웹 취약점 점검 및 취약점별 조치 우선 순위 산정에 관한 연구

성 종 혁*, 이 후 기**, 고 인 제*, 김 귀 남***

요 약

오늘날 우리는 웹 사이트의 홍수 속에 살고 있으며, 다양한 정보를 얻기 위해서 인터넷을 통해 수많은 웹 사이트에 접속을 하고 있다. 하지만 웹 사이트의 보안성이 담보되지 않는다면, 여러 악의적인 공격들로부터 웹 사이트의 안전성을 확보할 수가 없다. 특히 금전적인 목적, 정치적인 목적 등 다양한 이유로 웹 사이트의 보안 취약점을 악용한 해킹 공격이 증가하고 있다. SQL-Injection, 크로스사이트스크립트(XSS), Drive-By-Download 등 다양한 공격기법들이 사용되고 있으며, 그 기술 또한 갈수록 발전하고 있다. 이와 같은 다양한 해킹 공격들을 방어하기 위해서는 웹 사이트의 개발단계부터 취약점을 제거하여 개발하여야 하지만, 시간 및 비용 등 여러 문제들로 인해 그러지 못하는 것이 현실이다. 이를 보완하기 위해 웹 취약점 점검을 통해 웹 사이트의 취약점을 파악하고 조치하는 것이 중요하다. 이에 본 논문에서는 웹 취약점 및 진단기법에 대해 알아보고 실제 웹 취약점 진단 사례를 통해 각 사례별 조치현황에 따른 개발단계에서의 취약점별 조치 우선 순위에 대해 알아보고자 한다.

A Study on Web Vulnerability Assessment and Prioritization of Measures by Vulnerabilities

JongHyuk Seong*, HooKi Lee**, InJe Ko*, Kuinam J. Kim***

ABSTRACT

Today we live in a flood of web sites and access numerous websites through the Internet to obtain various information. However, unless the security of the Web site is secured, Web site security can not be secured from various malicious attacks. Hacking attacks, which exploit Web site security vulnerabilities for various reasons, such as financial and political purposes, are increasing. Various attack techniques such as SQL-injection, Cross-Site Scripting(XSS), and Drive-By-Download are being used, and the technology is also evolving. In order to defend against these various hacking attacks, it is necessary to remove the vulnerabilities from the development stage of the website, but it is not possible due to various problems such as time and cost. In order to compensate for this, it is important to identify vulnerabilities in Web sites through web vulnerability checking and take action. In this paper, we investigate web vulnerabilities and diagnostic techniques and try to understand the priorities of vulnerabilities in the development stage according to the actual status of each case through cases of actual web vulnerability diagnosis.

Key words : 웹 취약점, 웹 취약점 점검, 시큐어 코딩, OWASP, CWE/SANS

접수일(2018년 8월 31일), 게재확정일(2018년 9월 23일)

* 경기대학교 산업보안학과
** 숭실대학교 IT정책경영학과
*** 경기대학교 융합보안학과(교신저자)

1. 서 론

오늘날 대부분의 기업, 기관 등은 자신만의 웹 사이트를 운영하고 있으며, 대다수의 사용자들은 웹 사이트에 접속함으로써 다양한 정보를 얻을 수 있다. 하지만 사용자가 웹 사이트에 접속할 시 보안성이 담보되지 않은 웹 사이트는 악의적인 공격에 취약할 수 있으며 이로 인해 심각한 위협에 직면할 수 있다. 시스템마다 차이는 있지만, 일반적으로 천라인(KLOG당 5개에서 50개 정도의 오류가 발견된다.[1] 대부분의 상업 소프트웨어들은 좀 더 많은 오류를 가지고 있으며, 이러한 문제들이 보안 취약점과 연관성을 가지고 있다고 보고되고 있다.[2-3] 실제, 보고된 보안사고의 90% 정도는 소프트웨어의 설계나 코드의 결점들을 악용(exploits)함으로써 발생한다고 한다.[4] 이와 같은 악의적인 공격 또는 취약점으로부터 웹 사이트를 보호하는데 사용되는 여러 유형의 보안기술을 웹 보안이라고 한다. 특히, 다양한 방식의 웹 취약점 점검을 통해 이와 같은 웹 취약점을 조기에 발견하고 이를 통해 선제적으로 취약점을 제거할 수 있다.

