객체지향 개발환경에서의 보안 요구사항명세에 관한 연구

김기한* · 채수영* · 최명렬* · 박상서*

요 약

소프트웨어 개발시 내재될 수 있는 취약성을 최소화하기 위해서는 요구사항 분석단계에서부터 보안 요구사항을 잘 정의하여야 한다. 본 논문에서는 객체지향 개발 방법론에서 소프트웨어 보안 요구사항 명세를 위한 체계적인 방안을 제시한다. 본 논문에서 제시한 방안은 크게 보안 목표 설정, 위협식별, 공격트리 작성 그리고 보안기능 명세로 이루어진다. 이 방법을 이용하면 소프트웨어가 가져야 할 보안 요구사항과 기능을 보다 명확하고 체계적으로 작성할 수 있다.

A Study on Security Requirements Specification in an Object-Oriented Development Environment

Gi Han Kim* · Soo Young Chae*
Myeong Ryeol Choi* · Sangseo Park*

ABSTRACT

Security requirements must be defined well to reduce software vulnerabilities in requirement specification phase. In this paper, we show how to specify security requirements in structured manner for object-oriented development methodology. Our method specifies security requirements through four phases: defining security objectives, identifying the threat, construct attack tree, and specifying security function. This method would help developers to specify security requirements and functions which software have to possess clearly and systematically.

Key words : Secure Systems Development, Security Requirement

* 국가보안기술연구소

1. 서 론

현재 대부분의 소프트웨어 보안향상 방법은 개발한 소프트웨어가 취약성을 가지는 경우 공격 기법이 발표되고 나서야 패치를 수행하는 penetrate-and-patch 방식이다. 그러나 이 방식의 문제점은 패치가 발표되는 시점까지 공격을 막을 수 있는 방법은 없을 뿐만 아니라 패치가 발표되더라도 패치에 새로운 결함이 포함될 수 있는 것이다[1].

이와 같은 문제점을 해결하기 위한 전통적인 방식은 소프트웨어 개발 정형화 방법을 이용하여 보다 안전한 소프트웨어를 개발하는 접근법도 있으나 정형화 방법에 대한 교육이 필요하고 정형화 명세를 작성하는데 비용이 많이 들어가 현실적으로 적용이 어려운 실정이다[2].

본 논문에서는 객체지향 개발환경에서 보안성 향상을 위한 방법으로 보안 요구사항을 작성하는 방법을 제안한다. 개발 초기단계에서 명확한 보안 목표와 보안 요구사항의 식별은 소프트웨어의 보안성 향상에 도움을 준다. 또한 개발 초기단계에서 소프트웨어와 운영 환경에 대한 다양한 위협의 식별과 위협에 대한 분석을 수행하여 개발 후에 발생할 수 있는 취약점을 줄이는데 위협모델링은 도움을 준다.

본 논문의 2장에서는 기존의 객체지향 개발 방법론에서 적용되는 보안 요구사항 명세방법에 대해 알아보고, 3장에서 위협 모델링을 포함하는 보안명세 방법을 제시한다. 4장에서는 본 논문에서 제시하는 보안 요구사항 명세방법을 다른 방법과 비교하며 5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 대표적인 객체지향 방법론인 RUP[3]에서의 보안 요구사항 명세 방법에 대해 설명한다. 그리고 객체지향 보안 소프트웨어 개발 방법론인 UMLsec[4], CLASP[5]에서의 보안 요구사항 명세에 대해 설명한다.

2.1 RUP의 보안 요구사항 명세

RUP에서는 요구사항 명세를 위해 액터와 use case를 식별하여 먼저 use case 모델을 작성한다. Use case 명세서에는 use case의 사전조건에 대해서 기술하고 이벤트 흐름 정의(Flow of events)에서 주요흐름과 선택 가능한 흐름을 기술한다. RUP에서는 use case를 이용한 요구사항 명세 외에 보충 요구사항(Supplementary requirements)에 비기능적 요구사항을 기술한다. 보충 요구사항에 포함되는 항목은 <표 1>과 같다.

〈표 1〉 RUP의 보충 요구사항

요구사항	설 명
인터페이스 요구사항	외부 요소와의 인터페이스에 관련된 제약사항을 기술한다. 예를 들어 시스템 간의 인터액션의 타이밍, 포맷 등을 설명한다.
물리적 요구사항	시스템이 갖추어야하는 물리적인 제약사항을 기술한다. 예를 들어 시스템 하드웨어 요구사항과 네트워크 구성 등을 설명한다.
설계 제약사항	시스템 설계에서 확장성, 유지보수에 관련된 제약사항을 기술한다. 예를 들어 리가시 시스템의 재사용성 등을 설명한다.
구현 제약사항	코딩과 시스템 구축에 관련된 제약사항을 기술한다. 예를 들어 코딩 가이드라인, 구현언어, 운영체제 환경 등을 설명한다.
기타 요구사항	보안성, 가용성, 교육 등에 대한 내용을 설명한다.

