Journal of the Korea Convergence Society
Vol. 8. No. 6, pp. 17-27, 2017
https://doi.org/10.15207/JKCS.2017.8.6.017

공격시스템을 위한 보안-역-공격공학 생명주기 모델과 공격명세모델

김남정[1], 공문수[1], 이강수[1*]
[1]한남대학교 컴퓨터공학과

Security-Reverse-Attack Engineering Life-cycle Model for Attack System and Attack Specification Models

Nam-Jeong Kim[1], Mun-Soo Kong[1], Gang-Soo Lee[1*]
[1]Department of Computer Engineering, Hannam University

요 약 최근 사이버공격이 활성화됨에 따라 이러한 많은 공격사례들이 다양한 매체를 통해 접해지고 있다. 사이버 공격에 대한 보안공학이나 역공학에 대한 연구는 활발하지만, 이들을 통합하고 비용효과적인 공격공학을 통해 공격시스템을 연계하여 적용시킨 연구는 부족하다. 본 논문에서는, 보안강화형 정보시스템을 보안공학적으로 개발하고, 역공학을 통해 취약점을 식별한다. 이 취약점을 이용하여 공격공학을 통해 공격시스템을 구축하거나 리모델링하는 생명주기모델을 비교·분석하여 각 시스템의 구조 및 행동을 명세화하고, 더욱 실효성 있는 모델링을 제안한다. 또한, 기존의 모델·도구를 확장하여 공격방법 및 시나리오를 기능적, 정적, 동적과 같은 모델의 관점에서 명세하는 도형적 공격명세모델을 제시한다.

• 주제어 : 융합, 소프트웨어공학, 정보보호융합, 정보통신, IT응용기술

Abstract Recently, as cyber attacks have been activated, many such attacks have come into contact with various media. Research on security engineering and reverse engineering is active, but there is a lack of research that integrates them and applies attack systems through cost effective attack engineering. In this paper, security - enhanced information systems are developed by security engineering and reverse engineering is used to identify vulnerabilities. Using this vulnerability, we compare and analyze lifecycle models that construct or remodel attack system through attack engineering, and specify structure and behavior of each system, and propose more effective modeling. In addition, we extend the existing models and tools to propose graphical attack specification models that specify attack methods and scenarios in terms of models such as functional, static, and dynamic.

• Key Words : Convergence, software engineering, convergence of information protection, information communication, IT application technology

1. 서론

현대의 정보사회에서 ICT 기술의 발달로 인해 컴퓨터에 대한 의존도 높아지고 정보화의 역기능이 심화됨에 따라 안전한 정보시스템을 개발해야한다[1,2]. 따라서, 대부분의 정보시스템은 보안기능이 강화된 정보보호시스템이며 이를 비용-효과적으로 개발 및 운영하기 위한 보안공학 (security engineering)기술이 사용되고 있다.

보안공학기술은 기존의 소프트웨어공학기술을 정보보호분야에 맞도록 제작한 것이다. 보안공학기술을 통해 정보시스템을 개발하고 운영 할 때 시험과정을 통해 대부분의 취약점이 시스템 내에서 삭제되지만 일부 취약점은 잔류하여 공격자에 악용되어진다. 공격자는 정보시스템에 대한 역공학 (reverse engineering)을 통해 잔여 취약점을 식별하고 이를 악용 (exploitation)하여 시스템을 공격한다. 공격하는 시스템을 공격시스템이라 하며 공격공학 (attack engineering) 기술을 통해 개발할 수 있다. 현재 각 기술을 연계적으로 적용한 연구는 부족하다.

본 연구에서는 공격시스템을 효과적으로 구축하거나 기존의 정보시스템을 리엔지니어링하기 위한 생명주기 모델로서, 보안공학, 역공학 및 공격공학을 연계한 “보안-역-공격공학 생명주기 모델”을 제시한다. 또한, 공격명세 모델링을 위해 기존의 모델들을 확장하여 사용하는 방법을 제시한다.

본 논문의 2장에서는 보안공학, 보안 역공학 및 공격공학 연계형 생명주기 모델을 제안하고 각 과정에 대해 설명한다. 3장에서는 기존의 모델들을 확장한 공격명세 모델을 제안한다. 4장에서는 본 논문에서 제시한 생명주기 모델과 기존의 모델간의 차이점을 분석하며 공격명세 모델들을 평가한다. 끝으로. 5장에서 결론을 맺는다.

2. 보안-역-공격공학 생명주기 모델

일반적으로, 방화벽이나 VPN 등 보안 기능만을 제공하는 솔루션을 ‘정보보호제품’이라 한다. 최근의 정보시스템에는 대부부분 정보보호제품이나 독립적인 보안강화기능이 포함되어있으므로, ‘정보보호시스템’이라 볼 수 있다. 즉, 정보시스템과 정보보호시스템은 동일한 것으로 본다.

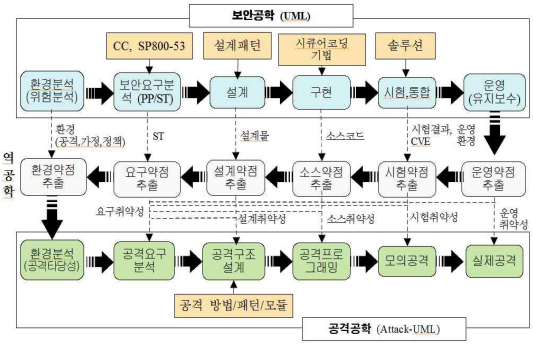
정보보호제품 또는 정보보호시스템은 개발 및 운영자에 입장에서 사이버 공격에 대한 방어대상시스템이라 할 수 있으며, 공격자 입장에서 공격대상 (ToA: Target of Attack) 시스템이 된다. ToA는 개발자가 ‘보안공학을 통해 개발 및 운영한다[3,4,5]. 공격자는 공격 준비단계에서 ToA 시스템의 운영취약점분석을 통해 운영환경의 취약점을 식별하고, ToA 자체의 취약점을 ‘보안 역공학’ 과정을 통해 식별한다. 공격자는 식별한 취약점을 악용하여 ToA를 공격한다.

