6 시그마 방법론을 이용한 웹사이트 응용프로그램의 보안 수준 평가 및 개선

황우* 이희조
고려대학교 컴퓨터정보통신대학원
e-mail : wooya@hotmail.com*, heejo@korea.ac.kr

A Study on Web-Site Application Security Level Measurement and Improvement of Using Six Sigma method.

Woo Hwang* Hee Jo Lee
Dpt. Software Engineering, Graduate of School Computer and Information Technology,
College of Information and Communications, Korea University.

요 약

보안수준 평가 및 개선에 있어서 온라인 웹사이트 응용프로그램의 경우 측정 기준 설정과 보안 수준 표시에 많은 어려운 점이 있다. 이에 모토롤라부터 도입한 경영혁신 도구로서의 6 시그마 수준 표현 및 개선 기법을 웹사이트 보안수준 평가에 적용할 수 있도록, WASC(Web Application Security Consortium)에서 제공하고 있는 웹 애플리케이션 보안 평가 체크리스트를 사용하여 Bottom-Up 방식으로 웹 사이트에 대한 실제 침해 시도의 결과를 측정, 이를 보안수준 측정 및 개선에 활용할 수 있는 방안을 제시한다.

1. 서론

웹사이트 어플리케이션의 보안품질의 측정과 개선은 서비스의 안정적 운영과 경영적 측면에서도 주된 관심사가 되고 있으나, 그 수준의 표시 및 개선 정도는 표현이 어려운 점이 있다. 이에 경영적 품질개선 도구인 6 시그마를 웹 서비스 보안 측정 전반에 적용함으로서 보안수준을 시그마 수준으로 명시적으로 정의하고, WASC 에 기반한 여러 공격 유형을 실제 운영중인 사이트에 적용하여 발생하는 문제점에 대한 근본 원인 분석 및 제거를 통한 보안 수준 향상을 위한 방법론을 제시한다. 본 논문의 2 장에서 국내외의 보안수준 표시와 연구현황과 6 시그마에 대한 설명, 3 장에서는 6 시그마를 통한 보안수준 표시와 개선을 위한 평가와 근본 원인 분석, 4 장에서는 대상 응용프로그램의 최근 개선된 내용 확인을 통한 개선내용 분석, 5 장 마무리에서는 향후 연구방향에 대해 기술한다.

2. 관련연구

2.1 웹사이트 보안 품질 평가에 관한 연구

웹사이트 어플리케이션 보안 수준 측정에 관련된 방법은 정확한 표준 제정은 이뤄지지 않았으며, OWSAP Top 10, 국정원 8 대 홈페이지 취약점 차단이 기본적으로 통용되고, WASC(Web Application Security Consortium)의 웹 애플리케이션 방화벽 평가 기준(WAFEC ; Web Application Firewall Evaluation Criteria Version 1.0 ; 2006 년 1 월 16 일 버전 1.0, 9 가지 카테고리로 분류)과 NSS 그룹의 웹 애플리케이션 방화벽 테스팅 절차(v 1.0)에 하나의 변형을 포함한 31 가지 공격 인식 테스트 등을 참고하고 있다.[1] 또한 웹사이트 보안 평가에 있어서 많은 논문이 보안 평가 기준 설정 및 측정을 통한 정량적 방법이 아닌 AHP 등을 이용한 정성적인 접근을 통한 방법[2] 을 사용하고 있다. 본 논문은 WASC 의 보안 침해 목록을 실제 운영중인 테스트 사이트에 적용시켜 얻은 결과를 보안 평가 기준자료로 이용하고자 하는데, 보통 소프트웨어 보안 사고의 90%이상이 알려진 보안결함을 악용하여 발생한다고 할 때[3], 실제 악용사례를 정리한 WASC 리스트에 대한 사전 검사와 원인분석을 통한 개선은 중요한 의미를 지니고 있다.

