제33회 한국정보처리학회 춘계학술발표대회 논문집 제17권 제1호 (2010. 4)

웹서비스의 사이버 위협을 줄이기 위한 소프트웨어 개발 방법론

김지용*, 이송희**, 최진영***
*고려대학교 컴퓨터정보통신대학원 소프트웨어공학과
**고려대학교 컴퓨터·정보통신 연구소
**고려대학교 컴퓨터·통신공학부
e-mail : criskim@mopas.go.kr, {shlee, choi}@formal.korea.ac.kr

A Web Service Software Development Lifecycle to Reduce Cyber Attack

Ji-Yong Kim*, Song-Hee Lee**, Jin-Young Choi***
*Dept. of Software Engineering, Korea University
**Institute of Computer. Information and Communications, Korea University
***Dept. of Computer Information & Communication, Korea University

요 약

우리나라의 전자정부 서비스 대부분은 웹 환경을 기반으로 하고 있으며, 이는 국민생활의 질적인 향상을 가져온 반면 개인정보의 수집, 활용, 유통이 급격히 증가되면서 각종 프라이버시침해, 사이버 위협 등의 부작용이 수반되고 있으며 웹 브라우저의 다양화에 따른 웹접근성이 해결되어야 하고, 장애우 차별을 금지하기 위한 서비스가 동시에 제공되어야 한다. 따라서 웹서비스를 대상으로 하는 사이버 위협에 대한 보안대책과 다양한 서비스에 대한 대책이 중요한 요소로 부각되고 있다. 웹 환경에서의 보안문제를 해결하기 위해 기존에는 방어벽등 보안 모듈 부분을 강화하는데 초점을 맞추었으나, 본 논문에서는 그에 국한되지 않고 소프트웨어 개발초기 단계인 분석, 설계단계에서부터 보안문제를 고려하여 운영단계에 이르기까지 보안 취약점을 해결할 수 있는 방안을 제시하였다.

1. 서론

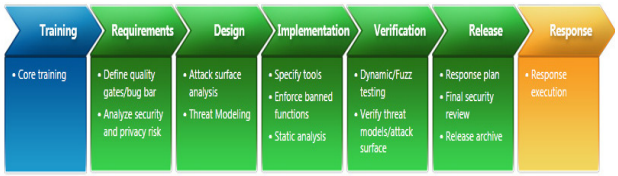
전자정부 11대 과제와 31대 로드맵 과제를 거치며 전자정부 대부분의 서비스는 웹 환경을 기반으로 만들어졌으며, 편리한 웹 접근성을 제공함으로써 국민생활에 질적인 향상을 제공하고 있다. 그러나 최근 발생하고 있는 사이버 위협의 80%이상이 웹 서비스를 대상으로 하는 해킹공격으로 웹서비스에 대한 보안 대책 마련이 중요한 요소로 부각되고 있다[1].

SANS(SysAdmin, Audit, Network, Security)기관의 보고서[2]에 따르면, 어플리케이션 취약점이 OS 취약점을 상회하였다고 한다. 이는 지금까지 알고 있었던 OS나 네트워크가 아닌 어플리케이션 공격이 많아졌음을 알 수 있다. 특히 “Injection”과“XSS(Cross Site Scripting)"에 의한 공격에 취약한 국내 웹서버의 수는 최소한 50%이상이다[3]. 기존에는 이러한 보안문제를 해결하기 위해 방화벽을 추가 제공하거나 보안 모듈을 강화하였으나 이는 근본적인 해결책을 제시하지 못하고 있다. 미 국토안보부에서는 사이버위협의 근본적인 보안 취약점을 가능한한 제거하기 위해서는 소프트웨어 개발 단계시 보안 요소를 강화하는 시큐어 소프트웨어 개발 생명주기(Secure SDLC: Secure Software Development Lifecycle)를 추천하고 있으며[4], 마이크로소프트사에서는 수년전부터 자체적인 보안개발생명주기(MS-SDL: Security Development Lifecycle)를 개발하여 모든 제품 개발시 이를 적용하고 있다[5]. 마이크로소프트사는 MS-SDL을 적용함으로써 이전에 비해 보안 취약점을 50%~60% 줄이고 있다[6]. 따라서 본 논문에서는 해킹과 같은 사이버 위협을 줄이기위해 소프트웨어 개발 초기 단계인 분석, 설계단계에서부터 보안문제를 고려하여 운영단계에 이르기까지 보안 취약점을 해결하고, 장애인 및 고령자와 같은 정보 취약계층의 웹 접근성에 도움을 줄 수 있는 방안을 제시한다.

논문의 구성은 다음과 같다. 2장에서는 관련연구로서 MS-SDL방법론을 소개하고 3장에서는 제안된 방안을 상세히 설명하고, 4장에서 결론을 맺는다.