RUP에서 보안 요구사항은 보충 요구사항의 기타 요구사항으로 명세하기 때문에 구체적인 보안기능을 제시하지 않고 요구사항 명세 단계에서 위협 모델링 방법도 제시하지 않았다.

2.2 UMLsec의 보안 요구사항 명세

UMLsec은 UML확장을 이용하여 사용자 데이터의 기밀성, 무결성, 등의 보안 요구사항을 분석과 설계에 반영하는 표준화된 기호를 제공하고 설계의 검증을 지원한다. 보안 요구사항 명세관점에서 UMLsec은 use case를 이용하여 보안 요구사항을 식별하는 것으로 RUP에 비해 특별한 요구사항 명세 워크플로우를 제시하지 않기 때문에 명세를 하고자 하는 보안 기능의 기준이 없다.

UMLsec에서의 위협 모델링은 특정 공격에 대해서 데이터의 읽기, 삭제, 삽입 위협에 대해서만 고려할 뿐 SW 운영환경에 대한 위협은 고려하지 않고 있다.

2.3 CLASP의 보안 요구사항 명세

CLASP는 RUP에 보안 프로세스를 포함한 방법으로 IBM에서 제안하였다. CLASP에서는 ① 시스템의 역할과 자원을 식별하고, ② 자원을 추상화화된 카테고리로 분류하고, ③ 시스템에서 자원간의 인터액션을 식별하며, ④ 마지막으로 각각의 카테고리별로 보안 서비스 매커니즘을 명세한다. 명세에 기준이 되는 서비스 매커니즘은 권한부여, 인증과 무결성, 프라이버시를 포함하는 기밀성, 가용성, 부인 봉쇄를 포함하는 계정정보이지만 보안감사와 접근제어에 관한 사항에 대한 부분은 포함하지 않고 있다.

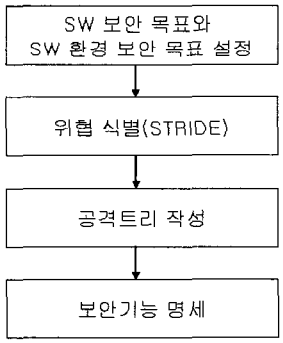
또한 CLASP에서의 위협 모델링은 시스템의 역할에서 공격자를 포함하여 공격자가 수행할 수 있는 위협을 식별한다.

이러한 역할과 자원의 분류는 공격자가 접근하는 자원에 대한 식별은 용이하지만 공격에 대한 구조적인 분석은 힘든 단점이 있다.

3. SEM 보안 요구사항 명세 방법

본 논문에서는 SEM(Security Enhanced software development Methodology) 보안 요구사항 명세 방법을 제시한다.

SEM 보안 요구사항 명세를 작성하는 단계를 표현하면 (그림 1)과 같다. SEM 보안 요구사항 명세는 우선 Common Criteria의 보호프로파일을 이용하여 소프트웨어 보안 목표와 소프트웨어 환경 보안 목표를 설정한다. 두 번째로는 STRIDE 기법을 이용하여 소프트웨어와 소프트웨어 환경에 존재하는 위협을 식별하고 세 번째로, 식별된 위협을 공격트리로 작성하여 위협에 대한 구조적인 분석과 대응을 고려하는데 도움을 준다. 마지막으로 CC의 보안 기능 클래스와 컴포넌트를 기준으로 보안기능을 명세한다.



(그림 1) SEM 보안 요구사항 명세 단계

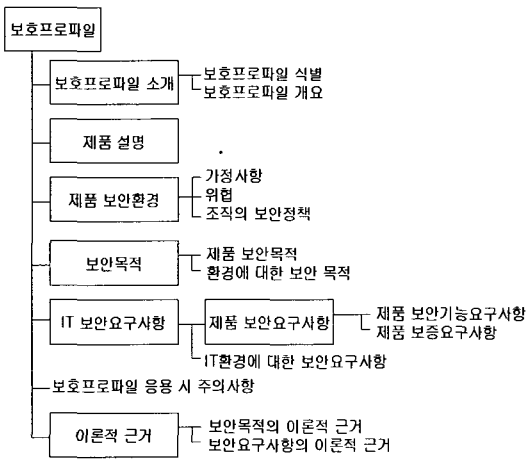
3.1 SW 보안 목표와 환경 보안 목표 설정

CC에서 보호 프로파일은 일련의 보안 요구사항을 포함하고 있는데, 이는 CC에서 선택한 보안 요구사항 또는 별도로 명시된 보안 요구사항들로 구성되고 하나의 평가보증등급을 포함하며 추후 보호 프로파일 평가를 받는데 사용된다[6].

