	Accident response planning	2
	51	Security review for deployment procedure	2
9.Operation	52	Monitoring planning	1
	53	Continuous monitoring	2
	54	Vulnerability report	2
	55	Vulnerabilities assessment	2
	56	Solution establishment	4
	57	Vulnerability disclosure & patch/update	2
	58	Configuration management after release	2

기업체의 Secure SDLC 프로세스의 세부 보안 활동 항목 58개 중 6개의 보안 활동만 Microsoft SDL과 동등한 수준을 가진다고 판단하였다. 또한 기업체가 경쟁사인 Microsoft와 동등하다고 판단되는 단계는 주로 구현, 검증, 운영 단계에 집중되어 있고, 보안 교육, 착수 및 계획, 요구사항 분석, 설계, 배포 단계에서는 두 조직 간 보안 수준 차이가 현저한 것으로 분석되었으며, 그 결과는 [Fig 6]과 같다. 이를 통해 우리는 기업체에게 Microsoft SDL의 수준에 한 번에 도달하는 것은 매우 어렵다는 점을 전달하고, 점진적으로 개선해나가기 위해 Level2를 우선적으로 달성할 것을 제시하였다. 또한 제대로 수립된 Secure SDLC 프로세스의 경우 보안 활동 별 편차가 없이 일정한 그래프를 출력해야 하지만, 해당 기업체의 경우 보안 활동 별 편차가 심하여 미흡한 수준의 보안 활동들을 평균 수준으로 끌어올리는 것이 중요하다는 의견을 보안 담당자들 및 임원들에게 전달하였다.

이것은 CIA-Level 기반 보안내재화 개발 프레임워크를 실제 현장에 처음으로 적용해본 사례로, 우리가 예측했던 대로의 결과가 도출되었다. 또한 기업체가 실제 환경에 즉시 적용하고 프로세스를 개선함에 따라 본 방법론의 실효성을 입증하였다.

IV. 결 론

미국 정부는 1970년대부터 모의 해킹을 통해 제품을 개선하는데 한계가 있음을 인식하고, 1980년대부터 제품의 보안 품질을 향상시키기 위해서는 개발 프로세스 자체를 체계적이고 엄격하게 관리해야 함을 인식하였다. 이와 같은 이유로 요구사항 분석 및 설계 단계와 같이 개발 초기 단계부터 보안을 고려하여 제품의 복잡도를 감소시키는 보안내재화라는 개념이 사용되었다. 보안내재화를 통해 보안 제품의 복잡도를 감소시키고 궁극적으로 신뢰성을 달성하고자 하였다. 이러한 보안내재화가 개발 프로세스 내에 적용된 것을 Secure SDLC라고 명명한다.