본 논문에서는 주요 웹 취약점 유형, 주요 취약점에 대한 진단 기준 및 조치 방법 등에 대해 알아보고, 이를 바탕으로 실제 웹 취약점 점검 사례를 통해 개발 단계에서 취약점별 조치 우선 순위를 산정하고자 한다.

2. 관련 연구

홈페이지를 운영하면서 어떤 부분이 취약한지, 보완해야할 점은 어떤 부분인지 정확하게 알고 있어야 보안사고를 사전에 예방할 수 있다. 하지만 다양한 취약점을 모두 파악하고 점검하는 것은 현실적으로 불가능하기 때문에 점검 기준을 마련하는 것이 중요하다. 점검 기준에 대해 알아보기 전에 먼저 보안약점과 보안취약점에 대해 간단히 알아보고, OWASP Top 10, CWE/SANS TOP 25, 국가정보원 8대 취약점 등 주요 보안약점 유형에 대해 알아보고자 한다.

2.1 보안취약점 및 보안약점

보안취약점은 공격자에 의해 실제 침해로 이어지는 소프트웨어의 허점이다. 특정 소프트웨어의 특정 부분과 같은 구체적인 사례로 제시되며, 사용 중인 소프트웨어의 문제점으로 발표되는 것이 일반적이다. 보안약점은 보안취약점이 될 수 있는 일반적인 형태를 말한다.[5]

보안약점 목록의 대표적인 사례로는 CWE(Common Weakness Enumeration)를 들 수 있다.[6] CWE는 미국 DHS(Department of Homeland Security)의 지원하에 MITRE에서 관리하고 있으며, 다음과 같은 보안약점 항목들로 이루어져 있다.[5]

- 관점(View) : CWE 내의 보안약점을 특정 관점에 의하여 분류한 부분목록이며, 단순나열 또는 비순환 그래프의 형태를 가진다. 현재 32개 관점이 제공되고 있다.
- 카테고리(Category) : 공통점을 가진 보안약점들의 집합으로, 244개의 카테고리가 제시되어 있다.
- 보안약점(Weakness) : 실제적인 개별 보안약점 항목들로서 현재 719개 항목이 포함되어 있다. 항목의 일반성 정도에 따라 Class, Base, Variant로 구분된다.
- 복합원소(Compound Element) : 여러개의 약점이 모여서 하나의 취약한 보안약점을 이루는 경우로서 8개 항목이 포함되어 있으며, 대표적인 예로 CSRF(Cross Site Request Forgery)를 들 수 있다.

2.2 주요 보안약점 유형

CWE의 경우 727개의 보안약점을 제공하고 있으나 모든 항목에 주의하면서 프로그램을 개발하는 것은 실질적으로 어렵다. 이를 위해 중요한 보안약점들의 목록을 제시하기 위한 활동이 이루어지고 있으며, 대표적인 주요 보안약점 유형으로 OWASP Top 10, SANS Top 25, 국가정보원 8대 취약점 등이 있다.

2.2.1 OWASP TOP 10

국제웹보안표준기구에서는 4년마다 가장 위험한 웹 애플리케이션 10대 보안위협을 발표하고 있다. OWASP의 Top 10 프로젝트의 주요 목표는 조직에서 직면한 가장 중요한 몇 가지 위험요소를 식별해 어플리케이션 보안에 대한 인식을 향상시키는 데 있다. 소프트웨어는 갈수록 복잡해지고 있어 어플리케이션 보

안을 달성하는 어려움 역시 더욱 더 힘들어지고 있다. 이에 따라 위험을 더욱 빠르고 정확하게 발견하는 것이 더욱 중요하게 되었다. OWASP Top 10을 통해 이와 같은 어려움을 관리하고 예방할 수 있는 방법을 제시하고 있다.[7-8]

<표 1> OWASP TOP 10 2017

순위	OWASP TOP 10 2017
A1	인젝션
A2	인증 및 세션 관리 취약점
A3	크로스 사이트 스크립팅(XSS)
A4	취약한 접근 제어
A5	보안 설정 오류
A6	민감 데이터 노출
A7	공격 방어 취약점(신규)
A8	크로스 사이트 요청 변조(CSRF)
A9	알려진 취약점 있는 컴포넌트 사용
A10	취약한 API(신규)

• A1 - 인젝션

SQL, OS, XXE, LDAP 인젝션 취약점은 신뢰할 수 없는 데이터가 명령어나 쿼리문의 일부분이 인터프리터로 보내질 때 발생한다. 공격자의 악의적인 데이터는 예상하지 못하는 명령을 실행하거나 적절한 권한 없이 데이터에 접근하도록 인터프리터를 속일 수 있다.