2. MS-SDL 방법론

MS-SDL은 개발 생명주기 내에 단계별 주체의 보안 활동을 통하여 소프트웨어 보안을 강화하는 방법으로 그림 1에서 보여진다.



(그림 1) MS-SDL

그림 1에서 보여지는 바와같이 MS-SDL은 Pre-SDL(교육) 단계, 요구사항 단계, 설계 단계, 구현 단계, 검증 단계, 릴리즈 단계, Post-SDL(대응) 단계로 구성되어 있다[6]. 각 단계별 세부 내용은 다음과 같다.

2.1 Pre-SDL 단계

보안 교육은 소프트웨어 개발팀의 구성원들이 보안 기초와 최신 보안 동향에 대한 정보를 교육받을 수 있는 기회를 제공한다. 소프트웨어 프로그램을 개발하는 인력은 최소 매 년 한 번의 보안 교육에 참가해야 한다. 소프트웨어가 보안성을 위배하지 않고 안전하게 동작하고 하는 것을 보장하기 위해서는 보안 교육이 기초가 되기 때문이다. 보안 교육은 시큐어 설계, 위협모델링, 시큐어 코딩, 보안 테스팅, 프라이버시 등의 내용이 포함된다.

2.2 요구사항 단계

프로젝트 개시와 함께 신뢰성 있는 소프트웨어를 구축하기 위하여 기본 보안 요구사항과 프라이버시 요구사항을 정의한다. 또한 비용 분석을 통하여 보안 위협의 영역을 정의하고 테스트 계획을 세운다.

2.3 설계 단계

SDL의 구현에서부터 배포에 이르는 동안 수행해야하는 작업 계획을 수립하는 단계로, 보안 설계 검토, 방화벽 정책 준수, 위협 모델링, 위협 모델 품질 보증, 위협모델 검토 및 승인 등을 포함한다.

2.4 구현단계

이 단계에서는 사용자가 소프트웨어를 안전하게 사용할 수 있도록 도와주는 사용자가 사용할 문서들과 도구들을 개발하게 된다. 구현 단계에서 보안 및 프라이버시 문제점을 발견하고 제거하기 위하여 개발 best practice를 수립하고 따르도록 한다.

2.5 검증단계

검증 단계에서는 코드가 이전 단계에서 설정한 보안과 프라이버시를 지켰는지 확인해야 한다. 이것은 보안 및 프라이버시 테스팅과 보안 푸쉬(security push), 문서 리뷰를 통하여 이루어진다. 보안 푸쉬는 팀 전체에 걸쳐 위협 모델 갱신, 코드 리뷰, 테스팅에 초점을 맞춘 작업이다. 공개 배포용 프라이버시 검사 역시 이 단계에서 완수된다.

3. 제안된 소프트웨어 개발 방법론

소프트웨어의 개발주기는 요구사항정의, 분석, 설계, 개발, 테스트, 운영관리 등으로 이루어져 있으나, 본 논문에서는 아래의 표와 같이 세분화된 소프트웨어 개발주기를 A, B, C 세영역으로 크게 구분하여 각 단계에서 발생하는 산출물 및 활동을 소개함으로써 소프트웨어 설계자, 개발자 및 운영자가 어플리케이션 보안 취약성 및 웹 접근성에 좀 더 적극적으로 접근하는데 도움을 주고자 한다.

<표 1> 보안이 강화된 SDLC 산출물 및 활동

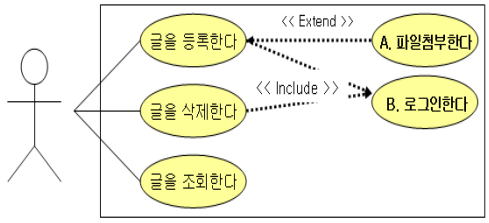
구분	SDLC 단계	산출물 및 활동
A	분석~설계	○ UML를 활용한 위협 모델링 설계
B	개발~테스트	○ 위협분석 체크리스트 작성 - 웹 접근성 및 표준화 가이드라인 준수 ○ 시큐어 코딩 가이드라인 준수 ○ SW 보안 테스트 실시 ○ 웹 접근성 및 표준화 도구를 활용한 진단
C	운영관리	○ 상용 및 응용 SW 패치 방안 수립

3.1 Section A.

미국 CMU 소프트공학연구소 SEI[8]에 의하면, 소프트웨어 보안 사고의 90%이상이 알려진 보안 결함을 악용하여 발생하며, 45개의 e-비즈니스 응용프로그램의 분석결과, 보안사고의 70%가 설계오류로부터 발생한다고 한다. 따라서 외부에서 들어오는 데이터는 반드시 신뢰할 수 없는 것으로 간주하여 어플리케이션의 모든 진입점들을 반드시 나열하여 위협모델링을 이용한 위치별 취약에 대한 대비책을 마련해야 한다.

