情 報 保 護 學 會 誌
第22卷 第3號, 2012. 5

웹 애플리케이션 취약성 수치화 프레임워크 연구 동향

조 성 영*, 전 상 훈**, 임 채 호***, 김 세 헌****

요 약

웹 애플리케이션의 취약점을 이용한 공격으로 인한 보안 사고가 급증함에 따라, 웹 애플리케이션의 취약점을 진단하고 취약성을 수치화하는 프레임워크가 개발 및 제시되고 있다. 이 논문에서는 현재 웹 애플리케이션의 취약성을 수치화할 수 있는 CVSS와 CWSS와 같은 여러 가지 프레임워크 연구를 분석하고, 현재까지 제시된 취약성 수치화 프레임워크가 가지고 있는 문제점을 제시함으로써 향후 보완해야 할 방향을 제시한다.

Ⅰ. 서 론

최근 빈번하게 발생하고 있는 보안사고로 인한 피해의 규모가 점점 커지고 있고, 이러한 피해를 최소화기 위한 노력이 여러 분야에서 진행되고 있다. 최근 발생하는 보안사고의 상당수는 웹 애플리케이션의 취약점을 이용한 공격으로 인한 것이며, 이러한 취약점을 사전에 진단하고 공격으로부터 보호하는 것이 효과적인 예방책이라고 할 수 있다.

현재 웹 애플리케이션의 취약점을 분석하고 결과를 제공하는 자동화된 점검도구들이 있으며, IBM Rational AppScan, Acunetix, WebInspect 등과 같은 상용 제품과 Paros, Grabber 등과 같은 오픈소스 프로그램, 비트스캐너(BitScanner)와 같은 SaaS (Software as a Service) 제품 등이 그 예이다. 이러한 취약점 점검 도구를 이용하여 웹 애플리케이션에 대한 취약점 점검을 함으로써 웹 애플리케이션의 설계 및 구현 단계에서부터 보안을 고려할 수 있다. 예를 들어 취약점 점검 도구는 프로그래머가 입력 값 버퍼의 길이를 적절히 부여하지 않음으로써 발생될 수 있는 버퍼 오버플로우(buffer overflow)와 같은 심각한 취약점에 대하여 진단하고 적절한 해결책을 취할 수 있다.

취약점을 분석하여 개별적인 결과를 알려 주는 것도 중요하지만, 취약성에 대한 수치화를 제공함으로써 현재 개인이나 기업, 조직에서 운영하고 있는 웹 애플리케이션의 취약성 정도를 파악하는 것 또한 중요하다. 현재 웹스캐너에서 제공하고 있는 취약성 수치는 수치화 범위가 제각각이고 취약성을 측정하기 위한 요소들이 다르며, 취약성에 대한 수치가 도출되었을 때 어떻게 그런 수치가 도출되었는지에 대하여 정확한 정보를 제공하고 있는 경우가 드물다.

이러한 문제점을 보완하면서 취약점들에 대한 효과적인 취약성 수치화를 위한 여러 프레임워크가 제안되었다. 이 논문에서는 CVSS와 CWSS 등과 같은 여러 취약성 수치화 프레임워크를 살펴보기로 한다.

Ⅱ. CVSS (Common Vulnerability Scoring System)

2.1 개요 및 특징

CVSS[1][2]는 하드웨어나 소프트웨어에서 발견되는 개별적인 취약점들에 대한 취약성 수치화 프레임워크이며, 미국 국토안보부 (Department of Homeland Security) 소속 국가 기반시설 보장위원회 (National Infrastructure

이 연구는 대한민국 지식경제부 정보통신진흥기금으로 수행되었으며, 정보통신산업진흥원(NIPA)의 관리로 진행된 사이버보안연구센터 지원사업(NIPA-H0701-12-1001)임.
* KAIST 정보보호대학원 (sungyoung.cho@kaist.ac.kr)
** KAIST 사이버보안 연구센터 (p4ssion@gmail.com)
*** KAIST 사이버보안 연구센터, 정보보호대학원 (chlim@kaist.ac.kr)
**** KAIST 산업 및 시스템 공학과, 정보보호대학원 (shkim@kaist.ac.kr)

Assurance Council) 에 의해 구상되고 2005년 처음 발표된 이후로, FIRST (Forum of Incident Response and Security Teams) 재단에 의하여 관리되고 있다.