하지만 대표적으로 활용되고 있는 Secure SDLC 표준이나 가이드라인들은 너무 일반적이며 구체적인

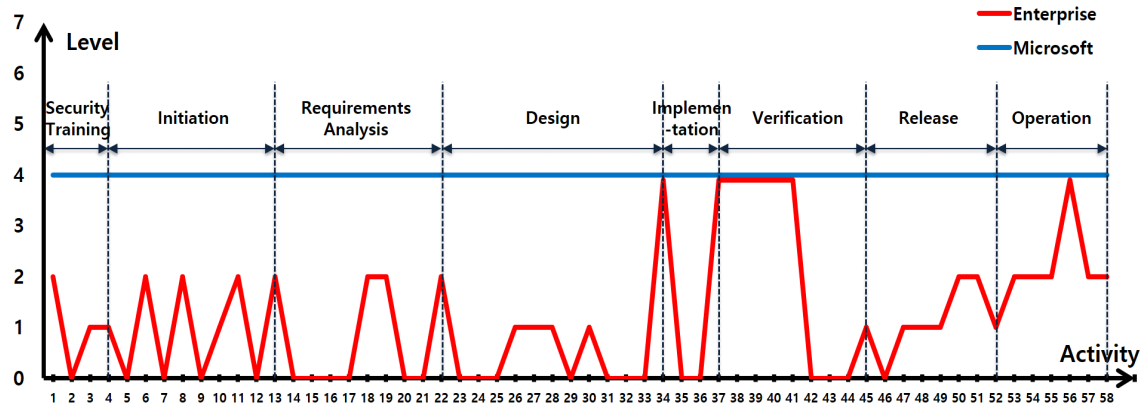

Fig. 6. Gap analysis result between Microsoft and enterprise

내용들이 포함되어 있지 않기 때문에 실제 현장에서 활용하기가 매우 어렵다. 특히 각 단계들 간 추적성을 확보하기 위해 반드시 필요한 산출 문서의 경우 예시 템플릿이 공개되지 않거나 템플릿의 내용이 구체적이지 않다. 따라서 우리는 Secure SDLC를 기업체가 원하는 수준에 맞게 구체화시키기 위하여 CIA-Level 기반 보안내재화 프레임워크를 제안하였다.

CIA-Level 기반 보안내재화 프레임워크를 사용할 경우, 첫째, 자사와 경쟁사 간 갭 분석을 통해 Secure SDLC 프로세스의 수준 차이를 정량적으로 분석하도록 하였다. 둘째, Secure SDLC를 필요로 하는 조직이 원하는 수준의 Secure SDLC를 구축하는데 필요한 상세한 세부 보안 활동 및 산출해야 하는 문서에 대한 예시 템플릿 등을 제공하여 실제 현장에서의 Secure SDLC를 구축할 때 활용이 용이하도록 하였다.

우리는 CIA-Level 기반 보안내재화 프레임워크의 실효성을 입증하기 위하여 국내의 대표적인 소프트웨어 개발사를 선정한 후 실제 환경에 적용해 보았다. 적용한 결과, 우리가 예측한대로 보안 활동 별 결과가 도출되었고 결과보고서를 기업체에 전달하여 즉각 활용하여 프로세스가 개선됨을 확인할 수 있었다. 이를 통해 본 방법론의 실효성을 입증하고 실제 환경에 적용 가능함을 확인하였다.

본 연구를 수행하며 Secure SDLC 표준 및 가이드라인 10건과 증거 기반 보안 방법론 3건을 매핑하는 작업은 방대하고 반복적이기 때문에 실무에 활용할 수 있을 정도의 상세한 결과를 도출하기 위해서는 매우 많은 시간이 소모되었다. 또한 산출해야 하는 문서 중 공개되지 않는 템플릿이 있어 수집하는데 어려움이 있었다. 따라서 본 연구를 수행하며 수동으로 작업했던 부분들을 자동화한다면 실제 현장에서 더욱 활용이 용이할 것으로 기대된다.

References

[1] Anderson, James P. "Computer security technology planning study." Oct. 1972.

[2] Voas, Jeffrey, et al. "Defining an adaptive software security metric from a dynamic software failure tolerance measure." Proceedings of 11th Annual Conference on Computer Assurance. COMPASS'96. IEEE, Jun. 1996.

[3] Hunt, Edward. "US Government computer penetration programs and the implications for cyberwar." IEEE Annals of the History of Computing 34.3, pp.4-21, Dec. 2011.

[4] Viega, John, and Gary McGraw. "Building Secure Software: How to Avoid Security Problems the Right Way." Addison-Wesley Professional, Aug. 2011.

[5] McGraw, Gary. "Testing for security during development: why we should scrap penetrate-and-patch." IEEE aerospace and electronic systems magazine 13.4, pp.13-15, Apr. 1998.

[6] Sabo, Sandra R. "Security by Design." American School Board Journal 180.1, pp.37-39, 1993.

[7] Casola, Valentina, et al. "Security-by-design in Clouds: A Security-SLA Driven Methodology to Build Secure Cloud Applications." Cloud Forward, pp.53-62, Oct. 2016.

[8] Geismann, Johannes, Christopher Gerking, and Eric Bodden. "Towards ensuring security by design in cyber-physical systems engineering processes." Proceedings of the 2018 International Conference on Software and System Process, May. 2018.

[9] Hardin, Russell. "Trustworthiness." Ethics 107.1, pp.26-42, Oct. 1996.

[10] Avizienis, Algirdas, "Basic concepts and taxonomy of dependable and secure computing." IEEE transactions on dependable and secure computing 1.1, pp.11-33, Mar. 2004.

[11] Spiekermann, Sarah. "The challenges of privacy by design." Communications of the ACM 55.7, pp.38-40, Jul. 2012.