• A2 - 인증 및 세션관리 취약점

인증 및 세션관리와 관련된 어플리케이션 기능이 종종 잘못 구현되어 공격자에게 취약한 암호, 키 또는 세션 토근을 제공하여, 다른 사용자의 권한을(일시적으로 또는 영구적으로) 얻도록 익스플로잇 한다.

• A3 - 크로스 사이트 스크립팅(XSS)

XSS 취약점은 어플리케이션이 적절한 유효성 검사 또는 이스케이프 처리없이 새 웹 페이지에 신뢰할 수 없는 데이터를 포함하거나 JavaScript를 생성할 수 있는 브라우저 API를 사용하여 사용자가 제공한 데이터로 기존 웹 페이지를 업데이트 한다. XSS를 사용하면 공격자가 희생자의 브라우저에서 사용자 세션을 도용하거나, 웹 사이트를 변조시키거나, 악성사이트로 리다이렉션 시킬 수 있다.

• A4 - 취약한 접근 제어

인증된 사용자가 수행할 수 있는 작업에 대한 제한이 제대로 적용되지 않는다. 공격자는 이러한 결함을 악용하여 다른 사용자의 계정에 액세스하거나, 중요한 파일을 보고, 다른 사용자의 데이터를 수정하거나, 접근 권한을 변경하는 등 권한 없는 기능 및 root 디렉터리 또는 데이터에 접근할 수 있다.

• A5 - 보안 설정 오류

바람직한 보안은 어플리케이션, 프레임워크, WAS, 웹 서버, DB 서버 및 플랫폼에 대해 보안 설정이 정의되고 적용되어 있어야 한다. 보안 기본 설정은 대부분 안전하지 않기 때문에, 정의, 구현 및 유지되어야 한다. 또한 소프트웨어는 최신 버전으로 관리되어야 한다.

• A6 - 민감 데이터 노출

대부분의 웹 어플리케이션과 API는 금융정보, 건강정보, 개인식별정보와 같은 민감정보를 제대로 보호하지 않는다. 공격자는 신용카드 사기, 신분 도용 또는 다른 범죄를 수행하기 위해 취약한 데이터를 훔치거나 변경할 수 있다. 브라우저에서 중요 데이터를 저장 또는 전송할 때 특별히 주의하여야 하며, 암호화와 같은 보호조치를 취해야 한다.

• A7 - 공격 방어 취약점

대부분의 어플리케이션과 API에는 수동 공격과 자동 공격을 모두 탐지, 방지 및 대응할 수 있는 기본 기능이 없다. 공격 보호는 기본 입력 유효성 검사를 훨씬 뛰어넘으며 자동 탐지, 로깅, 응답 및 익스플로잇 시도 차단을 포함한다. 어플리케이션 소유자는 공격으로부터 보호하기 위해 패치를 신속하게 배포할 수 있어야 한다.

• A8 - 크로스 사이트 요청 변조(CSRF)

CSRF 공격은 로그인 된 피해자의 취약한 웹 어플리케이션에 피해자의 세션 쿠키와 기타 다른 인증 정보를 자동으로 포함하여, 위조된 HTTP 요청을 강제로 보내도록 한다. 예를 들어, 공격자가 취약한 어플리케이션이 피해자의 정당한 요청이라고 오해할 수 있는 요청들을 강제로 만들 수 있다.

• A9 - 알려진 취약점이 있는 컴포넌트 사용

컴포넌트, 라이브러리, 프레임워크 및 다른 소프트웨어 모듈은 어플리케이션과 같은 권한으로 실행된다. 이러한 취약한 컴포넌트를 악용하여 공격하는 경우 심각한 데이터 손실이 발생하거나 서버가 장악된다.