2.2 6 시그마의 개념과 특징

시그마(σ)는 알파벳의 18 번째 글자의 이름으로 통계학의 표준편차 기호로 사용되며, 6 시그마는 1 백만건 중 3~4 개의 불량이 있는 수준을 이야기한다. 불량은 정해진 범위 밖에 있는 모든 요소로서 이의 데이터를 수집하고 통계를 분석함으로써 오류의 원인을 찾고 제거 하는 것이다. 따라서 6 시그마를 통한 품질향상은 모든 제품과 프로세스에서 결점이 없는 품질성과를 달성하기 위해 프로세스를 정량적으로 평가하고 문제점과 원인을 찾아 통계적 사고로 해결 할 수 있는 방법론을 제공한다. 6 시그마의 문제 해결과정은 DMAIC 방식과 DFSS(Design For six sigma) 방식으로 나눌 수 있는데, 전자의 경우 기존 프로세스의 개선에 주로 사용되며 후자의 경우 새로운 제품이나 서비스의 초기 설계, 재설계 단계에서 사용된다. DMAIC 는

데이터 중심의 체계적인 개선 전략으로 본 논문에서는 DMAIC 방식을 연구에 사용하였다.

절차	진행 내용
정의 (Define)	요구사항을 파악하고 목적을 정함
측정 (Measure)	측정지표의 현수준(Y) 파악하고 잠재 인자(X) 발굴
분석 (Analysis)	파악한 잠재인자(X)의 분석을 통해 핵심요인 및 영향 정도를 확인
개선 (Improve)	개선안을 도출하고 최적 안을 평가,선정,실행
관리 (Control)	개선결과의 점검 계획을 수립하고 실행

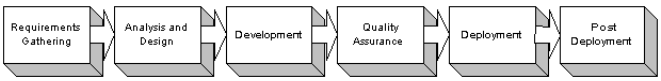
<표 1> DMAIC 기법의 진행

3. DMAIC 을 이용한 보안 수준 도출 및 평가

3.1 정의

온라인 서비스를 운영하는 시스템 부서는 보안이슈 발생시 운영중인 시스템의 보안 수준 파악 및 이의 개선이 필요하다. 이를 위해선 보안에 영향을 주는 주요 원인 파악 및 현수준 표시 기준을 정립해야 하는 절차를 따라야 하는데 이 절차에 6 시그마 기법을 적용하여 개선하고자 한다.

개발 과정 진행 프로세스는 아래 그림과 같은 표준적인 선형 라이프사이클로 설정했다.



(그림 1) 소프트웨어 개발 라이프사이클[4]

3.2 측정.

3.2.1 프로젝트 Y 의 선정과 CTQ(핵심요구사항) 도출

소프트웨어 보안 품질 개선으로 설정하였다. CTQ 는 Critical To Quality 로 내부 외부 고객과의 인터뷰를 통하여 VOC 를 정리하고, 이를 통해 CCR 즉 Critical Customer Requirement 를 도출한 다음 다음의 표와 같이 CTQ 를 선정한다.

→		CTQ	←	
VOB	CBR		CCR	VOC
매출증대 이익증가	보안성확보 안정적서비스	보안 품질향상	개인 정보유지	기본적 품질

<표 2> CTQ 도출

CTQ 를 수치화 하기 위해 아래와 같이 보안 침해 시도를 통하여 발생된 문제점으로 설정하였다.

$$CTQ(\text{보안관련버그발생비율}) = \frac{\text{발생된문제점}}{\text{전체시도}} \times 100$$

3.2.2 테스트 및 측정

많은 커뮤니티 사이트에서 사용하는 제로보드를 이용한 테스트 사이트를 구축했다.
보안수준 측정을 위한 체크리스트는 WASC 의 보안위협체크리스트[5] (총 24 개 분류, 세부 12000 개 정도의 체크리스트)를 여러 가지로 변형하여 watchfire 사의 appscan 을 이용하여 전체를 체크 할 수 있다. 우선 제로보드에 접속 권한을 주지 않은 상태로(최소)와 주었을 때와 손님권한을 주었을 때를 제로보드 버전에 따라 각각 1 회씩 총 4 번 실시하였다.

구분	측정도구	제로보드 버전	제로보드 권한
Test1	Appscan 7.5 840 update	41p17	무권한
Test2	Appscan 7.5 840 update	41p17	관리자
Test3	Appscan 7.5 840 update	41p18	무권한
Test4	Appscan 7.5 840 update	41p18	관리자

<표 3> 테스트 계획

3.2.3 현 수준 측정 및 시그마 표현

Test-1,2 를 진행하여 다음의 결함 데이터를 얻었다.