[12] Cavoukian, Ann, and Mark Dixon. "Privacy and security by design: An enterprise architecture approach." Information and Privacy Commissioner of Ontario, Canada, Dec. 2013.
[13] Herrmann, Debra S. "A practical guide to security engineering and information assurance." CRC Press, Dec. 2001.
[14] Cherdantseva, Yulia, and Jeremy Hilton. "Information security and information assurance: discussion about the meaning, scope, and goals." Standards and Standardization: Concepts, Methodologies, Tools, and Applications. IGI Global, pp.1204-1235, 2015.
[15] Latham, Donald C. "Department of defense trusted computer system evaluation criteria." Department of Defense, Dec. 1986.
[16] Jahl, Christian. "The information technology security evaluation criteria (ITSEC)." 13th International Conference on Software Engineering. IEEE, 1991.
[17] Bacic, E. Mate. "The canadian trusted computer product evaluation criteria(CTCPEC)." Proceedings of the Sixth Annual Computer Security Applications Conference. IEEE, 1990.
[18] Instruction, DoD. "DoD Information Technology Security Certification and Accreditation Process (DITSCAP)." 1997.
[19] Instruction, DoD. "National Information Assurance Certification and Accreditation Process (NIACAP)." 2000.
[20] Instruction, DoD. "DoD Information Assurance Certification and Accreditation Process (DIACAP)." 2011.
[21] Williams, Peter, and Tiffani Steward. "DoD's Information Assurance Certification & Accreditation Process." DEFENSE AT AND L 36.5, 2007.
[22] Lipner, Steve. "The trustworthy computing security development lifecycle." 20th Annual Computer Security Applications Conference. IEEE, 2004.
[23] Viega, Jon. "Security in the software development lifecycle." (2005).
[24] Microsofot. "Microsoft Vulnerabilities Report."
[25] Microsoft, "Security Development Lifecycle - SDL Process Guidance Version 5.2", 2012
[26] United States Congress, "NIST SP 800-64 Revision 2 - Security Considerations in the System Development Life Cycle", 2019
[27] CSA, "Security by Design Framework version 1.0". 2017
[28] SAFECode, "Fundamental Practices for Secure Software Development 2nd Edition"
[29] Tiirik, Karl. "Comparison of SDL and Touchpoints." Last retrieved 11 (2004): 16-18.
[30] OWASP, "Comprehensive, Lightweight Application Security Process."
[31] Cigital, "Building Security in Maturity Model 1.0."
[32] OWASP, "Software Assurance Maturity Model 2.0 - A guide to building."
[33] NIST, "NIST Special Publication 800-37 Revision 2 - Risk Management Framework for Information Systems and Organizations."
[34] SAE, "Cybersecurity Guidebook for Cyber-Physical Vehicle Systems"
[35] Lockeed Martin, "Cyber Resiliency Level."
[36] UNECE, "Draft Cyber Security Regulation-final clean version (2020)."
[37] Debouk, Rami, "Overview of the 2nd Edition of ISO 26262: Functional Safety-Road Vehicles." General