알려진 취약점이 있는 구성 요소를 사용하는 어플리케이션과 API는 어플리케이션을 약화시키고 다양한 공격과 영향을 줄 수 있다.

• A10 - 취약한 API

최신 어플리케이션에는 일종의 API(SOAP, XML, REST / JSON, RPC, GWT 등)에 연결되는 브라우저 및 모바일 어플리케이션의 JavaScript와 같은 여러 클라이언트 어플리케이션 및 API가 포함되는 경우가 많다. 이러한 API는 대부분 보호되지 않으며 수많은 취약점을 포함한다.

2.2.2 CWE/SANS TOP 25

SANS와 미국 및 유럽의 여러 소프트웨어 보안전문가 등에 의해 CWE/SANS TOP 25라는 이름으로 CWE에 등록된 1000여개의 SW 취약점 중 소프트웨어 개발자가 범하기 쉽고 위험한 25가지 취약점 목록을 유형별로 분리해 놓았다.[9]

CWE(Common Weakness Enumeration)는 근본적, 원인 측면의 보안 약점을 의미하며, SW의 소스적 특징에 의한 취약점이므로 개수가 한계가 있으며 'CWE+고유번호'로 나타낸다.[9-10]

CVE(Common Vulnerabilities and Exposure)는 원인에 기반한 결과, 현상적 측면의 보안취약점을 의미하며 'CVE+발견년도+고유번호'로 나타낸다.[9]

2011년에는 중요 보안약점의 선별을 위하여 중요도 정량평가 방법인 CWSS(Common Weakness Scoring System)가 사용되었으며[11] <표 2>는 산출된 25개 항목의 순위와 중요도 점수를 보여준다.

<표 2> CWE/SANS TOP 25

순위	점수	CWE-ID	이름
1	93.8	89	SQL 인젝션
2	83.3	78	운영체제 명령어 삽입
3	79.0	120	버퍼 오버플로우
4	77.7	79	크로스 사이트 스크립트
5	76.9	306	적절한 인증 없는 중요 기능 허용
6	76.8	862	권한 미인가
7	75.0	798	코드에 직접 삽입된 신용정보 사용
8	75.0	311	민감 데이터의 암호화 부재
9	74.0	434	위험한 파일 업로드
10	73.8	807	신뢰할 수 없는 입력값에 의존한 보안 결정
11	73.1	250	부적절한 권한을 통한 실행
12	70.1	352	크로스사이트 요청 변조(CSRF)
13	69.3	22	경로 순회
14	68.5	494	무결성 검사 없는 코드 다운
15	67.8	863	부정확한 권한 인가
16	66.0	829	신뢰할 수 없는 영역에서 제공되는 기능 포함
17	65.5	732	주요 자원에 대한 잘못된 권한 설정
18	64.6	676	잠재적 위험성이 있는 함수 사용
19	64.1	327	취약/위험한 암호화 알고리즘 사용
20	62.4	131	버퍼 크기의 부정확한 계산
21	61.5	307	과도한 인증 시도에 대한 부적절한 제한
22	61.1	601	신뢰하지 않는 URL주소로 자동 연결
23	61.0	134	제어되지 않는 문자열
24	60.3	190	정수 오버플로우
25	59.9	759	솔트 없는 단방향 해시 사용

또한 SANS TOP 25 취약점은 크게 3가지의 카테고리로 분류하고 있다. 구성요소 간 안전하지 않은 상호작용으로는 1,2,4,9,12,22 순위의 6개 취약점이, 위험한 자원 관리에는 3,13,14,16,18,20,23,24 순위의 8개 취약점이, 방어미비에는 5,6,7,8,10,11,15,17,19,21,25의 11개 취약점으로 분류할 수 있다.