CVSS는 다음과 같은 특징들을 가지고 있다.

(1) 표준화된 취약성 측정: 기존의 취약점 분석을 통한 취약성 측정 프레임워크들은 애플리케이션에 종속적이고 서로 다른 크기(scale)로 측정되기 때문에 측정된 결과가 제각각이었다. CVSS는 모든 IT 취약점들에 대해서 같은 측정 프레임워크를 적용함으로써 단일의 취약점 관리 정책을 취할 수 있어 어떤 취약점을 개선해야 하는지에 대한 빠른 판단을 돕는다.

(2) 문맥론적 수치화: CVSS는 노출된 취약점들로부터의 실제 위험(risk)을 대표하며, 어떤 개선 노력을 우선적으로 취해야 하는지에 관하여 우선순위를 매기는 것을 돕는다.

(3) 개방 프레임워크: 기존의 애플리케이션 제공자들이 매기는 취약성 수치는 어떻게 그런 수치가 나오게 되었는지 살펴보기 힘들다는 문제점이 있다. CVSS는 취약점에 대해 수치화를 함에 있어서 구체적인 사항을 제공한다. 즉 어떻게 수치가 산정되는지에 대하여 어떤 항목들이 영향을 주었는지, 이전의 수치와 비교하여 어떤 부분이 달라졌는지에 대하여 확인할 수 있음으로 인하여 투명하고 객관적인 취약성 수치에 대하여 신뢰할 수 있다.

2.2 측정 항목

CVSS는 다음과 같이 세 가지 영역으로 구성된다.

2.2.1 기본 영역 (Base metric)

기본 영역은 취약점의 시간과 환경과 무관한 고유하고 본질적인 특성을 나타낸다. 기본 영역에서 측정하는 요소에는 접근 경로 (access vector), 접근 복잡성 (access complexity), 인증 (authentication), 영향 (Impact)이 있다.

접근 경로는 어떻게 취약점이 익스플로잇(expliot) 되는지를 나타내고, 로컬, 근접한 네트워크, 네트워크 등의 요소로 측정된다.

접근 복잡성은 공격자가 대상 시스템에 접근할 수 있을 때 취약점을 익스플로잇하기 위하여 요구되는 공격의 복잡성을 나타내며, 높음, 중간, 낮음으로 측정된다.

인증은 취약점을 익스플로잇하기 위하여 대상 시스템에 인증하기 위해 요구되는 횟수를 나타내며, 여러 번, 한 번, 없음으로 측정된다.

영향은 취약점이 익스플로잇되었을 때 IT 자산에 직접적 영향을 측정하며, 기밀성(confidentiality), 무결성(integrity), 가용성(availability) 세 가지 측면에서 완전함, 부분적, 없음 항목으로 측정할 수 있으며, 세 요소는 서로 독립적이다.

2.2.2 시간 영역 (Temporal metric)

시간 영역은 취약점의 시간에 따라 변화하는 특성을 나타낸다. 시간 영역에서 측정하는 요소는 악용가능성(exploitability), 개선 등급(remediation level), 보고 신뢰성(report confidence)이 있다.

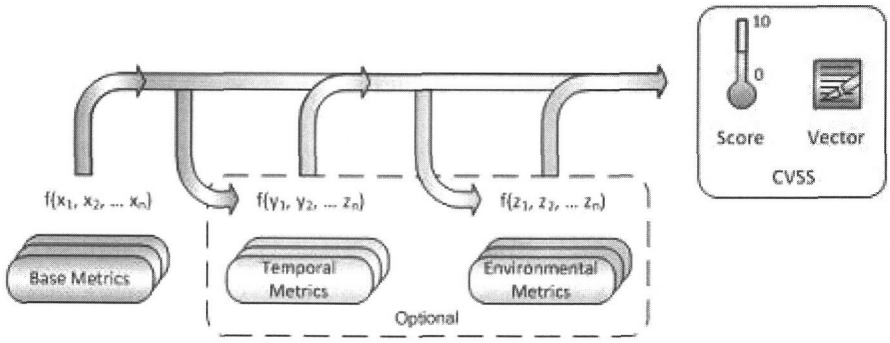
악용가능성은 취약점을 통한 익스플로잇 기술 또는 코드 가용성의 현재 상태를 말한다. 쉽게 사용 가능한 익스플로잇 코드의 가용성 증가는 비전문적인 공격자들도 취약점을 통하여 쉽게 공격할 수 있으며 이는 심각한 피해로 이어질 수 있다. 악용가능성은 증명되지 않음(unproven), 개념증명(proof-of-concept), 기능적(functional), 높음(high), 정의되지 않음 등으로 측정된다.

