KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.8 pp.189~196 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2016.5.8.189

Cost-Effective and Active Security Verification Framework for Web Application Vulnerabilities

KyungHyun Han[†] · Trong-Kha Nguyen[††] · Hun Joe[†††] · Seong Oun Hwang[††††] · Chaeho Lim[†††††]

ABSTRACT

Many companies have struggled to manage Web vulnerabilities and security incidents have also frequently happened. The current inspection methods are mainly based on the OWASP vulnerabilities. In practice, however, it is very difficult to cope with frequent changes of Web applications. In this paper, we first investigate the existing quantification of Web application vulnerabilities and verification process. Then we propose an improved inspection framework which is focused on removing essential and realistic vulnerabilities and active verification process.

Keywords : Cost-Effective, Web Security, Vulnerabilities, Inspection

웹 애플리케이션 취약점 분석을 위한 비용 효과적인 능동 보안 검수 프레임워크

한 경 현[†] · Trong-Kha Nguyen[††] · 조 훈[†††] · 황 성 운[††††] · 임 채 호[†††††]

요 약

많은 기업들은 웹 취약점 관리에 어려움을 겪고 있으며, 보안 사고도 자주 발생하고 있다. 현재는 OWASP 취약점에 기반을 둔 점검도구로 검수를 진행한다. 하지만 현재의 보안 검수는 웹 애플리케이션의 잦은 변화에 현실적으로 대응하기 어려운 실정이다. 본 연구에서는 먼저 기존 취약점 수치화 연구와 검수 프로세스를 검토하였다. 이에 기반을 두어 현실적이며 필수적인 주요 취약점 제거와 능동적인 검수 프로세스에 바탕을 둔 개선된 검수 프레임워크를 제안한다.

키워드 : 비용 효과, 웹 보안, 취약점, 검수

1. 서 론

최근 개인정보 유출 사고를 일으킨 인기 온라인 커뮤니티 '뽐뿌'에 대해 1억 1700만원의 과징금이 부과되었다[1]. 'SQL 인젝션'을 통해 회원 약 196만 명의 개인정보가 탈취되었기 때문이다[2]. SQL 인젝션이란 DB에 대한 질의(SQL 구문)를 조작해 정상적인 자료 외에 해커가 원하는 자료까지 DB로부터 유출해가는 사이버 공격 기법이다[3]. 공격당한 웹페이지는 당초 숫자만을 질의할 수 있도록 돼 있었는데 숫자 외에 ID, 생년월일, 이메일 주소 같은 개인정보를 질의하는 SQL 구문을 삽입해 실행할 수 있는 취약점을 악용 당한 것이다.

인터넷 보안은 무결성(Integrity) 점검이 필수이다. 이것은 서비스나 정보의 수정이 필요하다면 정상적인 절차를 거쳐야 함을 말한다. 특히 웹 애플리케이션 보안 검수는 웹 코드 수정이 이루어질 때마다 반드시 필요한 보안 검수이다. 하지만 '뽐뿌'를 비롯하여 사실상 많은 기업들은 잦은 애플리케이션의 수정으로 인해 매번 보안 검수를 하기가 현실적으로 어려운 실정이다. 또한 협력업체를 통한 공격이 있을 수 있으므로 대기업은 많은 협력업체에 대한 웹 취약점도 분석하여야 한다. 하지만 기존의 SDLC (Secure Development Life Cycle) 방식은 즉각적인 검수조사, 즉각적인 재 코딩, 재 검수 등이 이루어지지 않으므로 서버의 수나 잦은 수정

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(IITP-2015-R2213-15-0002, 전자영수증 처리, 분석시스템). 또한 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 해외ICT전문인력활용촉진사업의 연구 결과로 수행되었음(NIPA-2014-H0905-14-1004).

† 준 회 원 : 홍익대학교 전자전산공학과 석사과정
†† 비 회 원 : 홍익대학교 전자전산공학과 석사과정
††† 비 회 원 : 한국과학기술원 전산학과 박사과정
†††† 비 회 원 : 홍익대학교 컴퓨터정보통신공학과 교수
††††† 종신회원 : 한국과학기술원 전산학과 교수

Manuscript Received : March 30, 2016
First Revision : June 17, 2016
Accepted : July 25, 2016

* Corresponding Author : Seong Oun Hwang(sohwang@hongik.ac.kr)

에 비례하여 엄청난 시간과 노력 등의 비용이 요구되고, 새로운 취약점이 발견되어도 즉각적으로 웹 어플리케이션의 보안 업데이트를 하기 어렵다.