효과적으로 공격하는 기술을 공격공학이라 정의하며 ToA가 개발되었고 운영되는 과정 (즉, 보안공학)상의 취약성을 고려하여 공격시스템을 개발한다. 그림 1은 본 연구에서 제시하는 보안공학, 역공학 및 공격공학 연계형 생명주기모델을 보인다.

2.1 보안 공학

전통적인 소프트웨어공학 기법과 보안요구사항공학, 보안패턴, 보안 아키텍처, 시큐어 코딩 등 기존의 보안공학 기술을 이용하여 ToA를 개발하는 과정이며, 사이버 공격에 대한 방어과정이라 볼 수 있다. 본 논문에서는 국제공통기준(CC) 패러다임을 이용한 보안공학을 제시한다. 그림 1처럼 환경 분석, 보안요구사항명세, 설계, 구현, 통합 및 시험, 운영 및 유지보수 과정을 거친다. 각 과정에서는 UML과 같은 모델을 이용하여 개발할 것이다.

보안공학을 완벽하게 수행하였고 (즉, 완벽한 보안보증), 운영환경에서도 약점이 없다면, ToA에는 취약성이 없고 공격의 여지가 없지만, 실제로는 이것이 불가능하다. 따라서, 공격자는 ToA의 취약성을 식별하여 공격할 수 있다.



[Fig. 1] Security engineering, reverse engineering and attack engineering linked life-cycle model

1) 환경분석 과정: 개발할 정보시스템의 ‘보안환경’ (보

안위협, 보안정책, 보안가정)을 분석한 후 '보안목적'(즉, TOE에 대한 보안목적, 환경에 대한 보안목적)을 도출한다. 필요시, 정보시스템이 보호할 자산에 가해질 정보보안위험을 분석하고 위험수준을 예측한다. 자산에 대한 보안위험이 높은 것을 식별하고 정보시스템을 운영할 조직의 보안정책에 따라 허용 가능한 위험수준을 결정한다.

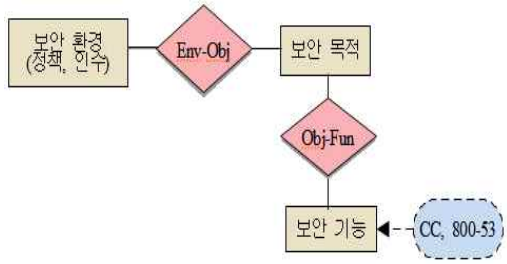
2) **보안요구사항분석 과정:** '보안목적'을 달성하기 위해, 공통적인 보안기능 또는 보안통제 목록 (예: CC내의 보안기능요구사항 집합, SP 800-53내의 보안통제 집합)으로부터, 자산의 보호에 필요한 보안기능요구사항을 도출하여 '보안요구사항명세서'를 작성한다[6,7,8]. 보안 제품일 경우, 제품유형별 공통 보안기능요구사항인 보호프로파일 (PP)을 제작하여 특정 제품별 보안요구사항 명세서인 보안보목표명세서 (ST)를 작성한다. 제품유형별 PP는 CC나 IT보안인증사무국과 같은 정보보호 제품인증기관 에서 제공하고 있다[9]. 그림 2는 보안요구사항의 스키마를 보인다.

3) **설계 과정:** 보안요구사항에 대해 구조설계와 상세설계를 실시한다. '설계패턴'이나 '보안패턴'을 이용 할 수도 있다. 예컨대, 식별 및 인증 요구사항을 설계하기 위해, 전자서명메커니즘, 인증 프로토콜, 암호모듈 등을 이용하여 설계할 수 있다[10,11].

4) **구현 과정:** 설계 결과를 코딩하는 과정이다. 구현 언어별 '시큐어코딩' 기법을 적용한다[12,13]. 2017년 3월 20일 현재 오버플로우와 관련된 취약성은 전체 취약성 82902개중 9227개 (11.3%)에 이른다[14]. 또한 주입(injection)과 관련된 취약성은 8033개 (9.6%)이다. 따라서, 시큐어코딩을 통해 오버플로우와 주입취약성만 제거하더라도, 최소한 20.9%의 취약성을 줄일 수 있다.

5) **통합 및 시험 과정:** 모듈단위의 단위시험, 통합시험, 시스템시험, 보안시험 및 인수시험을 실시하여 정보보증을 얻는다. CC 인증제품 등 기존의 정보보호 '솔루션'을 사용하는 경우, 통합을 위한 인터페이스를 개발한 후 통합시험을 한다. 침투시험이나 취약성분석과 같은 보안시험도 실시한다[15]. 보안시험은 취약성 스캐닝, 보안 스캐닝, 침투시험, 위험평가, 보안감사, 윤리적 해킹 (시스템 결함을 노출시킴), 자태 (posture) 평가 (여러 보안시험 방법을 통합) 등이 있다.

6) **운영 및 유지보수 과정:** 개발된 시스템을 실제 환경에서 운영한다. 운영결과는 운영프로파일로 표현된다. 운영 중에 예방, 오류수정, 환경적응, 기능개선 등의 유지보수 활동을 수행한다. 정보시스템 운영 중에 적용하는 보안대책은 주로 '관리적 대책'이며 가장 빈번히 발생하는 심각한 취약성이 된다. 예를 들어, 보안 솔루션의 유지보수자가 사용하는 노트북에서 바이러스 감염 또는 망 분리기능 중단 등은 빈번히 발생하는 취약성이다.



[Fig. 2] Security requirements scheme

2.2 보안 역공학

보안 역공학은 그림 1과 같이 개발 및 운영중인 ToA 시스템으로부터, 운영 취약성, 코드내의 취약성, 설계상의 취약성, 및 요구사항상의 취약성을 식별하는 과정이다. 사실, 각 과정은 순차적일 필요는 없으며 순서 없이 처리해도 된다. 식별된 취약성 정보는 공격을 위한 환경분석 및 요구사항명세 작성 시에 사용한다.