2.2.3 국가정보원 8대 취약점

국가정보원은 [홈페이지 보안관리 매뉴얼]을 통해 홈페이지 8대 취약점을 정의하고 있다.[12]

<표 3> 홈페이지 8대 취약점

순서	내용
1	디렉토리 리스팅 취약점
2	파일 다운로드 취약점
3	크로스 사이트 스크립트(XSS) 취약점
4	파일 업로드 취약점
5	WebDAV 취약점
6	테크노트(Technote) 취약점
7	제로보드(Zeroboard) 취약점
8	SQL Injection 취약점

홈페이지 보안취약점은 웹서버 프로그램 또는 컴퓨

터의 운영 환경 설정 실수에 대부분 기인하지만 개발 중에 보안요소를 고려하지 않아 웹 응용프로그램에 내재된 보안결함으로 인한 경우도 상당히 많다.

3. 취약점별 우선 대응 순위 산정

3.1 웹 취약점 점검 항목 선정

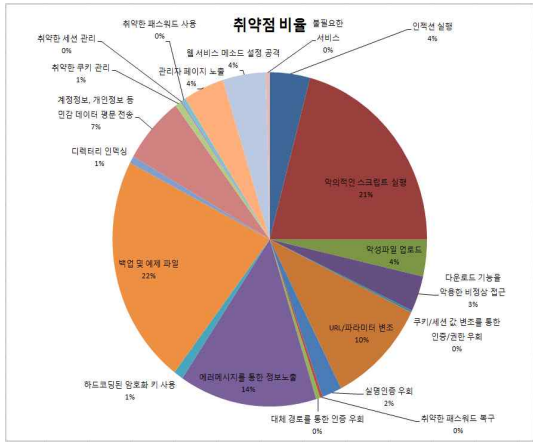
본 논문에서는 OWASP Top 10, CWE/SANS Top 25, 국가정보원 8대 취약점, 행정안전부 47개 기준 등 주요 취약점 점검 기준 등을 참조하여 20개 항목을 선정하여 100개 홈페이지에 대해 취약점 점검을 수행하였다. 취약점 점검 항목은 <표 4>와 같다.

<표 4> 취약점 점검 항목

분류	세부 항목
인젝션	인젝션 실행
XSS	악의적인 스크립트 실행
파일업로드	악성파일 업로드
파일다운로드	다운로드 기능을 악용한 비정상 접근
CSRF	쿠키/세션 값 변조를 통한 인증/권한 우회
불충분한 인증 및 인가	URL/파라미터 변조
	실명인증 우회
	취약한 패스워드 복구
	대체 경로를 통한 인증 우회
정보유출 및 부적절한 에러처리	에러메시지를 통한 정보노출
	하드코딩된 암호화 키 사용
	백업 및 예제 파일
	디렉터리 인덱싱
데이터 평문 전송	계정정보, 개인정보 등 민감 데이터 평문 전송
취약한 인증 및 세션 관리	취약한 쿠키 관리
	취약한 세션 관리
약한 문자열강도	취약한 패스워드 사용
URL접근통제 실패	관리자 페이지 노출
보안설정 우회	웹 서비스 메소드 설정 공격
정보	불필요한 서비스

3.2 항목별 취약점 비율

취약점 점검 수행 결과 각 세부 항목별 취약점 비율은 (그림 1)과 같다.



(그림 1) 항목별 취약점 비율

백업 및 예제 파일 노출 취약점이 22.20%로 가장 많이 발견되었으며, 악의적인 스크립트 실행 20.87%, 에러메시지를 통한 정보노출 14.15%, URL/파라미터 변조 10.29%, 계정정보 및 개인정보 등 민감 데이터 평문 전송 6.51%순으로 많이 발견되었으며, 취약한 세션 관리, 쿠키/세션 값 변조를 통한 인증/권한 우회, 취약한 패스워드 복구, 대체경로를 통한 인증 우회, 취약한 패스워드 사용, 불필요한 서비스 등은 0.5% 이하로 상대적으로 적게 발견되었다.

3.3 항목별 미조치 비율

취약점 점검을 수행한 후 이에 대한 긴급하게 조치하는 것이 무엇보다 중요하다고 할 수 있다. 각 항목별 조치 기간 설정 후, 조치 기간 내 해당 홈페이지의 미조치율을 산정하였다. 미조치율이 높을수록 조치가 힘든 것으로 판단할 수 있으며, 이 항목들은 개발 단계에서 우선적으로 고려하여 개발을 하여야 한다.