○ UML을 활용한 위협모델링 설계

- 기능요구사항을 모델링 한 USE CASE 다이어그램을 시퀀스다이어그램으로 변환하여 소프트웨어에서 위협이 발생하는 위치를 파악하고 이 소프트웨어 위협위치에 존재하는 보안취약성 분석하여 취약성을 제거, 방어, 완화할 수 있는 모델을 설계한다.(위협모델링 설계를 위해 OWASP(Open Web Application Security Projec)의 가장 심각한 10가지 어플리케이션 보안 취약점, SANS의 25가지 치명적인 인터넷 보안 취약점 목록 리스트와 CWE(Common Weakness Enumeration)를 취약성 유형별로 정리하여 제공하는 NVD(National Vulnerability Database) 자료 및 국정원8대 취약점을 활용한다. 그림 2와 3은 위협모델링을 구현한 예를 보여주고 있다.



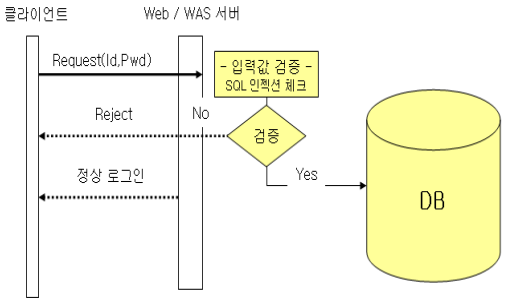
(그림 2) use case모델 작성

위 그림2와 같이 게시판 기능을 구현하는데 기능요사항이 아래와 같이 도출되었다고 가정하자

· 사용자는 로그인후, 글을 등록 및 삭제 가능하다.
· 사용자는 글을 등록시, 파일등록도 가능하다.

파일을 등록하는 구간(A)와 로그인하는 구간(B)에 아래와 같은 취약점이 존재한다.(OWASP TOP10, 국정원8대 취약점)

- (예시) A구간
  · File Upload / Download 취약 등
- (예시) B구간
  · SQL 인젝션 취약 등



(그림 3) 시퀀스 다이어그램을 이용한 위협모델링 작성

그림2의 use case 모델을 이용해서 B구간에서 발생할 수 있는 위협모델링을 작성(그림 3)후, 위험체크리스트를 작성한다.

3.2 Section B.

위협체크리스트는 개발자가 작성된 위협모델링을 이용하여 소프트웨어 보안 취약성과 웹 접근성 및 표준화 가이드라인을 준수하여 개발할 수 있도록 지원하고 업무담당자 및 담당 PM이 체크 완료된 최종리스트를 기반으로 하여 결함주입 보안 테스팅을 하거나 웹 접근성 및 표준화 진단시 활용한다.

○ 위협 체크리스트를 활용방안

- 위협모델링의 분석결과를 바탕으로 소프트웨어 취약성을 위협위치별, 등급별로 구분하여 처리할 수 있는 가이드라인 제시
- 웹 접근성 및 웹표준을 위한 가이드라인 준수[9]

○ 시큐어 코딩 가이드 활용

- 시큐어 코딩은 보안 개념이 보강이 된 방어적 프로그램이라고 볼 수 있듯이 보안 취약점이 발생할 수 있는 코딩의 허점을 사전에 코딩 법칙으로 제거하고자 하는 것이 목적이다. 이러한 시큐어 코딩은 시큐어 코딩 규칙과 그 이외의 이름 규약등 그 이외의 규칙으로 정의가 되는 데, 시큐어 코딩을 실천하면 보안 취약가능성이 매우 줄어드는 것이 알려져 있다. 또한 유지 보수의 비용이 급격히 줄어드는 것으로 나타나고 있다.

○ 소프트웨어 보안테스트 실시

- 위협모델링을 적용한 위협 체크리스트를 기본으로 하여 소스코드 진단, 모의해킹 시도, 시스템에서 운영되고 있는 상용 SW에 대해 침투시나리오를 작성하여 검증한다.

<표 2> 웹 취약점 진단내역

점검항목	진단항목
모의해킹	OWASP Top 10 및 국정원8대 취약점 점검
	어플리케이션 로직, 인증 등 우회기법
소스진단	소스코드레벨에서의 로직, 권한 우회 분석
	자동진단(Tool), 수동진단(eye checking)
보안솔루션 취약성 진단	인증 및 권한우회 가능성 검증
	SW역공학 분석

○ 웹 접근 및 표준화 지침을 활용한 웹사이트 진단

- 「전자정부법」제25조 및 같은 법 시행령 제33조에 따라 행정기관이 전자정부 웹사이트를 신규 구축하거나 개선, 유지보수 및 운영함에 있어서 웹사이트의 호환성 확보[10] 및 접근성 향상을 위해 아래 표와 같은 진단도구를 활용하여 웹사이트를 진단할 수 있다.