공격자가 공격지점이나 방법을 결정하기 위해 취약점을 파악하기 위해 실시하기도 하지만, 시스템의 감사(감리)자가 평가와 인증을 위해 실시하기도 한다. 운영중인 TOA 시스템에 대한 정보 (예: 소스코드, 운영프로파일, 구조도 등)는 사회공학 등을 통해 정보를 입수할 수 있다[16,17].

보안 역공학은 소프트웨어공학에서의 'V 모델'과 보안개발과정에서의 '보안 V 모델 개념을 확장한 것이며, '취약성 스택' 모델 (즉, 개발과 역공학시의 취약성이 개발중인 시스템 내에 누적되고 제거되는 현상을 나타냄)을 바탕으로 한다[18,19]. 역공학 과정은 기존의 보안 V 모델이서 시험과정들에 대응한다. 즉, '코드취약성추출 과정', '설계적취약성추출 과정', '요구사항적취약성추출 과정' 및 '환경적취약성추출 과정'은 기존의 V 모델이서 각각 단위시험, 통합시험, 시스템시험, 인수시험에 해당한다.

ToA의 요구사항명세서인 ST내의 보안환경부분 (즉,

자산, 위협, 취약성, 보안목적 등) 으로부터, 환경적 취약성을 분석한다. 즉, ST 작성 시 가정된 보안환경 부분과 시스템 개발 및 운영시의 실제 운영환경간의 차이는 환경적 취약성이 된다. 취약성은 취약면, 취약시간, 취약절차로 구분할 수 있다.

1) **운영취약성 추출 과정:** '운영프로파일'로부터 시스템이 실제 운영될 때의 취약성을 분석한다. 주로 관리적 보안 업무상에서 발생하는 취약성이다. 예를 들어, 보안패치를 늦게 할 때 발생하는 제로데이 취약성, 디폴트 패스워드를 수정안할 때의 취약성 등 대부분의 보안 사고는 운영 취약성 때문에 발생한다. Nikto, Wikto, Paros, Netsparker, Nessus, skipfish, Acunetix 등 네트워크, 시스템, 앱, 및 웹 취약성분석 도구들을 활용할 수 있다.

   오픈소스나 상용솔루션을 실장한 시스템의 경우, CVE, CWE 나 Exploit DATABASE에 나타난 각종 솔루션에 대한 취약성 및 약점을 활용한다[20,21,22].

2) **시험취약성 추출 과정:** 시험자료로부터, 누락된 시험자료, 시험결과의 잘못된 해석 등 시험시의 취약성을 추출한다. 즉, 개발 시 개발자에 의한 보안시험을 통해서도 발견되지 못했거나 잘못 검출된 취약성을 조사한다.

3) **코드취약성 추출 과정:** 만일 ToA의 소스코드를 얻을 수 있다면, 소스코드로부터, 코드에 포함된 비보안적 코드 (예: 포인터사용, 오버플로우를 야기할 수 있는 동적할당 명령 사용 등)를 분석한다. 만일 완전하게 시큐어코딩을 실시했다면, 코드 취약성은 없을 것이다. 시큐어코딩 규칙을 평가하는 '코드취약성 검출도구' 등을 이용할 수 있다[23].

4) **설계적 취약성 추출 과정:** ToA의 설계 문서와 보안설계패턴 등으로부터, 시스템의 아키텍처상의 취약성 (예: 비 도달성, 중복 모듈, 병목 구조, 약한 사슬 구조 등)을 분석한다.

5) **요구사항적 취약성 추출 과정:** PP나 ST와 같은 보안요구사항명세서로부터, 요구사항명세상의 취약성을 분석한다. 만일 요구사항이 검증 (validation)되었다면, 요구사항 내에 취약성은 없을 것이다. 또한 인증된 PP를 이용하여 ST를 오류없이 작성했을 때도 취약성은 없을 것이다.

6) **환경적 취약성 추출 과정:** ToA의 요구사항명세서인 ST내의 보안환경부분 (즉, 자산, 위협, 취약성, 보안목적 등) 으로부터, 환경적 취약성을 분석한다. 즉, ST 작성 시 가정된 보안환경 부분과 시스템 개발 및 운영시의 실제 운영환경간의 차이는 환경적 취약성이 된다.

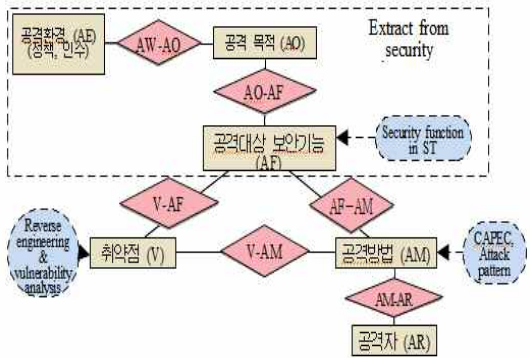
2.3 공격 공학

ToA를 공격할 때, 보안 역공학을 통해 식별한 취약점과 운영상의 취약점을 악용 (exploitation)하여 '공격시스템'을 구축한다. 공격시스템은 공격자의 행동 또는 이를 프로그래밍한 공격도구를 의미한다. 최소한의 비용으로 최대의 공격효과를 얻도록 공격시스템을 개발하는 방법을 '공격공학'이라하며 '보안공학'과 연관되는 개념이다.

공격공학은 ToA를 공격할 때 뿐 아니라, 보안성을 높이기 위해 재공학 또는 구조조정 (re-engineering, re-structuring)할 실시할 때도 필요하다. 즉, ToA내의 취약성을 없애고 기능을 효과적으로 보완하여 ToA를 재구축한다. 또한, 침투시험을 이용해 보안성을 평가할 때도 공격공학이 필요하다.