본 논문에서는 1) SDLC 기반 보안 검수방법, 2) OWASP 기반 취약점 분석, 3) 이미 발표된 취약점의 권한 획득 정도에 따른 웹 애플리케이션 취약점 수치화 프레임워크 등을 검토하여 중요하다고 판단된 SQL인젝션이나 XSS 등 5가지 취약점 점검을 이용하여 보안 검수를 수행할 수 있는 개선된 형태의 능동 웹 취약점 검수 프레임워크를 제안한다. 이 검수 방식은 기존의 SDLC 절차와 달리 능동적인 방식을 사용하기 때문에, 보안부서가 다른 부서의 요청없이 자율적으로 보안 검수를 시작할 수 있다. 따라서 보안부서가 주기적으로 새로 발견된 취약점에 대한 검사가 포함된 보안 검수를 시작하는 등의 적극적인 운영이 가능하다.

2. 관련 연구

2.1 웹 보안 사고와 취약점 분포

Table 1은 2005년 하반기에 Gartner가 전체 1,000여개 가량의 Web site를 상용 웹 애플리케이션 스캐너(Web application scanner)를 이용하여 점검한 결과이다. 이에 따르면 98% 가량의 웹 서버들이 웹 코드로 인해 Table 1과 같은 다양한 보안 취약점을 가지고 있음을 알 수 있다[4, 5].

Table 1. Web vulnerability distribution (2005, Gartner)

보안 취약점	비중(%)
세션 가로채기, 개인 정보 유출	32
제어 정보 접근 가능	21
개인정보 노출	27
온라인 쇼핑 정보 누출	11
정보수정 가능	7
웹 사이트 삭제	2

Table 2. Web attack vulnerability distribution[8]

순번	내용	비율(%)
1	DDOS	32
2	SQL 인젝션	21
3	XSS	9
4	전수공격(Brute-force)	4
5	XSS 요청 위조	4
7	알려진 취약점 공격	4
8	알려지지 않은 공격	10
9	기타 공격 유형	16
	합계	100

온라인 커뮤니티 '뽐뿌' 사고와 2011년 발생한 현대캐피탈 사건에서 볼 수 있듯이 개인정보 유출 등의 사고는 보안 취약점이 악용 당해서 발생한다[6, 7]. 이것은 금융을 비롯한 많은 기업이 인터넷 서비스를 제공한다는 점을 고려한다면 큰 문제이다. Table 2는 2011년 상반기에 Trustwave가 발표한 주요 웹 취약점의 악용 빈도를 나타낸다. 이 중에서 SQL 인젝션 등 웹 코드로 인한 웹 취약점을 음영으로 표시해두었다.

2.2 SDLC 보안 검수 프레임워크

SDLC는 시스템 개발과정에서 개시, 분석, 설계, 구현, 유지 및 폐기단계에 필요한 보안 검수가 추가된 프로세스를 의미한다[9]. 이를 웹 취약점 제거를 통해 안전한 웹 상태를 유지하는 State Transition Diagram으로 본다면 다음 Fig. 1과 같다. Code Verification은 보안성 검수 프로세스가 필요함을 말한다. 위험한 상태로의 전이는 사업부 개발자의 신규 웹 코딩 혹은 신규 취약점 발견으로 발생할 수 있다. 따라서 서비스를 개시하려면 주기적으로 검수 프로세스를 필히 실시해야 한다.

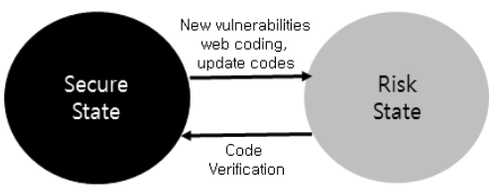

Fig. 1. Web security state transition diagram

이와 같이 웹 서비스를 제공하는 기업에게 보안 검수는 필수적인 절차이지만, 많은 기업들은 보안 검수에 소모되는 비용에 민감하기 때문에 보안 검수를 하지 않으려고 한다. 아직까지 사업자의 입장에서는 보안 검수는 수익이 없고, 서비스 제공이 지연되며, 비용만 소모되는 작업이라는 인식이 있기 때문이다.

또한 취약점으로 인한 보안 위험 뿐만아니라 정부의 법적 요구사항(Compliance)을 정하는 것도 민감한 문제이다. 정부는 보안 사고를 방지하기 위해 각 기업이 스스로 보안 검수를 하도록 정책을 만들어야 한다. 하지만 기업은 위에서 설명한 바와 같이 시간과 비용 문제로 인해 정책 결정에 반발 혹은 정책 불이행을 하게 된다. 따라서 정부가 만든 정책을 기업이 부담없이 이행하기 위해서는 보안 검수의 시간과 비용을 줄이는 방법이 필요하다.