개선 등급은 취약점의 우선순위를 매기기 위한 중요한 요소이다. 취약점이 최초에 발견되었을 때는 핫픽스(hotfix)나 패치와 같은 개선사항이 없지만, 핫픽스를 통하여 취약점을 보완하고 최종적으로 공식 패치나 업데이트를 통하여 취약점을 개선한다. 따라서 이 항목에서는 공식적 픽스, 임시 픽스, 작업중, 불가능, 정의되지 않음 등으로 측정된다.

보고 신뢰성은 취약점의 존재에 대한 신뢰성의 정도와 알려진 기술적 사항에 대한 신뢰성을 확인되지 않음, 입증됨, 확인됨, 정의되지 않음 등으로 측정된다.

2.2.3 환경 영역 (Environmental metric)

환경 영역은 취약점의 사용자의 환경에 고유한 특성을 나타낸다. 환경 영역에서 측정하는 요소에는 부수적인 잠재적 손해(collateral damage potential), 대상 분포(target distribution), 보안 요구사항(security require-



〔그림 1〕 CVSS에서 각 요소들을 바탕으로 한 취약성 수치화 방법

ments)이 있다.

부수적인 잠재적 손해는 재산이나 장비의 손해 또는 위협을 통한 물리적인 자산이나 생활의 잠재적 손해를 나타내며, 없음, 낮음, 비교적 낮음, 비교적 높음, 높음, 정의되지 않음 등으로 측정된다.

대상 분포는 취약한 시스템의 비율을 측정한다. 이는 특정 취약점을 익스플로잇했을 때 영향을 받는 시스템의 비율을 나타내며, 없음, 낮음, 중간, 높음, 정의되지 않음 등으로 측정된다.

보안 요구사항은 기밀성, 무결성, 가용성 측면에서 사용자의 조직에 있는 IT 자산에 적용을 받는 요구사항에 대한 사항에 대해 보안상 요구사항을 측정하며, 기밀성, 무결성, 가용성 차원에서 낮음, 중간, 높음, 정의되지 않음 등으로 측정된다.

2.3 수치화 방법

기본 영역에서 정의된 요소들에 대한 수치는 0점에서 10점까지 부여되며, 각 요소들에 부여된 점수에 대하여 설명이 부가된 벡터가 생성된다. 이렇게 기본 영역에서 측정된 수치들은 시간 영역과 환경 영역의 요소들에 대해 수치를 산정하는 데 영향을 미치지만, 시간 영역과 환경 요소들에 대한 수치들은 요구사항은 아니다. 즉 CVSS를 사용하는 개인이나 조직의 목적에 따라 취약성을 수치화함에 있어 시간 영역과 환경 요소에 대한 수치에 대하여 반드시 수치화되어야 할 의무는 없고, 기본 영역에 대해서만 수치화되어 그들의 목적과 요구에 따라 취약성을 측정할 수 있다. 만약 시간 영역에 대한 수치화가 요구된다면, 기본 영역에서 측정된 요소들의 값에 따라 시간 영역의 각 요소들에 대한 값이 수치화되어 0점에서 10점까지 점수가 부여되며, 환경 영역에 대한 수치화 또한 요구된다면 이 요소들에 대한 값 또한 수치화되어 0점에서 10점까지 점수가 부여된다. 이러한 내용은 [그림 1]과 같이 표현될 수 있다.