1) 공격환경분석 과정 (ATE: Attack Environment)

공격의 법적, 경제적, 기술적 타당성을 분석하고 공격환경 (즉, '공격정책', '공격가정')과 '공격목적'을 명세한다. 공격정책과 공격가정은 타당성분석 결과, ToA의 보안정책, 보안가정으로부터 도출한다. 그림 3의 공격스키마의 윗부분에 해당한다.



[Fig. 3] Attack scheme of attack system

- 공격정책: 공격자가 공격 시 따라야할 공격조직이 결정한 행동방침이다. 여러가지 형태 (예: 법률, 정책, 사업, 사업계획, 방침, 지침, 결의사항 등)로 표현한다. 공격정책은 다시 공격전략 (strategy)으로 구체화하고, 공격전술은 공격전술 (tactic)로 구체화할 수 있다. 공

격전략(지침, 규약)은 공격목적에 해당하며 공격전술은 공격방법에 해당한다.

- 공격가정: 공격 시 공격환경, 공격자 및 방어자(즉, ToA의 보안기능)에 대한 가정이다. 가정을 함으로서, 공격의 범위를 단순화할 수 있다. 예를 들어, "공격자와 방어자는 공격시나리오대로 공격한다.", "공격자와 방어자는 협상하지 않는다." 등
- 공격목적 (AO): TOA의 보안목적을 부정 및 손상시키는 것이 공격목적이 된다. 즉, ToA의 무결성, 기밀성, 가용성, 부인봉쇄, 인증성, 인가성을 손상시키는 것이다. 일반적으로, 시스템의 보안목적은 시스템 자체의 보안목적과 운영환영에 대한 보안목적으로 구분하므로, 공격목적은 'TOA 자체의 공격목적'과 'TOA의 운영환경에 대한 공격목적'으로 구분한다.
- 공격환경-공격목적 (AE-AO) 관계: 공격환경으로부터 공격목적을 도출하므로, 공격환경과 공격목적 간의 n:m 대응관계를 나타낸다.

  필요시, ToA의 공격면 (attack surface)을 측정하여 상대적인 공격가능성 (attackability)을 구할 수 있다 [24,25]. 또한, ToA가 CC 평가를 받은 보안제품 또는 이를 이용한 시스템이라면, ToA의 보증수준 (EAL)값은 절대적인 공격가능성의 척도로 활용할 수 있다. 즉, EAL1은 공격가능성이 가장 쉽고, EAL7은 가장 어렵다.

2) 공격요구사항분석 과정 (ATR: Attack Requirement)

ToA내 무엇을(what) 공격할 것인가를 정의한다. 앞 단계의 결과인 공격목적을 달성하기 위한 전략에 해당한다. ToA의 보안요구사항명세서 등을 통해, '공격목적'으로부터 '공격대상기능'을 도출한다. 공격대상기능 명세서는 공격요구사항 명세서가 되며, 곧, ToA 내의 보안기능에 해당한다.

mis-use case 다이어그램을 사용하여 공격자가 실행할 공격기능 (예: 침입, 서비스거부 야기 등)을 명세한다. mis-use case 다이어그램 중 일반 use case는 정상액터 (즉, 사용자, 관리자)에게 제공하는 보안기능 (즉, 보안서비스)에 해당하며, mis-use case는 공격자가 행하는 공격기능이다. 그림 3의 공격스키마의 가운데 부분에 해당한다.

- 공격대상기능 (AF): 공격자는 ToA 내의 보안기능을 공격하는 셈이므로, 공격대상 (타겟)기능이 된다. 공격대상기능이 수행되는 위치는 IP 주소나 URI로 나타낸다. 예를 들어, ToA가 CC내의 "FIA_AFL.1 인증 실패처리" 기능을 구현하였다면, 이를 무력화하는 것이 곧 공격대상기능이다. 공격자는 TOA내의 '인증실패처리' 기능의 취약점을 공격할 것이다.
- 공격목적-공격대상기능 (AO-AF) 관계: 공격목적으로부터 공격대상기능을 도출하므로, 고객목적과 공격대상기능 간의 n:m 대응관계를 나타낸다.

3) 공격구조설계 과정 (ATS: Attack Structure)

공격대상기능에 대해 어떻게 (how) 공격할 것인가를 설계한다. ToA로부터 보안 역공학을 통해 얻은 '설계취약점'과 운영프로파일을 통해 얻은 '운영취약점'을 이용하여 상위수준의 공격방법을 설계한다. 상위수준 공격방법은 CAPEC 등과 같은 공격패턴 등을 이용하여 설계하며. 상세한 공격 방법이나 공격시나리오의 골격을 설계한다. 각종 공격명세모델을 이용하여 명세하며 다음 장에서 상세히 논할 것이다.

- 취약성(V): 취약성은 앞에서 논한 ToA의 운영취약성 분석과 보안 역공학으로부터 식별한다. ToA내에 상용 솔루션이 포함되었다면, CVE나 ExploitDATABASE로부터 그 솔루션의 취약성을 파악할 수 있다.
- 공격자 (AR): 공격자는 공격방법 (공격시나리오)에 따라 공격을 실시하는 사람 또는 프로세서이다. 그림 3의 공격스키마내의 공격자 엔티티에는 IP 주소와 같은 공격시작위치 속성이 포함된다. 또한, 공격자가 가진 공격용 자원, TOA로의 접근기회, 공격 동기, 공격 전문성, 위험혐오수준 (aversion) 등의 속성을 가질 수 있다. SurfWatch Lab에서는 공격자를 9가지로 분류하고 있다[26]. 국가지원, 개인, 핵티비스트, 사이버 테러리스트, 조직화된 범죄, 미지의 신분, 조직, 정보보안, 법률강화/당국. NIST SP-800-30-R1 에서는 인간 공격자(Adversal)를 다음과 같이 분류한다[27]. 개인 (외부자, 내부자, 신뢰된 내부자, 특권을 가진 내부자), 그룹 (Ad hoc, 설립된 그룹), 조직 (경쟁자, 공급자, 파트너, 고객), 국가.
- 공격방법 (AM): 공격방법은 원칙적으로 공격대상보안기능 (AF)과 취약성 (V)으로부터 도출한다. TOA내의 공격대상기능이나 취약성에대한 정보가 부족할 때, CAPEC과 같은 기존의 공격패턴 등을 활용할 수 있다 [28]. 공격방법에 대한 명세방법은 다음 장에서 논한다. 공격시작 IP, 타겟 엔티티에는 목표 IP 주소가 포함되

며 공격방법에 시작시간(AST)과 종료시간 (AET)이 포함된다. 공격시간구간 (AD)은 AD = AET - AST 와 같다.