2.3 검수가 필요한 주요 취약점

보안 검수의 비용을 줄이기 위한 방법에는 먼저 검수 대상을 줄이는 방법이 있다. 이를 위해 우리는 기존의 취약점 수치화 프레임워크에 대한 연구[8]의 결과를 아래에서 자세히 설명한다.

[8]에서는 먼저 해외에서의 여러 수치화 방법들을 분석하였다. 해외 수치화 방법의 분석 결과를 Table 3에 요약하였다. [8]에 따르면, 해외 수치화 방법 연구는 하나의 취약점이나 하나의 서비스에 대하여 점수를 산출하지만, 한 기업을 기준으로 여러 서비스나 여러 취약점이 발견될 때, 우선적

Table 3. Summary of Web vulnerability score study in overseas

방법	내용	비고
CVSS [10, 11]	취약점의 본질적인 특성, 시간에 따른 특성, 환경에 따른 특성을 고려하여 계산	하나의 취약점만 측정 애플리케이션 단위로 측정 불가 기업 단위로 우선순위 제공 불가
CWSS [10]	취약점의 특성, 공격하기 위해 통과해야 하는 장벽, 환경 및 악용 가능성 등을 고려하여 계산	조합에 따른 이중 계산 문제 발생 애플리케이션 단위로 측정 미흡 기업 단위로 우선순위 제공 불가
CWRAF [11]	다양한 시나리오를 가정하여 각 시나리오 별로 업무상 위험과 업무에 미치는 영향을 고려하여 계산	자동 점검 도구에서 사용 불가 설정할 시나리오가 많음 애플리케이션 단위로 측정 가능

* CVSS (Common Vulnerability Scoring System)
* CWSS (Common Weakness Scoring System)
* CWRAF (Common Weakness Risk Analysis Framework)

Table 4. Definition of five major Web vulnerabilities[8]

순번	취약점	피해 가능성
1	SQL 인젝션	데이터베이스의 레코드에 대한 변조, 누출, 삭제 서버 권한 획득, 내부망 침입, 악성 코드 유포지 활용
2	블라인드 SQL인젝션	SQL 인젝션과 동일(시간이 다소 오래 걸림)
3	XSS	세션 하이재킹, 피싱, 관리자 및 사용자 권한 유출, 리다이렉션, 사용자 정보 유출
4	예외적 오류	실행 경로 유출, 구성 패키지 정보 노출, SQL 인젝션, HTTP 500 에러 공격
5	의심스러운 오류	취약점 여부 미확인 또는 오류 가능성이 있으며, 검토가 필요한 오류

으로 보완해야 할 취약점을 알려주지 않는다고 지적하였다.

이 문제를 해결하기 위해 [8]에서는 여러 취약점을 권한 획득 정도를 기준으로 취약점을 평가하는 방법을 제시하였다. 제시하는 권한 획득 여부에 따라 3가지 등급으로 나누었다. 이는 각각 1) 정보시스템 정보획득, 2) 개인정보 획득, 3) 정보시스템 통제이다.

또한 [8]에서는 기존 수치화 프레임워크와 위에서 제시한 취약점 평가 방법을 이용하는 새로운 수치화 프레임워크를 개발하였다. 그리고 개발된 프레임워크를 사용하여 취약점을 분석하고, Table 4와 같이 필수적으로 점검이 요구되는 5가지 취약점을 점검하면 안전하다는 결과를 보였다.

2.4 기존의 웹 취약점 검수 프로세스

보안 검수의 비용을 줄이기 위한 다른 방법은 검수 프로세스를 개선하는 방법이 있다. 다음 Fig. 2는 기업들이 일반적으로 수행하는 SDLC 검수 프로세스를 보이고 있다.

SDLC 검수에 의한 프로세스는 ① 사업부서의 신규 사업으로 인한 웹 어플리케이션 개발(혹은 수정)을 요구하는 단계, ② 개발부서에서 개발된 웹 코드에 대한 보안 검수를 요청하는 단계, ③ 보안부서에서 취약점을 검사하고 그 결과를 통보하는 단계, ④ 개발부서에서 수정이 필요한 웹 코드를 수정하고 재검수를 요청하는 단계, ⑤ 보안부서에서 다시 취약점을 검사하고 그 결과를 통보하는 단계 ⑥ 개발부서에서 개발의 완료를 통보하는 단계, ⑦ 사업부서에서 취약점이 없는 웹 코드를 게재하는 단계의 과정을 가진다. 이 방법은 매우 일반적인 검수방식이지만, 단계가 많아 프로세스의 시작부터 웹 코드를 웹 서버에 적용하는 데까지의 시간이 많이 걸리고 취약점이 새로 발견되었을 때, 운영 중인 웹 서버를 검수할 수 없는 단점이 있다. 또한, 인터넷 비즈니스에 종속적인 기업은 신규 웹 애플리케이션의 출현 뿐 아니라, 기존 애플리케이션의 수정도 자주 있기 때문에, 보안 검수에 소요되는 시간과 비용이 큰 문제로서 나타난다.