Motors Company, Warren, MI, USA Oct. 2018.
〔38〕 Volve, "Connected Vehicle Cyber-security Volvo Group Trucks Technology." Chalmers, Oct. 2018.
〔39〕 Lee, Younghwa, Jintae Lee, and Zoonky Lee. "Integrating software lifecycle process standards with security engineering." Computers & Security 21.4, pp.345-355, Aug. 2002.
〔40〕 Mellado, Daniel, Eduardo Fernández‐Medina, and Mario Piattini. " A common criteria based security requirements engineering process for the development of secure information systems." Computer standards & interfaces 29.2, pp.244-253, Feb. 2007.
〔41〕 Sheikhpour, Razieh, and Nasser Modiri. "A best practice approach for integration of ITIL and ISO/IEC 27001 services for information security management." Indian journal of science and technology 5.2, pp.2170-2176, Feb. 2012.
〔42〕 Futcher, Lynn, and Rossouw von Solms. "SecSDM: a model for integrating security into the software development life cycle." IFIP World Conference on Information Security Education. Springer, New York, NY, pp.41-48, Oct. 2007.
〔43〕 Nayerifard, Tahereh, Nasser Modiri, and Sam Jabbehdari. "An Approach for Software Security Evaluation Based on ISO/IEC 15408 in the ISMS Implementation." International Journal of Computer Science and Information Security 11.9, pp.7-11, Oct. 2013.
〔44〕 Kriaa, Siwar, et al. "A survey of approaches combining safety and security for industrial control systems." Reliability engineering & system safety 139, pp.156-178, Jul. 2015.
〔45〕 Mesquida, Antoni Lluís, and Antonia Mas. "Implementing information security best practices on software lifecycle processes: The ISO/IEC 15504 Security Extension." Computers & Security 48, pp.19-34, Feb. 2015.
〔46〕 Mohammed, Nabil M., et al. "Exploring software security approaches in software development lifecycle: A systematic mapping study." Computer Standards & Interfaces 50, pp.107-115, Feb. 2017.
〔47〕 Sánchez-Gordón, Mary-Luz, et al. "Towards the Integration of Security Practices in the Software Implementation Process of ISO/IEC 29110: A Mapping." European Conference on Software Process Improvement. Springer, Cham, pp.3-14, Aug. 2017.
〔48〕 Morrison, Patrick, et al. "Mapping the field of software life cycle security metrics." Information and Software Technology 102, pp.146-159, Oct. 2018.
〔49〕 Casola, Valentina, et al. "A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach." Journal of Systems and Software 163.11537, May. 2020
〔50〕 Kitchenham, Barbara A., Tore Dyba, and Magne Jorgensen. "Evidence-based software engineering." Proceedings. 26th International Conference on Software Engineering. IEEE, May. 2004.
〔51〕 Dyba, Tore, Barbara A. Kitchenham, and Magne Jorgensen. "Evidence-based software engineering for practitioners." IEEE software 22.1, pp.58-65, Feb. 2005
〔52〕 Cicotti, Giuseppe. "An evidence-based risk-oriented V-model methodology to develop ambient intelligent medical

software." Journal of Reliable Intelligent Environments 3.1, pp.41-53, May. 2017.
[53] Formoso, Saul, and Massimo Felici. "Evidence-Based Security and Privacy Assurance in Cloud Ecosystems." IFIP International Summer School on Privacy and Identity Management. Springer, Cham, pp.205-219, Jul. 2016.
[54] Karim, Nor Shahriza Abdul, et al. "The practice of secure software development in SDLC: an investigation through existing model and a case study." Security and Communication Networks 9.18, pp.5333-5345, Nov. 2016.
[55] ACM, https://dl.acm.org/
[56] Elsevier, https://www.sciencedirect.com/
[57] IEEE, https://ieeexplore.ieee.org/
[58] Scopus, https://www.scopus.com/
[59] Springer, https://www.springer.com/
[60] Google scholar, https://scholar.google.co.kr/
[61] Eloff, Jan HP, and Mariki Eloff. "Information security management: a new paradigm." Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, pp.130-136, 2003.
[62] Hoxey, Cynthia, and Dan Shoemaker. "Navigating the information security landscape: Mapping the relationship between ISO 15408: 1999 and ISO 17799: 2000." AMCIS 2005 Proceedings 448, pp.3203-3214, Aug. 2005.
[63] Białas, Andrzej. "Development of an Integrated, Risk-based Platform for Information and E-services Security." International Conference on Computer Safety, Reliability, and Security. Springer, Berlin, Heidelberg, pp.316-329, Sep. 2006.
[64] Mellado, Daniel, Eduardo Fernández-Medina, and Mario Piattini. "Towards security requirements management for software product lines: A security domain requirements engineering process." Computer Standards & Interfaces 30.6, pp.361-371. Aug. 2008.
[65] Mellado, Daniel, et al. "A systematic review of security requirements engineering." Computer Standards & Interfaces 32.4, pp.153-165, Jun. 2010.
[66] Yin, Lei, and Fang-Liang Qiu. "A novel method of security requirements development integrated common criteria." 2010 International Conference On Computer Design and Applications. Vol. 5. IEEE, Jun. 2010.
[67] Beckers, Kristian, et al. "A structured comparison of security standards." Engineering secure future internet services and systems. Springer, Cham, pp.1-34, 2014.
[68] Young, William, and Nancy G. Leveson. "An integrated approach to safety and security based on systems theory." Communications of the ACM 57.2, pp.31-35, Feb. 2014.
[69] Beckers, Kristian. "The CAST Method for Comparing Security Standards." Pattern and Security Requirements. Springer, Cham, pp.51-83, Apr. 2015.
[70] Sabaliauskaite, Giedre, and Aditya P. Mathur. "Aligning cyber-physical system safety and security." Complex Systems Design & Management Asia. Springer, Cham, pp.41-53, Jul. 2015.
[71] Li, Hongbo, et al. "Fesr: A framework for eliciting security requirements based on integration of common criteria and weakness detection formal model." 2017 IEEE

International Conference on Software Quality, Reliability and Security (QRS). IEEE, Jul. 2017.