보호 프로파일은 제품 집합에 대하여 일련의 보안목적을 만족시키는 보안 요구사항들을 구현에 독립적으로 표현하고 보안목적 및 보안 요구사항에 대한 이론적 근거를 포함하고 있다. 보호

프로파일은 일련의 공통된 요구사항을 정의하는데 관심이 있는 사용자, 제품 개발자, 등에 의해 개발될 수 있고, 고객이 구체적인 요구사항을 작성하는데 참조할 수 있는 수단이 된다.

보호 프로파일의 구성요소는 (그림 2)와 같다.



(그림 2) CC의 보호프로파일 구성요소

SEM 보안 요구사항 명세 방법은 CC의 보호프로파일에서 "제품 보안 목적"과 "환경에 대한 보안 목적" 그리고 "제품 보안기능요구사항" 3가지 부분으로 작성된다.

제품 보안 목적 부분에서는 개발할 소프트웨어의 보안 요구사항이 가지는 목적을 기술하고 환경에 대한 보안 목적은 소프트웨어 자체의 위협이 아니라 운영 환경의 위협을 대응할 수 있는 보안 목적을 기술한다.

제품 보안기능요구사항은 제품과 환경에 대한 보안 목적을 만족할 수 있도록 CC의 보안기능 패밀리[7]의 측면에서 상세 보안기능요구사항을 기술한다. CC의 프로파일의 제품 보안기능요구사항은 보안기능 컴포넌트 단계로 구체적인 기능을 기술하지만 SEM 보안요구명세는 필요에 따라 패밀리 단계에서의 보안 요구사항의 추상화한 명세도 가능하다.

3.2 위협 식별(STRIDE)

위협 식별 단계에서는 Microsoft의 STRIDE 모델을 이용한다. STRIDE는 소프트웨어 보안 위협을 다음과 같이 6개로 분류하고 있다[8].

- 스푸핑 식별(Spoofing Identity) : 사용자 아이디, 비밀번호와 같은 다른 사용자의 인증 정보를 불법적으로 접근 및 사용하는 것으로 다른 사용자처럼 보이게 하는 것을 포함한다. man-in-the-middle 공격도 이 카테고리에 포함된다.
- 데이터 간섭(Tampering with data) : 데이터의 악의적인 변조를 의미한다. 특별한 예로 세션 하이잭킹 공격도 이 범주에 포함된다.
- 부인(Repudiation) : 보안성 지원하기 위한 부인봉쇄의 반대 의미이다. 트랜잭션에 참여한 참가자가 자신은 그 트랜잭션과 관련이 없다고 고의로 부인해버리는 경우로 트랜잭션의 위협을 의미한다.
- 정보유출(Information disclosure) : 공격자가 접근권한 없이 정보에 접근하는 위협을 의미한다. 예를 들어 네트워크 스니핑에 의한 공격도 이 카테고리에 포함한다.
- 서비스거부 공격(Denial of service) : 정상적인 사용자가 시스템이나 서비스를 사용하지 못하도록 서비스 자원을 소진시키는 위협을 의미한다.
- 권한 상승(Elevation of privilege) : 권한을 가질 수 없는 사용자가 불법적으로 권한 상승하는 위협을 의미한다. 예를 들어 일반사용자가 불법적으로 권한을 상승하여 Administrator 그룹의 권한을 가지는 경우이다.

STRIDE 모델을 이용하여 위협에 대해 식별할 때 세부적인 공격 절차에 대해서는 고려하지 않는다. 위협에 대한 세부적인 공격 절차는 공격트리 작성 단계에서 수행한다.

3.3 공격트리 작성

공격트리 작성 단계에서는 STRIDE 방법으로 식별된 각각의 위협에 대해 공격트리[9]를 작성하여 위협을 상세히 분석한다.

인가받지 않는 사용자가 로그인을 하는 위협을 공격트리를 이용하여 위협 모델링을 수행하면 (그림 3)과 같다. 인가받지 않은 사용자의 접속은 STRIDE의 스푸핑 식별에 대한 위협에 해당한다.