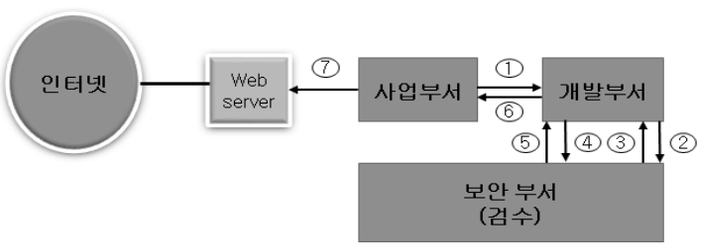

Fig. 2. Existing organizational structure of the Web vulnerability inspection flow

3. 비용 효율적인 능동 웹 취약점 검수 프레임워크 제안

3.1 검수 대상 선정

취약점 수치화 프레임워크에 대한 연구[8]에서는 1) SQL 인젝션, 2) Blind SQL인젝션, 3) 크로스 사이트 스크립팅(XSS), 4) 예외적 오류, 5) 의심스러운 오류 등 5가지 취약점이 가장 중요한 취약점임을 보였다. 특히 SQL 인젝션은 개인정보가 저장된 DB와 관련된 심각한 취약점이다. 분석대상이 되는 웹 취약점을 다음의 3가지로 분류하였다.

1) A Zone: [8]에서 제시한 5가지 취약점
2) B Zone: OWASP에서 발표한 주요 취약점
3) C Zone: 모든 취약점

각 집합을 그림으로 표현하면 다음 Fig. 3과 같다.

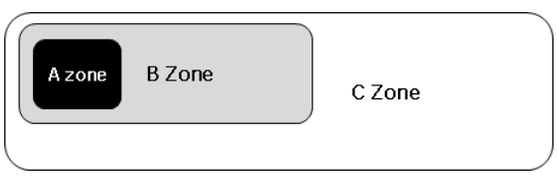

Fig. 3. Web vulnerability sets

다음 Table 5에서 각각의 취약점 집합(Zone)의 특징을 분석하였다. C Zone 취약점을 점검하기 위해서는 Syntax 오류를 포함한 매우 많은 점검을 해야 하고, 점검 결과의 양이 많다. 그리고 소스 코드를 분석해야 하기 때문에 운영중인 서버를 점검할 수 없다. B Zone 취약점을 점검하기 위해서는 공격적인 점검 방법을 사용하기 때문에 운영 중인 서버를 점검하면 피해가 발생할 수 있다. 그리고 이 방법 역시 점검 결과의 양이 많다. A Zone 취약점 점검 방법은 일부 취약점만 점검하지만 중요한 취약점들을 점검한다. 점검 대상이 줄어들었기 때문에 점검 결과의 양도 적다. 특히 A Zone 취약점 점검은 운영 중인 서버도 점검이 가능한 장점이 있다. 이 특징을 고려하여 본 논문에서는 기존에 점검하던 B Zone이나 C Zone 취약점 대신 A Zone 검수(주요 5가지 취약점을 점검)를 할 것을 제안한다.

3.2 제안하는 웹 취약점 검수 프로세스

Fig. 4는 본 논문에서 제안하는 비용 효과적인 능동 웹 취약점 검수 프로세스를 보여주고 있다.

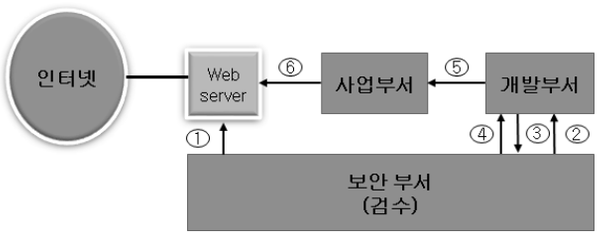

Fig. 4. Proposed active organizational structure of the Web vulnerability inspection flow