[72] Williams, Laurie. "Secure software lifecycle knowledge area." The National Cyber Security Centre, Aug. 2019.

[73] Chen, Earl, et al. "Designing security into software during the development lifecycle." U.S. Patent Application No. 13619581. 2013.

[74] Mir, Talhah Munawar, et al. "Threat analysis and modeling during a software development lifecycle of a software application." U.S. Patent No.8091065. 2012.

[75] Wilcock, Lawrence, et al. "Automated lifecycle management of a computer implemented service." U.S. Patent No.8312419. 2012.

[76] Silke Holtmanns and Rune Lindholm, "Enhanced lifecycle management of security module", Patent Application No.CN103988530A. 2018.

[77] Schilder, Marius, et al. "Secure device state apparatus and method and lifecycle management." U.S. Patent No.10223531. 2018.

[78] Bhalla, Nishchal, et al. "Security risk identification in a secure software lifecycle." U.S. Patent Application No.15784072. 2019

[79] Vincent, Benjamin, and Ariel Gordon. "Security configuration lifecycle account protection for minors." U.S. Patent Application No.16022554. 2020

[80] De Win, Bart, et al. "On the secure software development process: CLASP, SDL and Touchpoints compared." Information and software technology 51.7, pp.1152-1171, Jul. 2009

[81] Carter, Ashton. "The Department of Defense cyber strategy." The US Department of Defense, Washington, 2015.

[82] ISO/IEC 15408, "Information technology - Security techniques - Evaluation criteria for IT security(CC)."

[83] ISO/IEC 27001, "Information Security Management(ISMS)."

[84] ISO/IEC 27701, "Security techniques - Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management (PIMS)."

〈저 자 소 개〉

강 수 영 (Sooyoung Kang) 학생회원
2002년~2006년: 순천향대학교 컴퓨터공학부 공학사
2006년~2008년: 순천향대학교 컴퓨터공학부 공학석사
2008년~2010년: 한국인터넷진흥원(KISA) 연구원
2010년~2014년: 안랩(Ahnlab) 주임연구원
2013년~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 보안내재화 방법론, 위협 모델링, 보안성 평가/인증

김 승 주 (Seungjoo Kim) 종신회원
1994년~1999년: 성균관대학교 정보공학과(학사, 석사, 박사)
1998년~2004년: 한국인터넷진흥원(KISA) 팀장
2004년~2011년: 성균관대학교 정보통신공학부 부교수
2004년~현재: 한국정보보호학회 이사
2007년: 국가정보원장 국가사이버안전업무 유공자 표창
2010년: 방송통신위원회 정보통신망 침해사고 민관합동조사단 위원
2011년~현재: 고려대학교 사이버국방학과/정보보호대학원 정교수
2012년: 선관위 디도스 특별검사팀 자문위원
2014년~2015년: 육군사관학교 초빙교수
2014년~2016년: 다음카카오 프라이버시 정책 자문위원회 위원
2015년~현재: 방위사업청 방산기술보호 자문관
2016년~2018년: 개인정보분쟁조정위원회 위원
2016년~현재: 산업통상자원부 전략물자기술 자문위원
2016년~현재: 한국카카오뱅크 정보보호부문 자문교수
2017년~현재: 고려대학교 국방RMF연구센터(AR^2C) 센터장
2018년~2020년: 4차산업혁명위원회 위원: 대통령직속 4차산업혁명위원회 위원
2018년~현재: 고신뢰 보안운영체제 연구센터(CHAOS) 센터장
2020년~현재: 합동참모본부 정책자문위원회 자문위원
〈관심분야〉 보안공학 및 보안내재화 방법론, 보안성 평가/인증, RMF A&A, 암호학 및 블록체인