CVSS의 공식 문서[1]에 기본 영역, 시간 영역, 환경 영역에 대하여 구체적인 수치를 부여하는 방법과 이를 바탕으로 개별 취약점에 대한 전체적인 취약성을 나타내는 수치를 계산하는 공식이 언급되어 있다.

2.4 한 계

CVSS는 앞에서 언급한 표준화된 취약성 측정, 수치화에 있어서 문맥론적 접근, 개방 프레임워크라는 장점에도 불구하고, 개별 웹 애플리케이션 내에서 발견된 여러 취약점들을 종합하여 취약성을 수치화할 수 있는 방법을 제시하고 있지 못하다. 즉 CVSS는 하나의 웹 애플리케이션에서 발견된 개별 취약점들에 대한 취약성을 수치화할 수 있을 뿐이다. 기업이나 조직에서 운영하고 있는 웹 애플리케이션 기반의 서비스는 여러 종류인데, 어떠한 서비스가 가장 취약하며, 한 서비스 내에서 어떤 취약점을 우선적으로 처리해야 하는지에 대한 방법을 제공하고 있지 못하다.

Ⅲ. CWSS (Common Weakness Scoring System)

3.1 개요 및 특징

CWSS[3]는 미국 국토안보부 국가 사이버보안 부서(National Cyber Security Division)의 소프트웨어 보장 프로그램의 지원을 받아 CWE(Common Weakness Enumeration) 프로젝트의 일환으로 고안된 취약성 수치화 프레임워크이다. 2010년 12월 최초 버전(0.1)이

나온 이후로 개정을 거쳐 2011년 6월 V. 0.8이 발표되었다. CWSS는 웹 애플리케이션 개발자, 개발 관리자, 웹 애플리케이션 소유자, 코드 분석가와 보안 컨설턴트 등에게 안전한 웹 애플리케이션의 설계와 구현, 배포와 관련하여 도움을 줄 수 있다.

CWSS의 특징은 다음과 같다.

(1) 웹 애플리케이션을 포함한 소프트웨어의 보안 취약점에 대해 우선순위를 정할 수 있는 프레임워크를 제공한다.

(2) 웹 애플리케이션을 포함한 소프트웨어에서 발견된 보안 취약점에 대해 정량적인 측정 방법을 제공한다.

(3) 개발자들로 하여금 개별 웹 애플리케이션에서 발견된 취약점들에 대해 우선순위를 정할 수 있다.

(4) 소프트웨어의 고객에게 있어 CWSS가 CWRAF (Common Weakness Risk Analysis Framework)와 함께 적용되어 소프트웨어에 대한 보장을 받기 위한 소프트웨어의 획득과 보호 활동에 있어 업무 영역에서 가장 중요한 취약점이 무엇인지 확인할 수 있다.

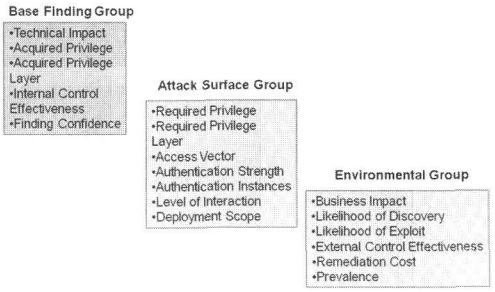
3.2 CWSS에서의 여러 가지 수치화 방법들

CWSS에서는 CWSS 프레임워크 내에서 지원되는 여러 가지 다른 수치화 방법에 대하여 언급하고 있다.

(1) **타킷(targeted) 방법**: 지정된(targeted) 웹 애플리케이션의 설계 및 구현 단계에서 발견된 개별적인 취약점들에 대한 취약성을 수치화하는 방법이다. 예를 들어 특정 웹 애플리케이션의 어떤 부분에서 버퍼 오버플로우가 발견되었을 때, 이 부분에 대한 취약성을 수치화한다.

(2) **일반화된(generalized) 방법**: 웹 애플리케이션의 종류와 관계없이 일반적인 웹 애플리케이션에서 발견될 수 있는 취약점들에 대한 우선순위를 정할 경우 각각의 취약점들에 대한 취약성을 수치화한 후 정렬한다. 가장 위험한 소프트웨어 취약점들을 목록화하여 발표하는 CWE/SANS Top 25[4]와 OWASP Top 10[5] 등에 사용되는 방법이다.