제안하는 검수 프로세스의 첫 번째 단계는 보안부서의 웹 서버 취약점 점검이다. 기존의 검수 프로세스는 사업부서에서 개발부서로, 개발부서에서 보안부서로 검수를 요청하기 때문에 각 부서간의 의사 전달 시간을 필요로 하지만 제안하는 검수 프로세스는 주기에 따라 사업부서나 개발부서의 요청 없이 보안부서에서 능동적으로 검수를 시작할 수 있기 때문에 각 부서간의 의사 전달 시간을 필요로 하지 않는다. 제안하는 검수 프로세스에서 보안부서는 기존의 개발된 웹 코드를 검수하는 것이 아니라, 운영 중인 웹 코드를 검수한다. 특히 A Zone 취약점을 점검함으로써 운영 중인 서버를 직접 점검할 수 있게 되었기 때문에 개발부서에서 취약점 점검용 웹 서버를 구축하는 데 소모하는 시간도 없앨 수 있다.

두 번째 단계는 개발부서에 결과를 통보하는 것이다. 개발부서의 요청에 의한 결과가 아닌 주기적인 점검 결과이기 때문에 리포트의 형식을 지정하여 전자우편을 통해 단순하고 신속하게 통보할 수 있게 되어 리포트 생성 시간도 줄일 수 있다.

세 번째와 네 번째 단계는 보안 업데이트가 필요한 경우에 대한 것이다. 웹 코드에서 취약점이 발견될 경우 개발부서에서 이를 수정하고, 보안부서에서 수정된 웹 코드에 대한 재검수를 해야 한다. 따라서 이 단계는 필요에 따라 생략되거나 반복될 수 있다. 이 단계는 기존과 차이가 없기 때문에 개발부서는 기존처럼 신규 개발된 웹 코드에 대한 검수를 요청할 수도 있다. 그리고 A Zone 취약점을 점검하게 되면 개발부서는 보안부서로부터 소량의 중요한 취약점

Table 5. Each zone analysis

Zone	내용	분석 방법	비고
C	웹 코딩 신택스(Syntax) 오류 포함 매우 큰 분량의 점검 필요	Source Code Audit	매우 많은 분량의 출력 서버 운영 중 점검 불가
B	OWASP Top 10 취약점으로 공격적인 점검 필요	공격 Scanner	많은 분량의 출력 서버 운영 중 점검 불가
A	Top 5 취약점으로 위험이 매우 크지만 소량의 점검 필요	Crawling 점검 가능	소량의 출력 서버 운영 중 점검 가능

Table 6. Procedure descriptions of the proposed active Web vulnerability inspection flow

절차	내용	비고
①	보안부서의 웹 서버 취약점 점검	개발 부서의 요구 없이 점검 시작 가능 새로운 취약점에 대한 즉각적인 보안 검수가 가능
②	개발부서에 결과 통보	전자우편으로 단순 신속 전달
③	개발부서의 웹 코드 수정	소량의 중요 취약점에 대한 신속한 보안 업데이트 가능 개발부서의 지속적인 보안 코딩 학습 가능
④	보안부서의 재검수 및 결과 통보	
⑤	개발부서의 완료 통보	기존과 동일
⑥	사업부서의 보안 운영	

에 대한 결과만 통보 받기 때문에, 신속한 웹 코드 수정과 보안 코딩 학습이 가능해진다. 이것은 신규 코딩으로 인한 취약점 발생을 억제할 수 있다는 의미이다.

다섯 번째와 여섯 번째 단계는 보안 업데이트가 완료된 후에 대한 것이다. 사업부서는 검수된 웹 코드를 받아서 운영 중인 웹 서버에 게재하는 단계이며 이 부분은 기존 검수 프로세스와 차이가 없다.

앞의 내용을 표로 정리하면 위의 Table 6과 같다.

4. 검증 및 분석

4.1 대상 취약점 변경으로 인한 효과 검증 및 분석

기업에서 사용 중인 검수 프레임워크와 본 논문에서 제시하는 검수 프레임워크와의 비교를 위해 검수에 필요한 소요비용 산출 공식을 정의한다[12]. 소요비용 산출에는 검수 대상(취약점)의 수, 서버대수(or 페이지 수), 점검회수, 및 인건비가 고려된다. 점검회수는 웹 애플리케이션의 중요도에 따라 달라진다. 예를 들어 개인정보를 가진 DB와 연계된 애플리케이션 등은 매일 점검이 필요할 것이다. 각 취약점의 산술적 집합으로 트랜잭션을 가정하였다. A Zone은 5개 취약점, B Zone은 25개(OWASP Top 25), C Zone은 50개로 가정한다. 인건비는 SW개발자가 취약점을 디버깅하는 시간에 비례한다. 검수 대상을 A Zone으로 한정하면, 1개 웹 서버를 1회 점검 후 디버깅했을 때는 반나절의 시간이 소요되었다. 각 취약점을 보완하는데 소요되는 시간이 동일하다고 가정하고, 미래부에서 발표한 SW개발자의 인건비 자료[13]를 고려하면, 취약점 당 약 1만원의 인건비가 필요하게 된다.