<표 3> 웹 접근성(표준화) 항목 및 진단도구

구분	진단항목	진단도구
웹표준수	1. 표준 HTML 준수	W3C HTML Validator
	2. 표준 CSS 문법 준수	W3C CSS Validator
웹접근성	접근성 준수	KADO-WAH
웹호환성	웹호환성 여부	최소3종의 브라우저에서 동등한 레이아웃 및 기능구현 가능여부

3.3 Section C.

상용 및 응용 소프트웨어는 Version Upgrade로 인해 소프트웨어 크기가 점차 증대되고 다양한 기능 구현을 위해 SW 크기가 커짐과 동시에 코드가 복잡해지면서 보

안 취약점도 증가 된다. 이에 상용 및 응용 소프트웨어의 기능 개선시 보안 취약점을 감소시킬 수 있는 절차와 응용 소프트웨어 기능 개선시 웹표준 및 접근성을 준수하여 시스템에 반영시킬 수 있도록 절차를 마련한다[11].

<표 4> 상용SW 반영절차

테스팅 단계	운영반영절차	산출내역
테스트 요청	1. 솔루션테스트(제조사)	운영이관요청서 테스트시나리오
	2. 테스트시나리오 작성	
테스트	3. 단위테스트(제조사)	테스트케이스 결과서 운영이관 대상목록
	4. 통합테스트(운영자)	
운영이관	5. 운영시스템 이관	
운영최종검증	6. 통합검증	솔루션 변경내역서
	7. 응용SW와 영향성파악	
	8. 승인여부	

<표 5> 응용SW 반영절차

테스팅 단계	운영반영절차	산출내역
테스트 요청	1. 요청기능개발	테스트 목록 테스트시나리오
	2. 테스트시나리오 작성	
웹표준 및 보안테스트	3. 보안여부 준수	테스트케이스 결과서 운영이관 대상목록
	4. 웹표준여부 준수	
테스트	5. 제3자테스트	
	6. 통합테스트	
	7. 사용자승인테스트	
운영이관	8. 운영시스템 이관	
운영최종검증	9. 제3자검증	프로그램명세서 화면설계서
	10. 통합검증 및 승인여부	

4. 결론

기존에는 웹기반의 환경의 보안강화를 위해서는 방화벽, 침입탐지시스템 등과 같은 네트워크 시스템 측면 혹은 보안모듈을 강화하였지만 이러한 노력은 근본적인 해결책을 제시하지 못하였으며 지속적인 갱신을 필요로 한다는 한계를 가지고 있다. 또한 최근에는 웹 어플리케이션에 대한 지식과 다양한 도구를 사용하여 고난도 공격을 시도하기 때문에 방화벽, IDS, IPS와 같은 기존 보안 솔루션으로는 적절히 대응을 하지 못하고 있다.

따라서 본 논문에서는 어플리케이션에 대한 근본적인 보안 취약점의 문제와 웹 접근성 문제 그리고 장애인 차별금지등을 개선하기 위하여 소프트웨어 개발 생명주기를 크게 A, B, C 영역으로 구분하여 각 영역에서 보안 취약성을 최소화 할 수 있는 방안을 제시하였고, 장애인 및 고령자와 같은 정보취약계층이 웹 사이트를 이용하는데 불편함을 줄이기 위한 절차도 함께 알아보았다. 향후에는 소프트웨어 개발 단계 각 영역에서 보안강화를 위해 다루어져야 할 심화된 연구가 필요하며 스마트 폰을 대상으로 서비스할 때 필요한 웹 응용프로그램에 대한 연구도 필요하다고 판단이 된다.

[참고자료]

[1] 한근희, “전자정부 정보보호관리체계(G-ISMS) 적용정책”, 정보보호학회지 논문, 2009.
[2] SANS: http://www.sans.org/top-cyber-security-risks
[3] 보안뉴스 “ [특별기고-7] 사용자들에게 단순 명료하고 풍부한 보안제공해야 ”,2010.
[4] Goertzel, Karen, Theodore Winograd, et al. Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance, October 2008.
[5] MS-SDL: http://www.microsoft.com/security/sdl/
[6] http://msdn.microsoft.com/en-us/magazine/cc163705.aspx
[7] 이송희, 최진영, 강인혜, 서동수, 안재영, 한근희, 시큐어 소프트웨어 개발을 위한 소프트웨어 개발생명주기 동향, 정보과학회지, 2010.
[8] CMU Software Engineering Institute, The Team Software Process and Security, http://www.sei.cmu.edu/tsp/tsp-security.html
[9] 웹접근성연구소, “http://www.wah.or.kr”
[10] 행정안전부, 전자정부 웹호환성 준수지침,2009
[11] 행정안전부,정부민원포털(G4C) 유지보수 착수보고회,2010.