(3) **문맥론적-조정(context-adjusted) 방법**: 웹 애플리케이션의 취약점을 점검하고 취약성을 수치화함에 있어 업무의 성격이나 업무 환경에 존재하는 위협, 위험 내성 등과 같은 다른 요인을 고려하는 방법이다.



〔그림 2〕 CWSS에서 취약성을 측정하는 요소들

(4) **종합적(aggregated) 방법**: 여러 취약점들을 하나의 단일하고 총체적인 수치로 도출하기 위한 방법이다. 웹 애플리케이션 고객들이 구매하고자 하는 웹 애플리케이션이 얼마나 안전한지에 관하여 이 방법은 유용할 수 있지만, 현재 CWSS에서는 종합적 방법에 대하여 명확한 지침을 제공하지 않고 있으며 다음 수정 작업에 반영할 예정이라고 밝히고 있다.

3.3 측정 항목

CWSS의 측정 항목은 [그림 2]처럼 CVSS처럼 세 가지 영역으로 나누어 분석할 수 있다.

3.3.1 기본 발견 영역 (Base finding group)

기본 발견 영역은 취약점에 내재하는 위험, 취약점을 발견함에 있어 정확성의 신뢰도, 통제 강도와 같은 항목을 다룬다.

기술적 영향(technical impact)은 취약점을 익스플로잇했을 때 발생할 수 있는 잠재적인 결과에 대해서 설명하고 있으며, 심각, 높음, 중간, 낮음, 없음 등 또는 정량화된 방법으로 측정될 수 있다.

획득된 권한(acquired privilege)은 취약점을 익스플로잇하여 공격자가 얻을 수 있는 권한을 말한다. 이러한 권한에는 관리자 권한, 부분적 권한, 일반 사용자 권한 등이 있다.

획득된 권한 계층(acquired privilege layer)은 취약점을 익스플로잇함으로 인하여 얻은 권한의 계층을 말하

며, 애플리케이션 계층, 시스템 계층, 네트워크 계층, 엔터프라이즈 계층 등으로 측정될 수 있다.

내부 통제 효과(internal control effecti- veness)는 공격자들이 취약점을 익스플로잇하지 못하도록 내부적으로 통제하고 있는 능력을 말한다. 없음, 제한된 통제, 적당한 통제, 간접적 통제, 최선(best-available), 완전한 통제 등으로 측정될 수 있다.

발견 신뢰성(finding confidence)은 취약점과 이러한 취약점이 공격자에 의하여 유발되거나 이용 가능한지와 관련한 신뢰성을 언급한다. 이 요소는 정량화되어 측정할 수 있다.

3.3.2 공격 표면 영역 (Attack surface group)

공격 표면 영역은 공격자가 취약점을 익스플로잇하기 위하여 넘어야 하는 장벽(barrier)을 묘사하고 있다. 여기에는 요구 권한, 요구 권한 계층, 접근 경로, 인증 강도 등을 포함한다.

요구 권한(required privilege)은 취약점을 포함하고 있는 코드와 기능에 접근하기 위하여 요구되는 권한을 말한다. 일반 사용자, 게스트, 없음, 관리자, 부분적 권한 등이 있다.

요구 권한 계층(required privilege layer)은 취약점을 익스플로잇하기 위해 필요한 권한의 계층을 말하며, 시스템, 애플리케이션, 네트워크, 로컬 등이 있다.

접근 경로(access vector)는 취약점을 포함한 코드나 기능에 접근하기 위하여 통신해야만 하는 채널을 말하며, 인터넷, 인트라넷, 사설 네트워크, 무선 네트워크를 포함한 인접 네트워크, 로컬, 물리적 채널 등이 있다.

인증 강도(authentication strength)는 취약점을 포함한 코드나 기능에로의 접근을 보호하기 위한 인증의 강도를 말한다.

인증 인스턴스(authentication instance)는 취약점을 포함한 코드나 기능에 접근하기 위해 반드시 거쳐야 하는 인증 인스턴스의 수를 말하며, 없음, 하나, 여러 개 등으로 측정된다.