검수 비용 = Scan * Server * 점검 횟수 * 인건비
- Scan ; A, B, C Zone 등 점검 대상 취약점 수
※ 가정 : A Zone ; 5, B Zone : 25, C Zone : 50
- #Server(혹은 # Page) ; 대상 웹 서버(페이지)의 수
- 점검 회수 ; 취약점을 점검하는 회수
- 인건비 ; 각 취약점을 보완하는데 필요한 SW개발자의 인건비(취약점 1개당 1만원)
※ 가정 : 모든 취약점의 디버깅 시간은 동일

위 공식에 따라 점검 회수를 다르게 하여 시뮬레이션 한 결과를 Table 7과 Fig. 5로 나타내었다. Fig. 5를 보면 웹 애플리케이션 검수 비용이 서버대수나 점검 횟수가 늘어남에 따라 B Zone과 C Zone의 취약점을 검수하는 경우에서는 큰 폭으로 증가하여 기업 측에서 부담을 느낄 수 있다. 이에 반해 본 논문에서 제시하는 A Zone의 취약점을 검수하는 경우에서는 상대적으로 작은 폭으로 증가하여 비용이 감소되는 것을 확인할 수 있다. 비록 점검하는 취약점의 수는 줄였으나 [8]에서 제시한 주요 취약점만 검수해도 안전성이 보장된다는 결과를 통해 A Zone만 검수해도 안전하다는 것을 알 수 있다.

Table 7. Each zone Web vulnerability analysis cost simulation (unit : Thousands won)

zone \ 점검 회수	1	10	100	1,000	3,000
C	500	5,000	50,000	500,000	1,500,000
B	250	2,500	25,000	250,000	750,000
A	50	500	5,000	50,000	150,000

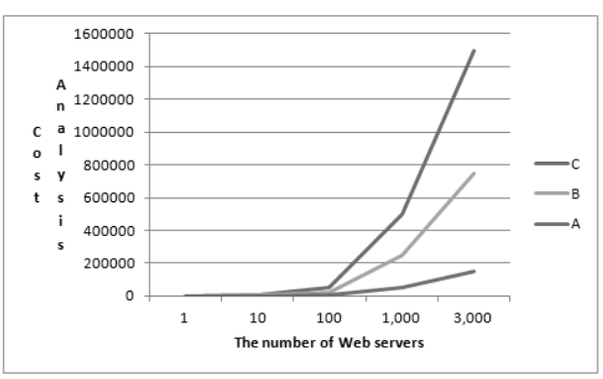

Fig. 5. Each zone Web vulnerability analysis cost

4.2 검수 프로세스 개선으로 인한 효과 검증 및 분석

본 논문에서는 검수 프로세스도 개선하였다. 제안하는 검수 프로세스에 대한 설명 및 기존과의 차이점은 3.2에서 자

세히 설명하고 있다. 검수 프로세스 개선에 의한 효과를 비교하기 위해 각 부서의 인력과 개인의 능력은 동일하다고 가정하고, 검수 프로세스의 각 단계를 다음 Table 8과 같이 검수 전, 검수 중, 검수 후의 3가지로 나누어서 비교하였다. A Zone 검수 방식과 조합되어 나타나는 효과도 같이 정리하였다.

검수 전 단계는 보안부서에서 검수를 시작하기까지의 단계가 포함된다. 기존 프로세스에서는 사업부서에서 개발부서로, 개발부서에서 보안부서로의 요청으로 인해 각 부서간의 의사 전달 시간이 필요하다. 하지만 제안하는 프로세스에서는 보안부서에서 능동적으로 시작하기 때문에 각 부서간의 의사 전달 시간이 필요하지 않다. 따라서, 각 부서간의 의사 전달 시간을 줄인 만큼 제안하는 검수 프로세스가 더 빠르다. 그리고 보안부서에서 능동적으로 검수를 시작할 수 있기 때문에 새로운 취약점이 발생하면 보안부서에서 즉시 검수를 시작하고 웹 코드 수정을 요구할 수 있다.