<표 5> 세부항목별 미조치율

세부 항목	미조치율
인젝션 실행	5%
악의적인 스크립트 실행	22%
악성파일 업로드	14%
다운로드 기능을 악용한 비정상 접근	16%
쿠키/세션 값 변조를 통한 인증/권한 우회	0%
URL/파라미터 변조	21%
실명인증 우회	14%
취약한 패스워드 복구	50%
대체 경로를 통한 인증 우회	50%
에러메시지를 통한 정보노출	23%
하드코딩된 암호화 키 사용	36%
백업 및 예제 파일	15%
디렉터리 인덱싱	0%
계정정보,개인정보 등 민감데이터 평문전송	29%
취약한 쿠키 관리	22%
취약한 세션 관리	0%
취약한 패스워드 사용	17%
관리자 페이지 노출	22%
웹 서비스 메소드 설정 공격	21%
불필요한 서비스	29%

취약점 조치 결과 쿠키/세션 값 변조를 통한 인증/권한 우회, 디렉터리 인덱싱, 취약한 세션 관리 취약점은 미조치율이 0%로 조치 기한 내 모든 취약점이 조치되어 상대적으로 취약점 조치가 수월한 것으로 나타났으며, 취약한 패스워드 복구, 대체 경로를 통한 인증 우회, 하드코딩된 암호화 키 사용 등의 취약점 항목은 30%가 넘는 미조치율을 보여 상대적으로 취약점 조치가 힘든 항목으로 나타났다. 모든 항목에 대한 취약점 조치가 중요하지만 미조치율이 높은 취약점 항목에 대해서는 개발 단계에서부터 선제적으로 대응하는 것이 중요하다고 볼 수 있다.

3.4 주요 항목별 해결책

주요 미조치 항목별 취약점 조치 방법은 다음과 같다.

3.4.1 취약한 패스워드 복구

취약한 패스워드 복구는 취약한 패스워드 복구 로직(패스워드 찾기 등)으로 인하여 공격자가 불법적으로 다른 사용자의 패스워드를 획득 및 변경할 수 있는 취약점으로, 해결책으로는 사용자의 개인정보로 패스워드를 생성하지 말아야 하며, 난수를 이용한 불규칙적이고 최소 길이 이상의 패턴이 없는 패스워드를 발급하여야 한다. 또한 소스코드 시큐어 코딩을 적용하여 사용자 비밀번호 찾기 시 데이터베이스에 저장되어 있는 사용자 정보와 입력한 사용자 정보가 일치하는지 비교하는 인증 구분을 적용하여야 한다.

3.4.2 대체 경로를 통한 인증 우회

대체 경로를 통한 인증 우회는 민감한 데이터에 접근 가능한 경로에 대한 인증 절차가 불충분할 경우 발생하는 취약점으로 사용자가 권한 외의 페이지에 접근하여 정보를 유출하거나 변조할 수 있는 취약점으로, 해결책으로는 개인정보 및 패스워드 수정 페이지와 같은 중요 정보를 표시하는 페이지에서는 본인 인증을 재확인하는 로직을 구현하고, 인증 후 사용자가 이용 가능한 페이지에 접근할 때마다 승인을 얻은 사용자인지 페이지마다 검증하여야 한다. 또한 접근 통제 정책을 구현하고 있는 코드는 구조화 및 모듈화가 되어 있어야 하고, 인증과정을 처리하는 부분에 Client Side Script(Javascript, VBScript 등)를 사용하면 사용자가 임의로 수정할 수 있으므로 Server Side Script(PHP, ASP, JSP 등)를 통하여 인증 및 필터링 과정을 수행하여야 한다.

3.4.3 하드코딩된 암호화키 사용

하드 코딩된 암호화키 사용은 서버 접속 정보나 주요 데이터가 암호화되거나 함수로 처리하지 않고 소스에 그대로 코딩되어 공격자에게 2차 공격을 하기 위한 중요한 정보를 제공할 수 있는 취약점으로, 해결책으로는 모든 웹 페이지에 대해 개발단계에서 디버깅 및 테스트를 목적으로 작성한 주석 구문에 서버 주요 정보가 포함되어 있을 경우 공격자가 해당 정보를 다른 취약점과 연계해 사용할 수 있으므로 제거해야 하고, html 소스 레벨에서 중요 정보를 마킹처리 등의 방법으로 사용자가 볼 수 없도록 코딩해야 한다.