상호작용 등급(level of interaction)은 사람의 상호작용으로 인하여 공격이 성공적으로 이루어질 때 그러한 상호작용의 정도를 나타내며, 제한적(일반적), 적당함, 기회적(opportunistic), 높음, 상호작용 없음 등으로 측정된다.

배치 범위(deployment scope)는 취약점이 발견되는 범위를 나타내는 것으로써, 모든 플랫폼이나 구성에서 발견되는지, 공통된 플랫폼이나 구성에서 발견되는지, 드물게 발견되는지, 아니면 잠재적인지 등에 대한 여부를 판단한다. 이 요소는 정량화되어 측정할 수 있다.

3.3.3 환경 영역 (Environmental group)

환경 영역은 업무 영향이나 익스플로잇 가능성, 외부 통제의 존재 등과 같은 특수한 운영적 맥락에 관한 요소들을 포함한다.

업무 영향(business impact)은 취약점이 익스플로잇됨으로써 발생할 수 있는 업무 또는 임무에 대한 잠재적 영향을 설명한다. 이 요소는 중요함, 중간, 낮음, 없음 등으로 측정되며, 정량적으로 측정될 수도 있다.

발견 가능성(likelihood of discovery)은 공격자가 취약점을 발견할 수 있는 가능성을 설명한다. 하지만 취약점의 발견이 이미 보고되었다면 다시 그 취약점은 언제든지 발견될 수 있는 것이고, 공격 경로, 배치 범위, 인증 인스턴스, 상호작용 등급 등과 같은 다른 CWSS의 측정 요소에 의해 영향을 받는다는 등의 이유로 이 요소는 차기 버전에서 빠지게 될 예정이다.

익스플로잇 가능성(likelihood of exploit)은 취약점이 발견되었을 때 요구되는 권한, 인증, 접근 경로 등을 통하여 성공적으로 익스플로잇할 수 있는 가능성을 말한다.

외부 통제 효과(external control effective- ness)는 공격자가 취약점을 포함하고 있는 코드나 기능에 접근할 수 없도록 애플리케이션의 외부에서의 통제 능력을 말한다. 없음. 제한적, 간접적, 최선(best-available), 완전함 등으로 측정된다.

개선 노력(remediation effort)은 취약점으로 인한 위험이 더 이상 노출되지 않도록 요구되는 개선의 노력을 측정하며, 광범위함, 적당함, 제한적 등으로 측정된다. 하지만 이 요소는 취약점으로 인한 위험에 직접적인 영향을 주는 것이 아니고, 실제 웹 애플리케이션의 고객의 경우는 이 요소에 대해 직접적인 연관이 없다. 애플리케이션의 개발자들에게는 어떤 취약점을 우선적으로 처리해야 하는지에 대해서 관심을 갖는 것은 중요하지만 실제 취약점을 개선하기 위한 계획은 이러한 취약점 점검과는 별개의 프로세스로 이루어져야 하므로 이 요소가 취약성 수치화에 기여할 수 있는 요소인지에 대한 의문이 제기되었고, 결국 이 요소는CWSS의 차기 버전에서

빠지게 될 예정이다.

유행(prevalence)은 애플리케이션 내에서 취약점이 얼마나 자주 발견되는지에 대한 것이다. 이는 주로 CWE/SANS Top 25와 같은 취약점 목록을 만들 때 사용되며, 자동화된 취약점 점검 툴에서 발견된 개별적인 취약점에 대해서 적용할 수 없는 요소이다. 광범위함, 높음, 일반적, 제한적 등으로 측정할 수 있으며, 정량화된 측정도 가능하다.

3.4 한 계

앞에서 설명한 측정 항목 중에서 발견 가능성과 개선 노력과 같은 일부 항목은 취약성 수치화에 있어 적절한지에 대한 논의가 있었고 결국 CWSS의 차기 버전에서 빠지게 될 예정이다. 아직까지 CWSS에서 취약성을 측정하기 위한 요소들이 완전하게 논의되지 않고 있다는 것이다. 그러나 이러한 한계는 취약점을 바라보는 관점에 따라서 취약성을 측정하는 요소들이 변화할 수 있다는 점에서 상대적인 한계인 것으로 생각된다.