검수 중 단계는 보안부서에서 웹 코드를 검수하고, 개발부서에서 웹 코드를 수정하고, 보안부서에서 수정된 웹 코드를 재검수하는 단계가 포함된다. 이 단계에서 기존 프로세스와 제안하는 프로세스에 포함되는 단계의 수는 차이가 없다. 하지만 제안하는 프로세스는 주기적인 검수로 인해 보고서 형식의 전자우편을 통해 단순하고 신속하게 통보하여 의사 전달 시간을 줄일 수 있다. 또한 A Zone 취약점을 점검하면, 보안부서는 운영 중인 웹 서버를 직접 분석할 수 있기 때문에 점검용 웹 서버를 구축할 시간이 필요하지 않고, 개발부서는 소량의 중요한 취약점에 대해서만 수정하기 때문에 신속한 보안 업데이트와 보안 코딩 학습이 가능해진다.

검수 후 단계는 검수가 완료된 후, 사업부서에서 검수된 웹 코드를 받아서 웹 서버에 게재하는 단계가 포함된다. 이 단계에서 기존 프로세스와 제안하는 프로세스는 차이가 없다.

위 분석을 통해 제시하는 검수 프로세스는 기존의 검수 프로세스에 비해 보안부서를 적극적으로 운영할 수 있고, 보안 검수의 시간적인 비용을 감소시킬 수 있으며, 개발부서의 보안 코딩 능력을 지속적으로 향상시킬 수 있다는 것을 확인할 수 있다.

5. 결 론

본 논문에서는 5가지 취약점을 분석하는 “A Zone” 검수 방식이 가장 비용 효과적인 방식으로 분석되었다. 이를 위하여 먼저 다음 사항을 연구하였다.

1) 웹 애플리케이션 취약점에 의한 사고 사례를 분석

2) 웹 취약점 수치화 연구 사례 분석

3) Gartner 보고에 의한 실제 취약점 Table 1을 분석

실제 취약점은 웹 코드로 인한 취약점인 DDoS공격, 전수공격, 알려지지 않은 공격 등을 제외한다면, [8]에서 언급한 5가지 취약점만 점검하면 대부분의 웹 애플리케이션에 대한 공격을 방어할 수 있음을 연구하였다.

또한 개선된 능동 웹 취약점 검수 프로세스는 많은 웹 서버를 가진 기업에는 필수적인 업무 구조가 될 것이다. 이는 많은 웹 서버를 가진 국내 대표적인 인터넷 기업에서도 증명되었고, 70만개 이상의 웹 페이지를 가진 UN 서버 등에서도 검증된 바 있다[14]. 따라서 이 기법을 각 기업에서 사용하도록 법적 요구사항 변경을 검토해야 한다. 이 기법은 다음 Table 9와 같은 특징이 있다.

본 논문에서는 5가지 취약점만 검수하는 방법과 능동 웹 취약점 검수 프로세스를 이용하여 개선된 능동 보안 검수 프레임워크를 제안하였다. 제안하는 검수 프레임워크는 기존의 방법에 비해 다음과 같은 장점이 있다.

1) 사업부서와 개발부서의 요청 없이 보안부서에서 운영중인 웹 서버를 검수할 수 있으며, 매우 적극적으로 보안 업무에 나설 수 있기 때문에 새로 취약점이 발견되어도 빠르게 대처할 수 있다.

2) 보안 검수에 걸리는 시간이 짧기 때문에 자주 개발되고 수정되는 비즈니스 환경에 맞추어, 신속하게 보안 검수를 실시할 수 있다.

Table 8. Comparison of the existing process and proposed process

절차	기존 프로세스의 단계	제안하는 프로세스의 단계	비고
검수 전	①, ②	①	제안하는 프로세스가 빠름
검수 중	③, ④, ⑤	②, ③, ④	제안하는 프로세스가 빠름
검수 후	⑥, ⑦	⑤, ⑥	동일

Table 9. Advantages of the proposed verification inspection

평가기준	내용	비고
효과성(Effective)	실제 공격 취약점을 분석	Best Security Practice
효율성(Efficiency)	온라인 검수 프로세스로 신속 처리	비즈니스에 적합한 신속 처리
법적요구사항(Compliance)	핵심 중요사항 점검	법적 요구사항 재검토 필요

3) 개발부서에 웹 코드에 존재하는 취약점을 신속히 알릴 수 있고, 개발부서의 보안 코딩 능력을 향상시킬 수 있다.