공격트리의 각각의 노드는 각 단계의 중간목표로서 자식 노드가 중간 목표를 성취하는 방법을 표현한다. 이러한 노드는 AND 노드와 OR 노드를 자식노드로 가진다.

OR노드는 선택적이다. OR노드는 위장인증 이라는 목표를 달성하기 위해 클라이언트 인증서와 비밀키 획득, 또는 TCP 연결 하이재킹, 둘 중 하나를 성공하면 된다.

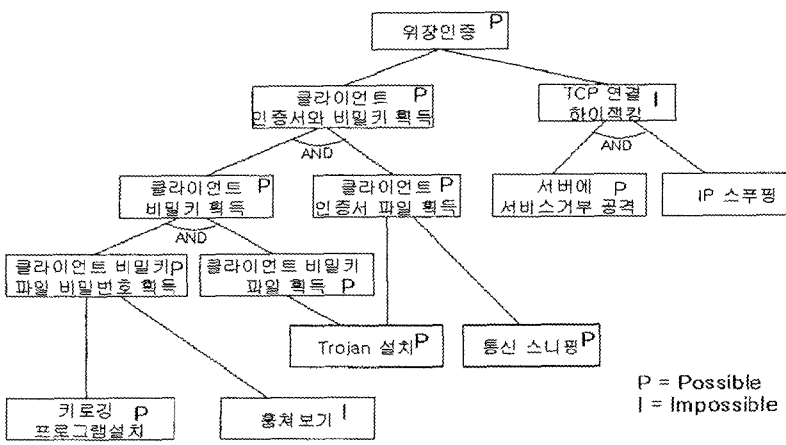
AND 노드는 부모노드의 목표를 달성을 위해 형제노드를 반드시 수행해야 함을 의미한다. 즉, 클라이언트 인증서와 비밀키 획득을 위해 자식 노드인 클라이언트 비밀키 획득과 클라이언트 인증서 파일 획득이라는 조건이 모두 만족되어야 한다.

(그림 3)의 공격트리에서는 I(Impossible)와 P(Possible)의 값이 측정되어 있다. I와 P값은 말단 노드에서만 계산된다. 만약 OR 노드의 경우 자식 노드 중 하나라도 P 면 P 값을 가지고 자식 노드 모두가 I 값을 가지면 I 값을 가지게 된다. AND 노드의 경우 자식 노드가 모두 P 값을 가져야 P 값을 가지고 자식 노드 중 하나라도 I 값을 가지면 I 값을 가진다.

본 예에서는 TCP 연결 하이재킹 노드에서 서버에 대한 서비스 공격은 가능하더라도 IP 스푸핑을 하려면 서버 네트워크 영역으로 공격자의 패킷이 흘러들어 가야하고 TCP 시퀀스 넘버를 예측해야하는 기술적인 어려움이 있음으로 불가능한 공격으로 본 예에서는 판명하였다. 이와 같이 P 와 I 값의 결정은 공격 기술의 난이도, 목표를 이루기 위한 비용 등의 기준을 평가하는데 이용할 수 있다.

3.4 보안기능 명세

마지막으로 소프트웨어 보안 목표와 운영환경 보안목표를 만족할 수 있도록 CC 보안기능 패밀리를 기준으로 보안기능요구사항을 명세한다. CC의 보안기능 패밀리를 참조함으로서 다양한 보안기능을 고려한 명세가 가능하다.



(그림 3) 공격트리의 예

〈표 2〉 보안 요구사항 명세 방법 비교

	UMLsec의 보안요구명세	CLASP의 보안요구명세	SEM의 보안요구명세
보안 요구사항 기술방법	· UseCase 다이어그램과 Use Case 명세를 이용한 보안 요구사항 이해	· 시스템의 역할과 자원식별 후 이들의 상호작용을 식별로 요구사항을 작성	· CC의 보호 프로파일을 이용한 보안 요구사항 기술
보안 기능 기준	· 특별한 기준없음	· CLASP에서 정의한 기준(권한부여, 인증과 무결성, 프라이버시를 포함하는 기밀성, 가용성, 부인 봉쇄를 포함하는 계정정보)	· CC Part2의 보안기능 클래스와 패밀리 기준을 이용하여 명확한 보안 기능 기준 제시
위협 모델링	· 허가받지 않은 데이터 삭제, 읽기, 삽입 위협의 식별만 수행 · 소프트웨어 운영환경에 대한 위협 고려 없음	· 공격자를 시스템 기본 역할에 포함하여 위협 표현 · 소프트웨어 운영환경에 대한 위협 고려 없음	· STRIDE와 공격트리를 이용하여 소프트웨어와 소프트웨어 운영환경에 대한 위협 식별가능

4. 보안 요구사항 명세 방법 비교

UMLsec과 CLASP을 이용한 보안 요구사항 명세 방법과 본 논문에서 제시한 SEM 보안 요구명세 방법을 비교하면 <표 2>와 같다.