또한, CWSS의 각 측정 항목들을 통하여 도출된 가중치를 공식에 대입하였을 때 그 결과가 한 쪽으로 치우치는(skewed) 현상이 발생한다. CWSS 0.3에서는 100점 만점으로 수치화하였을 때 대부분 0점에서 2점 사이에서 편향적인 결과가 도출되는 등의 문제점이 있었고, 이후에 개선 노력이 있었지만 측정 요소들 간 연관성이 있는 등의 이유로 이러한 특정 점수대에 편향되는 문제점이 발생한다.

측정 요소들 간 연관성뿐만 아니라 하나의 웹 애플리케이션에서 발견된 여러 취약점들이 존재할 경우, 이러한 취약점들은 독립적일 수 있지만, 마치 사슬과 같이 연결되어 서로 상호작용할 수도 있다. 이 경우 각각의 취약점들에 대한 취약성 수치를 단순히 산술적으로 계산하는 것은 그 취약점들 간의 연관 관계를 무시하는 것이고, 이는 왜곡된 취약성 수치 도출로 이어진다.

CVSS와 마찬가지로, CWSS에서도 웹 애플리케이션에서 여러 종류, 여러 개의 취약점들이 발견되었을 때 이를 모두 고려한 전체적인 웹 애플리케이션의 취약성 수치를 도출할 수 있는 방법을 제시하지 못하고 있다. CWSS 차후 버전에서는 이러한 문제점을 반영할 예정이다.

Ⅳ. Microsoft 보안 대응센터의 취약성 등급 부여

Microsoft 보안 대응 센터에서 제안한 보안 심각성 측정 시스템(security bulletin severity rating system)[6]은 Microsoft 제품군에 대한 보안 심각성을 다음의 [표 1]과 같이 네 가지 등급 (심각함, 중요함, 중간, 낮음) 으로 나누어 제시한다. 등급을 나눈 기준은 취약점을 익스플로잇함의 가능성 정도, 취약점을 익스플로잇함으로 인한 영향을 바탕으로 나누어진다.

〔표 1〕 Microsoft 보안 대응센터의 취약성 등급

등급	정의
심각함 (Critical)	취약점을 익스플로잇함으로 인하여 사용자의 행동 없이 인터넷 웜이 전파됨
중요함 (Important)	취약점을 익스플로잇하면서 사용자의 데이터에 대한 기밀성, 무결성, 가용성 침해, 또는 처리 자원에 대한 무결성과 가용성에 대한 침해
중간 (Moderate)	악용가능성이 기본 구성, 감사, 또는 익스플로잇의 가능성의 정도 등에 의하여 완화됨
낮음 (Low)	취약점을 익스플로잇하기 어려움, 또는 이로 인한 영향은 미미함

이러한 등급을 자세히 살펴보면 세부적인 요소들에 의하여 등급이 분류된 것이 아니며, 주관적인 판단에 따라 취약성 등급이 결정될 수 있다. 해당 제품군을 구매하는 기업이나 조직, 또는 개인의 입장에서 어떠한 항목에서 이 제품이 취약한지에 대한 상세한 정보를 얻기 어렵다. 또한 취약점을 익스플로잇함으로써 발생하는 영향에 대해서는 모든 개인, 조직에 대해서 그러한 영향들이 일정하고 고정되어 있다고 봄으로써 그들이 해당 제품을 채택하여 운용하는 환경 및 업무 영향 등에 대한 고려를 하고 있지 않다.

Ⅴ. 결 론

그 동안 웹 애플리케이션에 존재하는 취약점들을 점검하는 자동화된 점검 도구들은 각자의 취약성 수치화 방법을 이용하여 취약성 측정을 해 왔다. 하지만 같은 웹 애플리케이션에 대해서 서로 다른 점검 도구들을 이용하여 도출된 취약성 수치들은 제각각 다른 결과를 나타낼 뿐만 아니라 어떤 부분에서 어떻게 취약한지에 대

한 정확한 정보를 제공해 주지 못하는 한계가 존재했다.