웹 취약점 제거와 안전한 웹 운영은 안전한 웹 코딩을 통해 이루어져야 한다. 이를 위해서는 취약점 분석을 꾸준히 진행하여 보안 업데이트를 하고, 웹 코딩 개발자를 훈련시켜야 한다. 또한 계열사 및 협력업체를 가진 대기업과 지방자치단체를 가진 중앙정부부서의 경우, 유관 기업과의 DB 공유 사례가 많기 때문에, 유관 기업도 함께 웹 취약점 검수를 해야 한다. 그러므로 여러 계열사나 협력사를 운영하는 대기업 혹은 정부기관 등에서는 제안하는 능동 보안 검수 프레임워크를 중앙 집중적으로 운영한다면, 가장 비용 효율적인 보안 검수 관리체계를 가지게 될 것이다.

References

[1] 박태훈, "허술한 관리로 개인정보 해킹당한 '뽐뿌', 과징금 1억 1700만원" [Internet], http://www.segye.com/content/html/2015/11/20/20151120002048.html.

[2] 김민석, "'뽐뿌 해킹은 웹 취약점 DB 공격'… 200만 개인정보 털린 이유 밝혀져" [Internet], http://news.kukinews.com/article/view.asp?arcid=0009977886&code=41151111&cp=nv.

[3] 한국인터넷진흥원(KISA), "Mass SQL Injection 피해 DB 복구 방안," 2009.

[4] 전상훈, "웹 보안성 검수방법론," 2007.

[5] Amrit T. Williams, Neil MacDonald, "Organizations Should Implement Web Application Security Scanning," Gatner, 2005.

[6] 윤재섭, "현대캐피탈 사태 사고 예방 소홀한 '인재', 금감원 임직원 책임 묻기로" [Internet], http://ruliweb.daum.net/news/view/MD20110518193210260.daum.

[7] 한국일보, "현대캐피탈 해킹 어떻게 이루어졌을까," 2011.

[8] Sungyoung Cho, Suyeon Yoo, Sang-hun Jeon, Chae-ho Lim, and Sehun Kim, "A Web application vulnerability scoring framework by categorizing vulnerabilities according to privilege acquisition," *Journal of The Korea Institute of Information Security and Cryptology*, Vol.22, No.3, pp.601-613, 2012.

[9] NIST, "Information security handbook: a guide for managers," *The National Institute of Standards and Technology*, p.19, 2006.

[10] FIRST, "Common Vulnerability Scoring System" [Internet], http://www.first.org/cvss/.

[11] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System," *IEEE Security & Privacy*, Vol.4, No.6, pp.85-89, 2006.

[12] 김태형, "사이버전 무기 '악성코드' 감염 방지대책" [Internet], http://www.boannews.com/media/view.asp?idx=49835&kind=3.

[13] 빈꿈, "개발자 임금은 'SW 기술자 노임 단가'보다 훨씬 적다" [Internet], http://emptydream.tistory.com/3640.

[14] Bob Martin, "Common Weakness Scoring System (CWSS)" [Internet], http://cwe.mitre.org/cwss.

한 경 현

e-mail : co112kr@naver.com

2015년 홍익대학교 컴퓨터정보통신공학과 (학사)

2015년~현 재 홍익대학교 전자전산공학과 석사과정

관심분야 : 사이버보안

Trong-Kha Nguyen

e-mail : nguyentrongkha92@gmail.com

2015년 베트남 PTIT 전자통신학과(학사)

2015년~현 재 홍익대학교 전자전산공학과 석사과정

관심분야 : 사이버보안

조 훈

e-mail : h.joe@kaist.ac.kr

1999년 영국 The London College 컴퓨터공학(학사)

2001년 영국 King's College London 컴퓨터과학(석사)

2003년 삼성SDS 책임연구원

2012년~현 재 한국과학기술원 전산학과 박사과정

관심분야 : 정보보호 및 클라우드 컴퓨팅 보안

황 성 운

e-mail : sohwang@hongik.ac.kr

1993년 서울대학교 수학과(학사)

1998년 포항공과대학교 정보통신학과(석사)

2004년 한국과학기술원 전자전산학과(박사)

2008년~현 재 홍익대학교 컴퓨터정보통신공학과 교수

관심분야 : 정보보호

임 채 호

e-mail : chlim@kaist.ac.kr

1986년 홍익대학교 전산학과(학사)

1990년 건국대학교 전산학과(석사)

1991년 KIST/시스템공학연구소 선임연구원

1995년 우송정보대학 교수

2000년 한국정보보호진흥원 책임연구원

2001년 홍익대학교 전산학과(박사)

2003년 한국과학기술원 전산학과 교수

2005년 시큐리티맵(주) 대표이사

2006년 NHN(네이버) 보안실장

2008년 보안뉴스 연구센터장

2010년~현 재 한국과학기술원 전산학과 교수

관심분야 : 사이버보안