위협 / 권한	상	중	하	문제점 총합	총시도
무권한	24	17	5	46	18126
관리자	27	17	8	52	21345
결함합계	51	34	13	98	39471

<표 4> 보안 문제 발생 결과

이를 6 시그마 수준으로 표시하기 위해 데이터를 확인해본 결과 이산형이며 단일시점 데이터 임으로 DPMO(Defects Per Million Opportunities) 방식으로 측정을 진행한다. DPMO 를 구하는 방법은 다음과 같다.

$$DPMO = \frac{\text{발생된문제점}}{\text{시도횟수} \times 1\text{번시도당 기회횟수}} \times 1000000$$

이를 기초로 테스트 1,2 번의 DPMO 를 구하면 다음과 같다.

$$DPMO = \frac{98}{39471 \times 1} \times 1000000 = 2{,}483$$

결함율(%) = 0.25

DPO(전체기회당불량발생수 = 0.00248284

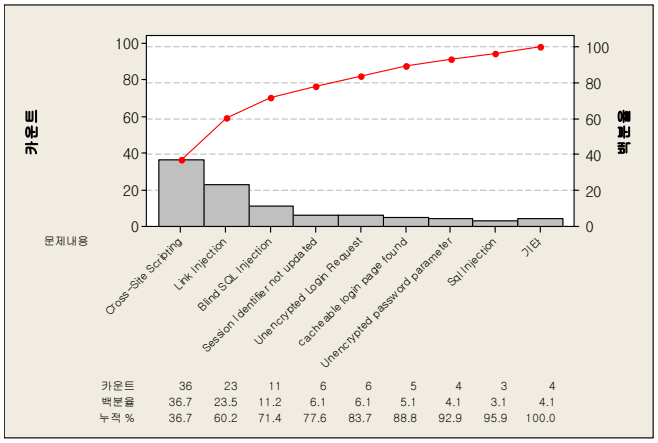
수율 = $1 - DPO$ = 0.9975

시그마수준 = 4.31

따라서 제로보드 ver 41p17 의 보안시그마 수준은 4.31 σ 정도로 도출된다.

3.3 분석 및 개선안

Test1,2 에서 발생된 내용을 Pareto 분석을 이용하여 문제점을 확인해본 결과 4 가지 형태의 보안 위험이 전체의 83%를 차지하고 있다.



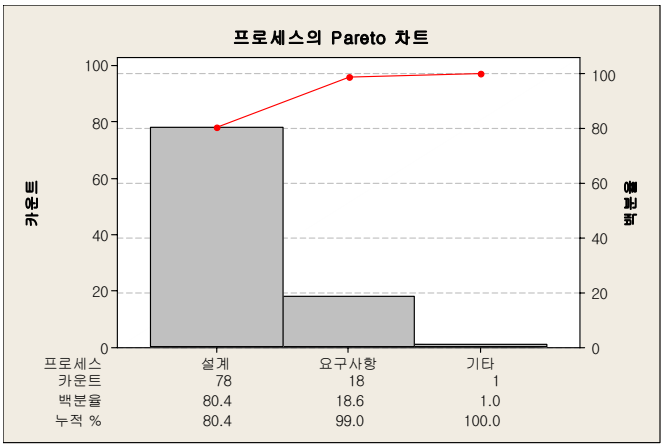
(그림 2) 보안 위협 Pareto 분석

위 보안문제가 주로 고려 되어야 할 개발개발 프로세스와 연결한다면 다음[6] 과 같다.