UMLsec의 보안 요구사항 명세는 use case 다이어그램과 use case명세를 이용하여 일반적인 객체지향 방법론의 요구사항 명세와 동일하다. CLASP의 보안 요구사항 명세는 보안 기능을 명세하는 자체기준을 제시하고 있으나 CC의 보안 기능 클래스에 비해 기준이 세분화되지 않았다.

또한 UMLsec과 CLASP의 방법은 위협모델링에서 소프트웨어 운영환경에 대한 위협의 고려가 불가능하다.

5. 결 론

RUP와 같은 기존의 소프트웨어 개발 방법에서는 보안 요구사항을 비기능적 요구사항의 일부로만 반영하였을 뿐 보안 요구사항을 소프트웨어 개발단계에 체계적으로 적용하지 못했다. 또한, 기존의 객체지향 방법론에서는 보안 요구사항을 비기능적 요구사항으로 인지하여 특별한 형식없이 명세하였다.

본 논문에서는 보안 요구사항 명세를 위한 단계를 제시하였다. 소프트웨어 보안 목표 설정과 운영환경 보안 목표 설정, 위협식별, 공격트리 작성, 보안기능 명세 단계를 수행함으로서 다양한 보안 기능을 고려한 보안 기능 명세가 가능하고 개발 초기단계에서 위협에 대한 분석을 수행하여 추후 분석과 설계단계에서 위협의 대응을 포함하는 데 도움을 줄 수 있다.

본 연구 결과를 발전시키기 위해서는 분석/설계 단계에서 정형화하여 보안 요구사항을 표현하기 위한 표준 모델링 기법에 관한 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] Gary McGraw, "Testing for Security During Development : Why We Should Scrap Penetrate-and-Patch", IEEE AES Systems Magazine, 1998.

[2] Jan Jürjens, UMLsec Presenting the Profile, 6th Annual Workshop On Distributed Objects and Components Security (DOCsec 2002), 2002.

[3] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison Wesley, 1999.

[4] Jan Jürjens, Using UMLsec and Goal Trees for Secure Systems Development, In the Proceedings of the 2002 ACM Symposium on Applied Computing, 2002.

[5] John Viega, Building Security Requirements with CLASP, In Proceedings of the 2005 workshop on Software engineering for secure systems-building trustworthy applications, 2005.

[6] CC, Common Criteria for Information Technology Security Evaluation Part1 : Introduction and general model, Version 2.1, CCIMB-99-031, 1999.

[7] CC, Common Criteria for Information Technology Security Evaluation Part2 : Security functional requirements, Version 2.1, CCIMB-99-032, 1999.

[8] David Aucsmith, The Digital Crime Scene : A Software Prospective, In Proceedings of the 1th CyberCrime and Digital Law Enforcement Conference, 2004.

[9] Bruce Schneier, Attack Trees, Dr. Dobb's journal, December, 1999.

김 기 한

1997년 중앙대학교 컴퓨터공학과(공학사)
1999년 중앙대학교 컴퓨터공학과(공학석사)
2001년 중앙대학교 컴퓨터공학과 박사과정 수료
2001년~현재 국가보안기술연구소 연구원

채 수 영

1989년 전북대학교 전산통계학과(이학사)
1999년 숭실대학교 정보통신 공학과(공학석사)
2006년 고려대학교 정보보호학과 박사과정 수료
2000년~2001년 한국정보보호진흥원 선임연구원
2001년~현재 국가보안기술연구소 선임연구원

최 명 렬

1991년 인하대학교 전자계산학과(공학사)
1993년 인하대학교 전자계산학과(공학석사)
1993년~2000년 국방과학연구소 선임연구원
2004년 인하대학교 전자계산공학과 박사과정 수료
2000년~현재 국가보안기술연구소 선임연구원

박 상 서

1991년 중앙대학교 전자계산학과(공학사)
1993년 중앙대학교대학원 전자 계산학과(공학석사)
1996년 중앙대학교대학원 컴퓨터공학과(공학박사)
1996년~1998년 국방정보체계연구소 선임연구원
1998년~1999년 국방과학연구소 선임연구원
2000년~현재 국가보안기술연구소 선임연구원