4. 결 론

홈페이지 보안취약점을 이용한 외부 공격은 더욱더 증가를 하고 있으며, 그에 따른 보안 대책 수립 역시 중요하게 되었다. 시큐어코딩, 보안 패치 등을 통해 선

제적으로 외부 공격에 대응할 수 있음에도 불구하고 시간 및 비용 등의 이유로 보안 약점을 고려하지 않고 개발을 하는 경우가 많은 것 또한 사실이다. 본 논문에서는 100개의 홈페이지를 대상으로 주요 취약점 항목에 대해 취약점 점검을 수행하였고, 이에 대한 결과로 항목별 취약점 탐지 비율을 살펴보았다. 또한 항목별 미조치 비율을 산정하여 개발 단계에서 우선적으로 고려해야할 보안 취약점 순위를 산정하였다. 외부의 공격으로부터 홈페이지를 보호하기 위해서는 보안 취약점 점검을 통해 취약점 점검을 수행하는 것도 중요하지만, 개발 단계에서 보안을 고려한 개발을 통해 보안 취약점을 사전에 제거하는 것이 더욱 중요하다고 할 수 있다.

참고문헌

[1] 이진영, 김동진 등 7명, “CWE와 7PK 취약점 분류 비교”, 한국정보과학회 학술발표논문집 Vol.36, No.2(D), 2009. 11.

[2] F. Piessens, “A Taxonomy of Causes of Software Vulnerabilities in Internet Software.”, Augst. 2002.

[3] C. V. Berghe, J. riordan, F. Piessens, “A Vulnerability Taxonomy Methodology applied to Web Services”, Nordic Workshop on Secure IT System., Proceedings of the 10th NordSec, 2005.

[4] Katkar Anjail S., Kulkarni Raj B., "Web Security”, International Journal of Innovative Research & Development, Vol1, Issue8, pp. 448-458, 2012.

[5] 안준선, 이은영, 창병모, “SW 개발보안을 위한 보안약점 표준목록 연구”, 정보보호학회지, 제25권 제1호, 2015. 2.

[6] Common Weakness Enumeration(CWE), http://cwe.mitre.org/

[7] 2017 OWASP(The Open Web Application Security Project) Top 10, https://www.owasp.org/index.php/Top_10_2017-Top_10

[8] 이도현, 이종욱, 김점구, “멀티테넌시 기반 웹 사이트의 OWASP TOP 10 보안취약성 검증 방법”, 융합보안 논문지 제16권 제4호, 2016. 6.

[9] 2011 CWE/SANS Top 25 Most Dangerous Software Errors, http://cwe.mitre.ofg/top25/

[10] 한경숙, 김태환, 한기영, 임재명, 표창우, “대한민국 전자정부 소프트웨어 개발보안 가이드 개선 방안 연구”, 정보보호학회논문지, 제22권 제5호, 2012. 10.

[11] Common Weakness Scoring System(CWSS), http://cwe.mitre.org/cwss/

[12] 국가정보원, “홈페이지 보안관리 매뉴얼”, 2005. 05.

[저 자 소 개]

성 종 혁 (Jong-Hyuk Seong)
1999년 2월 영남대학교 컴퓨터공학 학사
2001년 2월 영남대학교 컴퓨터공학 석사
2018년 현재 경기대학교 산업보안학과 박사수료
email : jhseong@kcisa.kr

이 후 기 (Hool-Ki Lee)
2010년 2월 동국대학교 정보보호학 석사
2018년 2월 숭실대학교 IT정책경영학 박사
현재 국가기관 사이버보안담당관
email : hk0038@korea.kr

고 인 제 (In-Je Ko)
2006년 2월 신라대학교 경영학 학사
2009년 2월 부경대학교 정보보호협동과정 석사
2018년 현재 경기대학교 산업보안학과 박사수료
email : injeko@kcisa.kr

김 귀 남 (Kuinam J. Kim)
미국 캔자스대학(학사)
미국 콜로라도주립대학(석사)
미국 콜로라도주립대학(박사)
현재 경기대학교 융합보안학과 교수
email : kuinam@icatse.org