보안침해 내용	관리자	무권한	총합	프로세스
Cross-Site Scripting	19	17	36	설계
Link Injection	12	11	23	설계
Blind SQL Injection	6	5	11	설계
Unencrypted Login Request	3	3	6	요구사항
Session Identifier not updated	3	3	6	요구사항
Cacheable login page found	2	3	5	요구사항
Unencrypted password parameter	2	2	4	설계
SQL Injection	2	1	3	설계
Login error messages credential enumeration	0	1	1	설계
Application error	1	0	1	코딩
Inadequate account lockout	1	0	1	요구사항

<표 5> 보안침해 내용과 관련된 프로세스

이를 다시 Pareto 그림으로 표현하면 다음과 같다. 문제가 발생하는 프로세스 기준을 확인해본 결과 (그림 1)의 표준개발프로세스 기준 요구사항 및 설계에 관련된 문제가 전체의 99%를 차지하고 있는 것을 알 수 있다.



(그림 3) 문제점을 일으키는 프로세스

3.4 개선안

보안 시그마 수준을 높이기 위해서는 "감지수준을 높이거나", "보안 문제점 유입을 차단"[7] 해야 한다. (그림 3)에서 보듯이 대부분의 보안 문제점 발생의 원인이 개발과정 중 설계/요구사항에 집중되는 것을 알 수 있다. 따라서 해당 프로세스 문제점에 대한 개선안을 설계 단계에서 "침해 모델링"과 요구사항 분석 부분에서 "오용 및 악용 사례" 분석 부분[8] 을 추가했다.

4. 검증

<표 3>의 Test3,4 에서 V8(41p18)로 개선된 제로보드의 보안 fault 측정하면 다음의 결과를 얻을 수 있다.

위협 / 권한	상	중	하	문제점 총합	총시도
무권한	5	3	5	15	18126
관리자	5	4	5	16	18126
총합	10	7	10	31	36252

<표 6> 개선된 버전의 보안 fault 측정

이는 불량률 0.09% 수율 99.91, 시그마 수준 4.64 σ 로 시그마 수준이 개선된 것을 볼 수 있다. 개선된 부분을 문제 프로세스로 분류해 본다면 대부분의 보안 문제점이 요구사항 분석과 설계 프로세스에 기인한 것을 알 수 있다.

개선된 fault 내용	프로세스
파일 업로드 시 확장자 무시	요구사항
SQL injection	설계
Cross-site scripting	설계

<표 7> 개선된 fault 와 해당 프로세스

하지만 제로보드 v8 에서는 <표 6>과 같이 여전히 많은 취약점이 발견되고 있다. 따라서 41pv17 과 41pv18 사용자의 경우에는 관련된 취약점을 해결 할 수 있도록 방안을 강구 해야 한다.

5. 결론

6 시그마는 문제점 자체의 개선보다는 문제점이 발생하게 되는 근본 원인의 파악 및 개선에 주 목적이 있다. 보안 문제의 경우 이러한 접근방법이 다른 어떤 경우보다 더 중요하다고 할 수 있으므로 6 시그마를 이용한 보안 문제 해결은 효과적인 접근 방법이라고 할 수 있다. 본 논문에서는 WASC 체크리스트와 제로보드를 이용하여 제로보드의 보안 수준을 시그마 수준으로 가시적으로 표시 했으며, 이와 동일한 방식으로 제로보드 이외의 서비스에도 확장 가능하다. 단, 평가 진행 단계에서 보안 fault 의 위협수준에 대한 가중치 등은 고려되어 있지 않으므로 가중치가 반영될 수 있도록 좀더 연구할 필요가 있다.

참고문헌

[1] http://www.eweekkorea.com/02_contents/print.asp?num=16538 2007.02.13 웹에플리케이션 보안

[2] 전자상거래를 위한 웹사이트 보안 평가에 관한 연구 김현우, 이근수,김세헌 2005 년 한국과학기술원 산업공학과

[3] 정보시스템 감리품질향상을 위한 보안감리평가에의 정량화 모델 적용 연구 page 9. 한국정보보호진흥원. 2003
[4] “Building Web Application Security into Your Development Process” Figure 1. Kevin Heineman. 2005.
[5] Web Application Security Consortium: Threat Classification version 1.0
[6] “Building Web Application Security into Your Development Process” Figure 3. Kevin Heineman. 2005.
[7] “Six Sigma Approach in Software Quality Improvement” Cvetan Redzic, Jongmoon Baik. 2006 IEEE
[8] “A Portal for Software Security” Figure A. Nancy R. Mead, Gary McCraw . 2005,7