- 공격대상기능-취약성 (AF-V) 관계: 공격대상기능으로부터 취약성이 도출되므로, 이들 간의 n:m관계를 나타낸다.
- 공격대상기능-공격방법 (AF-AM) 관계: 공격대상기능으로부터 공격방법이 도출되므로, 이들 간의 n:m관계를 나타낸다.
- 취약성-공격방법 (V-AM) 관계: 취약성으로부터 공격방법이 도출되므로, 이들 간의 n:m관계를 나타낸다.
- 공격방법-공격자 (AM-AR) 관계: 공격방법과 공격자 간의 n:m관계를 나타낸다.

4) 공격프로그래밍 과정 (ATP: Attack Programming)

공격구조설계 결과를 수행 가능한 공격시나리오, 공격 스크립트, 또는 공격프로그램으로 변환하여 공격시스템 (또는 공격엔진)을 구현한다. 공격시스템은 제어기에 해당하며 실행기 (액추에이터)는 공격을 실행하는 장치 또는 사람이다.

5) 모의공격 과정 (SAT: Simulated Attack)

공격자 내부 환경에서 모의 공격해보는 '알파공격', 베타공격 (실제 환경에서 모의 공격대보는 '베타공격'을 실시하여 공격의 성공가능성을 분석한다.

6) 실제공격 과정 (RAT: Real Attack)

공격시스템을 실제로 실행하는 것이다. 공격 후 증거 삭제 등의 후속작업도 수행한다.

3. 도형적 공격명세 모델

'공격명세 모델'을 이용하여 '공격방법'을 표현한 것을 '공격명세'라하며, 유사한 공격방법의 틀을 '공격패턴'이라한다. 공격방법은 다양한 관점이나 스키마를 통해 표현될 수 있으므로, 용도와 문맥에 맞도록 공격방법을 명세해야한다. 공격명세와 공격패턴은 공격자 뿐 아니라, 보안공학을 통해 시스템을 개발할 때 모든 생명주기 과정에서 사용된다. 건물을 지을 때 지진정보 (즉, 공격패턴)를 모든 건축 단계에서 활용하는 것과 같다.

본 장에서는 2장의 '공격구조설계' 과정에서 사용할 수 있는 공격명세방법을 비교 및 제안한다. 공격명세는 ToA의 특징과 공격방법에 따라, 다양한 관점 (또는 문맥)에서 명세화 할 수 있다. 관점별 명세모델은 다음과 같다.

- '기능'모델은 공격시스템의 기능, 프로세스 또는 서비스를 중심으로 명세하며 공격-트리, 공격-유스케이스, 공격-활동도, 공격-자료흐름도, 공격-구조도가 있다.
- '동적'모델은 시간과 사건에 의한 상태의 변화를 중심으로 명세하며 공격-그래프, 공격-시나리오, 공격-순차도, 공격-상태전이도, 공격-간트챠트, 공격-패트리넷이 있다.
- '정보'모델은 공격시스템의 자료구조나 객체를 중심으로 명세하며 공격-클래스다이어그램, 자료사전, 공격-ER다어그램이 있다.
- '융합'모델은 ToA의 구조에 공격시나리오를 스크립트 형태로 명세하며 미스유스케이스맵, 위협/공격모델링 다이어그램이 있다.

본 연구에서는 기존의 모델과 구분하기 위해 '공격' 단어를 추가하였다.

3.1 UML 기반 모델

공격-유스케이스와 공격-시나리오, 공격-활동도, 공격-순차도, 공격-클래스 다이어그램과 공격- ER다이어그램, 공격-상태전이도는 UML에 공격모델링을 할 수 있도록 확장한 것이다. UML은 시스템의 정적, 동적 및 정보구조를 모델링할 수 있는 다양한 도형적 모델로 구성되어 있으므로, 공격개념을 UML에 추가할 수 있다.

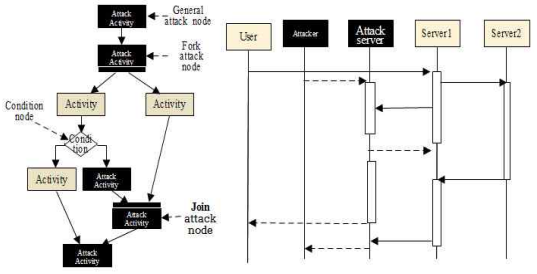
1) 공격-유스케이스와 공격-시나리오모델: 기존의 미스-유스케이스모델처럼 공격유스케이스 (즉, 공격기능 또는 공격서비스를 나타내며 음영으로 표시) 와 공격자 액터 (정상 액터와 구분하여 표시)를 추가하고 이들 간의 인터페이스 및 관계(<<완화관계>>, <<위협관계>>, <<포함관계>>, <<확장관계>> 등)를 표시한다[29,30]. 공격-유스케이스 모델은 정적인 공격상황을 모델링하므로, 상세한 공격 내용은 별도의 공격-유스케이스 표에서 '공격-시나리오'를 기술한다.