앞에서 살펴본 CVSS와 CWSS는 국가 차원에서 제안된 하나의 표준화된 취약성 수치화 프레임워크이며, 많은 곳에서 유용하게 쓰이고 있다. 그럼에도 불구하고 점점 기업이나 조직에서는 웹 애플리케이션 기반의 다양한 서비스를 제공하고 있고, 이러한 다양한 서비스 내에 존재하고 있는 다양한 취약점들을 고려한 종합화된 수치화 방법은 존재하지 않다. 점검 도구들을 이용하여 발견한 모든 취약점들을 개선하는 것보다는 어떤 취약점이 더 심각한 영향을 미치는지를 파악하여 우선순위를 두고 차근차근 대응해 나가는 것이 경제적으로도 효과적이다.

참고문헌

[1] Peter Mell, Karen Scarfone, Sasah Romanosky, "CVSS - A Complete Guide to the Common Vulnerability Scoring System Version 2.0", June 2007. http://www.first.org/cvss.

[2] Peter Mell, Karen Scarfone, Sasah Romanosky, "Common Vulnerability Scoring System", IEEE Security and Privacy, no. 6, vol. 4, pp. 85-59, 2006.

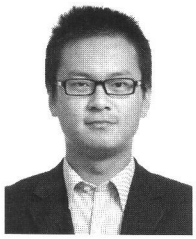
[3] Bob MArtin, Steve Christey, "Common Weakness Scoring System(CWSS)," June 2011, The MITRE Corporation, http://cwe.mitre .org/cwss.

[4] Bob Martin, Mason Brown, Alan Paller, Dennis Kirby, Steve Christey, "2011 CWE/SANS Top 25 Most Dangerous Software Errors", Sep. 2011, The MITRE Corporation, http://cwe.mitre. org/top25.

[5] The OWASP Foundation, "OWASP Top 10 - 2010, The ten most critical web application security risks", July 2010, https://www.owasp.org/index.php/Top_10.

[6] Microsoft Corporation, Microsoft Security Response Center Security Bulletin Severity Rating System, Nov. 2002, http://technet.microsoft.com/en-us/security/bulletin/rating.

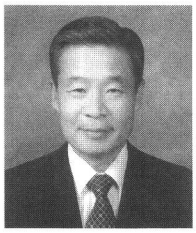
〈著者紹介〉



조 성 영 (Sungyoung Cho)
학생회원
2009년 8월: KAIST 정보통신공학과 졸업
2011년 2월~현재: KAIST 정보보호대학원 석사과정
<관심분야> 네트워크보안, 생체인증, 정보보호 평가 및 인증, 정보보호 정책

전 상 훈 (Sang-hun Jeon)
정회원
2000년 2월: 울산대학교 산업공학과 졸업
2010년 8월~현재: KAIST 사이버보안연구센터 연구개발팀장
2011년 5월~현재: 빛스캔(주) 개발팀장
<관심분야> 보안동향, 침해사고대응, 보안이슈 분석

임 채 호 (Chae-ho Lim)
종신회원
1986년: 홍익대학교 전산학과 학사
2001년: 홍익대학교 전자계산학과 박사
2006년~2009년: NHN(주) 보안실 실장, 연구센터 수석
2009년: 한국정보보호학회 부회장
2010년 8월~현재: KAIST 사이버보안연구센터 연구부소장
2011년 2월~현재: KAIST 정보보호대학원 연구교수
<관심분야> 정보보호 위험 관리, 정보보호 관리 및 정책



김 세 헌 (Sehun Kim)
종신회원
1972년: 서울대학교 물리학과 학사
1981년: 스탠포드 대학교 경영과학 박사
1982년~현재: KAIST 산업 및 시스템공학과 및 정보보호대학원 교수
1996년~1999년: 한국정보보호진흥원 이사
2003년: 한국정보보호학회 회장
2004년~2007년: 국가정보원 자문교수
2008년~2009년: 한국경영과학회 회장
2009년~현재: 방송통신위원회 인터넷 정보보호 협의회 회장
2012년~현재: 한국과학기술한림원 정회원
<관심분야> 침입탐지 및 조기경보, 보안경영 및 정책