공격-유스케이스내의 속성은 이름, 요약, 저자, 날짜, 기본경로, 대안경로, 완화(mitigation) 점, 확장점, 트리거, 전제조건, 가정, 완화 보장, 관련 비즈니스규칙, 잠재적 공격자 프로파일, 관련자 및 위협, 용어 및 설명, 스코프, 추상수준, 정밀도 등으로 구성한다. 여기

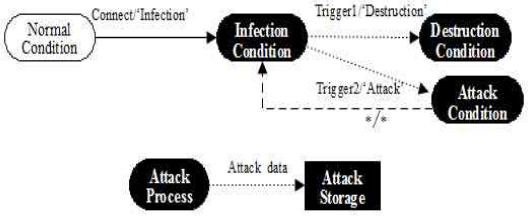
서, '기본경로'와'대안경로'부분이 공격시나리오에 해당하며 구문적으로 작성한다.

2) **공격-활동도:** fork/join 흐름을 확장한 순서도이며 공격자가 취할 공격활동과 그 순서를 모델링한다. 특히, 공격활동은 일반 활동과 구분하여 음영으로 표현한다. 공격공학에서 가장 많이 사용할 모델이다.
3) **공격-순차도:** 공격자와 다른 주체 (서버 등)간의 메시지 흐름을 인터페이스를 모델링하며 공격자나 공격서버는 역상처리한다.
4) **공격-클래스 다이어그램과 공격- ER다이어그램:** 공격자 클래스 (또는 엔티티)가 포함된 클래스다이어그램 (또는 ER다이어그램)이다. '공격자클래스'는 공격자 속성 (예: 공격자명, 동기, 역량 등)들과 공격메소드(예: 공격자가 할 수 있는 여러 가지 공격 메소드)로 구성한다. 공격자 클래스와 '공격대산클래스' 간에는 '관계클래스'가 존재할 수 있다.
5) **공격-상태도:** 전통적인 상태전이도에서 '공격상태'를 별도로 모델링한 것이다. 상태 간에는 이벤트와 액션을 모델링할 수 있다. 상태는 역상 처리하며 전이화살표는 점선으로 표시한다. 공격공학에서 많이 사용할 모델이다.

3.2 구조기반 모델



[Fig. 4] Attack – Activity and Attack - Sequence



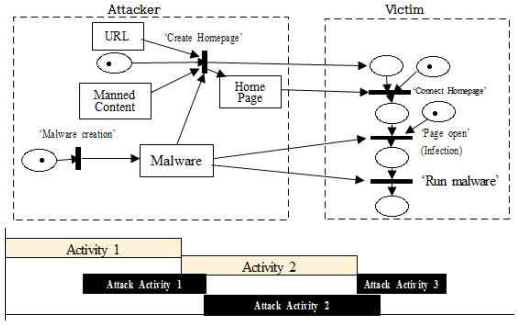
[Fig. 5] Attack – State Transition and Attack – Data Flow

1) **공격-트리와 공격그래프[31,32]:** 가장 활발히 연구되는 모것이며 모듈 (즉, 서브루틴)간의 And/Or 관계 및 호출관계와 파라미터를 모델링한다. 공격시스템 개발 시 공격시스템의 구조설계도로 사용할 수 있다.
2) **공격-구조도:** 기존의 구조도를 공격모델링에 활용한 델이며 공격벡터 (또는 공격방법) 간 또는 공격원인과 결과 간의 And/Or 계층관계를 모델링한다.

3.3 흐름기반 모델

1) **공격-자료흐름도:** 기존 자료흐름에 공격노드 (즉, 공격프로세스), 공격저장소 노드, 공격 자료와 화살표를 추가한 것이다. 노드는 역상처리하며 화살표는 점선으로 표시한다. 공격프로세스의 세부사항은 별도의 Mini SPEC이나 순서도로 명세하며, 자료는 별도의 자료사전에서 명세한다.
2) **공격-패트리넷[33,34]:** 패트리넷을 이용해 사이버공격을 모델링하기 위한 것이다. 구조에 제한을 두고나 확장하면 상태머신부터 튜링머신까지 정형적으로 모델링이 가능하다. 플레이스나 트랜지션에 변수개념, 시간개념, 확률을 확장하면 각각 컬러드, 시간 및 스토케스틱 패트리넷이 된다. 확장할수록 모델링 능력은 우수하지만 분석이 어려워진다.

3.4 통합형 모델



[Fig. 6] Attack – Examples of Attack Specification Using Patrinet and Attack – Gantt Chart

1) **공격-간트차트:** 공격활동을 역상처리하며 공격활동 간의 템포럴 로직을 명세할 수 있다.
2) **미스유스케이스 맵(공격-유스케이스 맵)[35]:** 컴포넌트의 도메인 구조상에 공격시나리오를 직접 표현한 것이다. 여러 가지 공격행동을 표시할 수 있다.

3) 위협/공격 모델링 다이어그램[36]: 컴포넌트 도메인 구조에 위협(공격자와 악성관리자)과 공격벡터(인젝션, 파일조작, 중간자공격, 직접 DB 조작 등)를 명세한 것이다.

4. 분석 및 평가

4.1 기존 보안생명주기 모델과의 비교

기존의 연구들에서는 전통적인 폭포수형 생명주기 모델에 보안성을 높이기 위한 보안공학활동을 추가하였다.

MS사의 SDL (security development lifecycle)프로젝트에서는 2017년 현재 훈련부터 대응에 이르기까지 7단계에 적용할 수 있는 17가지의 보안공학방법에 대한 지침 및 도구를 제공하고 있다[24]. 이들 중 '공격면(surface)분석', '퍼징 (fuzzing)을 이용한 침투시험' 등은 특이한 방법이다. 본 연구의 앞부분인 보안공학 부분의 각 과정에서 적용할 수 있을 것이다.

S2e (Secure Software engineering)에서는 본 연구와 가장 유사하며 관리 관점, 화이헤드 관점 및 블랙헤드 관점을 추가하였다[37]. '관리 관점'은 위험관리 등 보안 관리에 해당하며, '화이트헤드 관점'은 개발자의 관점으로서 전형적인 보안공학과정에 해당한다. 특히, '블랙헤드 관점'은 블랙햇 정보 분석 및 관리, 공격 모델링, 침입 및 공격공학, 악용, 코드 역공학, 쉘 코드 개발, 악성로직 방법들로 구성된다. 이는 본 연구에서 역공학의 일부분(코드역공학)과 공격공학 부분과 유사하다. 본 논문처럼 '공격공학'이라는 용어를 사용하였고 각 방법에 대한 세부적인 내용은 없으며, 본 논문처럼 보안공학, 역공학 및 공격공학의 각 과정간의 관계에 대해 기술하지 않았다.

AEGIS (appropriate and effective guidance for information security)에서는 자산식별 및 분류 후, 각 자산에 대한 가치평가, 위협/취약성/위험을 식별하고 대응책(즉, 보안기능구조)을 설계하고 비용-효과성을 평가하여 개선하였다[38].

이는 전형적인 위험분석 및 CC 패러다임이며 본 연구의 보안공학부분과 유사하다. OWASP (open web application security project) 에서는 위험분석과 침투시험 등 14개의 전형적인 보안공학방법과 4개로 구성된 보안품질게이트 개념을 제시하고 있다[39]. 본 연구의 앞부분인 보안공학 부분의 각 과정에서 적용할 수 있을 것이다. SEI의 TSP-Secure은 TSP (team software team)와 'Agile 모델'에 팀구성원의 전문화, 프로젝트 관리, 재사용, 품질보증, 검사, 정적분석, 보안, 문서 및 교육 업무를 추가하였다[40,41]. OWASP와 같이 위험관리 외 시큐어 코딩 등 전형적인 보안공학방법을 제공한다. SSDM (security SW development model)에서도 위협모델링과 침투시험 등 5가지 보안공학 프로세스를 추가하였다[42]. IGI의 SSE (SW security engineering)에서도 각 단계별로 위협분석, 코드 리팩토링 등 전통적인 보안공학 방법을 추가하였다[43].

4.2 공격명세 모델의 분석

3장에서 제시한 도형적 공격명세모델들은 잘 알려진 기존의 UML 모델을 공격명세를 위해 단순하게 확장한 것이며 다음과 같은 장점을 가진다.

- 거의 모든 도형적 명세모델을 포함하고 있다.
- 친숙한 모델이므로, 새롭게 익힐 필요가 없고 기존의 도구를 활용할 수 있다.
- 공격시스템을 모델링할 때, 공격의 구조수준 (예: 공격-구조도, 공격-유스케이스)부터 상세수준 (예: 공격-활동도)까지 필요한 수준에 맞는 모델을 선택할 수 있다.
- 공격모델의 관점 (예: 기능적, 동적, 정보적)에 따라 적합한 모델을 택할 수 있다.
- 제시한 개념을 확장하여 필요에 따라 각 모델을 추가로 확장할 수 있다.
- 공격시스템도 일정의 정보시스템이므로, 공격명세 모델들을 기존의 소프트웨어공학 방법과 함께 사용하여 시스템을 구축할 수 있다.

5. 결론

본 논문에서는 기존의 보안공학, 역공학 및 공격공학 개념을 연계한 생명주기 모델을 제시하였다. 즉, 보안기능이 강화된 정보시스템을 개발한 후, 운영상의 취약점을 식별하고, 시스템에 대한 역공학을 통해 개발상의 취약성을 식별한다. 이를 이용하여 시스템을 보안성 높게 리엔지니어링하거나 공격시스템을 구축하여 시스템을 공격한다. 비용 효과적으로 공격하는 개념을 공격공학이라 하며 보안공학에 대비되는 패러다임이다. 또한, 기존의 명세모델들을 조사하고 공격을 명세할 수 있도록 확장하는 방법을 제시하였다.

제시한 연계형 생명주기모델을 활용한다면, 보안공학

을 통해 비용-효과적으로 보안성 높은 정보보호시스템을 개발할 수 있고, 공격자는 역공학을 통해 개발된 시스템의 취약점을 체계적으로 파악할 수 있으며, 이를 이용하여 비용-효과적으로 공격시스템을 구축하거나 기존의 시스템을 리엔지니어링할 수 있을 것이다. 또한, 용도와 관점에 맞는 공격명세모델을 선택하여 사용함으로써 명확하고 효과적으로 공격방법을 명세할 수 있을 것이다.

제시한 생명주기모델을 실제로 적용하여 실효성을 입증하고 개선하며, 제시한 공격명세모델을 개선하는 일은 향후 연구과제로 남긴다.

ACKNOWLEDGMENTS

본 논문은 2016년도 한남대학교의 교비학술연구지원(2016A222)을 받아 수행된 것임.

REFERENCES

[1] J. W. Jung, J. D. Kim, Myeong-Gyun Song, Chul-Gu Jin, “A study on Development of Certification Schemes for Cloud Security”, Journal of digital Convergence , Vol. 13, No. 8, pp. 43-49, 2015.

[2] M. S. Gu, YongZhen Li, "A Study of Countermeasures for Advanced Persistent Threats attacks by malicious code," Journal of IT Convergence Society for SMB, Vol. 5, No. 4, pp. 37-42, 2015

[3] J. H. Allen, S. Barnum, Robert J, Software security engineering - A guide for project managers, Addison-Wesley Professional, pp. 315, 2008.

[4] M. Ramachandran, Software Security Engineering - Design and applications, Nova Science Publishers, Inc., p. 272, 2012.

[5] R. Ross, M. McEvilley, J. C. Oren, Systems security engineering - Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems, NIST SP 800-160 , pp. 242, 2016.

[6] Common Criteria for Information Technology Security Evaluation, Part1, Part2, Part3 Version 3.1, Revision 4, CCRA, 2012.

[7] Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 4, CCRA, 2012.

[8] Kelley Dempsey, Security and Privacy Controls for Federal Information Systemsand Organizations, NIST SP 800-53 Revision 4, 2013.

[9] http://www.commoncriteriaportal.org.

[10] E. Gamma, et al., Design patterns - elements of reusable object-oriented software, Addison-Wesley, pp. 431, 1995.

[11] C. Dougherty, Secure Design Patterns, SEI, CMU, 2009.

[12] C Secure Coding Guide for e-government SW Development - Operation, Ministry of the Interior, 11-1311000-000330-10, pp. 212, 2012.9.

[13] Java Secure Coding Guide for e-government SW Development - Operation, Ministry of the Interior, 11-1311000-000330-10, pp. 320, 2012.9.

[14] https://cve.mitre.org/cve.

[15] https://www.owasp.org,/index.php/OWASP_Testing_Guide_v4_Table_of_Contents 2017.

[16] J. H. Kim, J. Y. Go, K. H. Lee, “A Scheme of Social Engineering Attacks and Countermeasures Using Big Data based Conversion Voice Phishing”, Journal of The Korea Convergence Society“, Vol. 6, No. 1, pp. 85-91, 2015

[17] H. S. Yang, “A Study on Multi-level Attack Detection Technique based on Profile Table”, Journal of The Korea Society of Digital Industry and Information Management, Vol. 10, No. 4, pp89-96, 2014

[18] https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html.

[19] K. M. Goertztel, et al., Software security assurance, IATAC and DACS, 2007.

[20] https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=overflow.

[21] https://cve.mitre.org/data/downloads/index.html.

[22] https://www.exploit-db.com/.

[23] http://www.cert.org/secure-coding/tools/.

[24] https://www.microsoft.com/en-us/SDL.
[25] P. Manadhata and J. M. Wing, "An Attack Surface Metric," IEEE Transactions on Software Engineering, Vol. 37, Vo. 3, 2011.
[26] https://www.surfwatchlabs.com/.
[27] R. M. Blank, Guide for Conducting Risk Assessments, NIST SP 800-30, 2012.
[28] https://capec.mitre.org.
[29] Alexander I. "Misuse Cases: Use Cases with Hostile Intent," IEEE Software, Vol. 20, No. 1, pp.58-66, 2003.
[30] Sindre, G., Opdahl A.L. Eliciting Security Requirements with Misuse Cases. Requirements Engineering 10(1), pp. 34-44, 2005.
[31] Barbara Kordy, Ludovic Piètre-Cambacédès, Patrick Schweitzer, DAG-Based Attack and Defense Modeling: Don' Miss the Forest for the Attack Trees, Computer Science Review, Vol. 13, pp. 1-38, 2014.
[32] J. H. Eom, Park, S. H, Chung, Tai M, "A Study on an Extended Cyber Attack Tree for an Analysis of Network Vulnerability", Journal of the Korea Society of Digital Industry and Information Management, Vol. 6, No. 3, pp. 49-57, 2010
[33] G. Lee, J Lee, "Petri Net based Models for Specification and Analysis of Cryptographic Protocols", The Journal of Systems and Software, Vol. 37, pp. 141-159, 1997.
[34] Yongfu Zhou, "The Network Attack Model based on Hierarchical Expanded Stochastic Petri Net", International Journal of Security and Its Applications, Vol.8, No.6, pp.161-172, 2014.
[35] Peter Karpati, Guttorm Sindre, "Towards a hacker attack representation method", Proceedings of the 5th International Conference on Software and Data Technologies, pp. 92-101, 2010.
[36] https://capec.mitre.org/documents/An_Introduction_to_Attack_Patterns_as_a_Software_Assurance_Knowledge_Resource.pdf.
[37] Schneider, Thorsten, "Secure Software Engineering Processes: Improving the Software Development Life Cycle to Combat Vulnerability", Software Quality Professional 8, no. 1, 2006.
[38] I. Flechais, C. Mascolo, M. Angela Sasse, "Integrating Security and Usability into the Requirements and Design Process", International Journal of Electronic Security and Digital Forensics, Vol. 1, Issue 1, pp. 12-26, 2006.
[39] https://www.owasp.org/images/7/76/Jim_Manico_(Hamburg)_-_Securiing_the_SDLC.pdf.
[40] http://resources.sei.cmu.edu/asset_files/whitepaper/2013_019_001_297287.pdf.
[41] http://resources.sei.cmu.edu/asset_files/presentation/2016_017_001_493912.pdf.
[42] A. S. Sodiya,S. A. Onashoga, O. B. Ajayi, "Towards building secure software systems", Proceedings of Issues in Informing Science and Information Technology, Vol. 3, pp. 637-644, 2006.
[43] M. Zulkernine and S. I. Ahamed, Software Security Engineering: Toward Unifying Software Engineering and Security Engineering, Enterprise Information Systems Assurance and System Security: Managerial and Technical Issues, pp. 19, 2006.

저자소개

김 남 정(Nam-Jeong Kim) [정회원]

- 2016년 2월 : 한남대학교 컴퓨터공학과 졸업
- 2016년 3월 ~ 현재 : 한남대학교 컴퓨터공학과 석사

<관심분야> : 소프트웨어공학, 융합보안, 역공학, 보안공학

공 문 수(Mun-Soo Kong) [정회원]

- 2016년 2월 : 한남대학교 컴퓨터공학과 졸업
- 2016년 3월 ~ 현재 : 한남대학교 컴퓨터공학과 석사

<관심분야> : 소프트웨어공학, 가상현실, 웹 보안, 사물인터넷

이 강 수(Gang-Soo Lee) [정회원]

- 1983년 2월 : 서울대학교 전산학(이학 석사)
- 1989년 2월 : 서울대학교 전산학 박사 (이학 박사)
- 1992년 7월 ~ 1993년 7월 : 일리노이대 교환교수
- 1995년 : ETRI 부호기술부 초빙연구원
- 1987년 3월 ~ 현재 : 한남대학교 컴퓨터공학과 교수
- 2000년 1월 ~ 현재 : 한국 디지털콘텐츠학회 이사, 한국멀티미디어학회 이사

<관심분야> : 소프트웨어공학, 보안공학, 멀티미디어 교육, 제어시스템 사이버보안