情報保護學會論文誌
第 22 卷 第 3 號, 2012. 6

취약점의 권한 획득 정도에 따른 웹 애플리케이션 취약성 수치화 프레임워크*

조 성 영,[1†] 유 수 연,[3] 전 상 훈,[2‡] 임 채 호,[1,2] 김 세 헌[1,3]
[1]KAIST 정보보호대학원, [2]KAIST 사이버보안연구센터, [3]KAIST 산업 및 시스템공학과

A Web application vulnerability scoring framework by categorizing vulnerabilities according to privilege acquisition*

Sungyoung Cho,[1†] Suyeon Yoo,[3] Sang-hun Jeon,[2‡]
Chae-ho Lim,[1,2] Sehun Kim[1,3]
[1]KAIST Graduate School of Information Security,
[2]KAIST Cyber Security Research Center,
[3]KAIST Department of Industrial and Systems Engineering

요 약

안전한 웹 서비스를 제공하기 위하여 보안을 고려한 웹 애플리케이션의 설계와 구현이 요구되고 있다. 이에 따라 웹 애플리케이션의 취약성을 수치화할 수 있는 여러 가지 프레임워크들이 제시되고 있지만, 이러한 프레임워크에 의하여 도출된 수치는 누적 방식에 의하여 계산되기 때문에 취약점의 심각성을 제대로 분류할 수 없다는 문제점이 존재한다. 본 연구에서는 웹 애플리케이션에서 발생할 수 있는 취약점을 권한 획득 가능성에 따라 등급을 나누고 수치화함으로써 취약점에 대하여 우선순위를 둘 수 있다. 또한 개별 웹 애플리케이션뿐 아니라 한 조직에서 제공하는 여러 웹 애플리케이션에 대한 취약성을 수치화함으로써 어느 웹 애플리케이션이 가장 취약하며 우선적으로 처리해야 하는지 판단할 수 있다. 실제 크롤링 기반 웹 스캐너를 통하여 발견된 취약점들에 대하여 우리가 제안한 프레임워크를 적용하여 등급을 나누고 수치화함으로써 취약점의 권한 획득 가능성에 따른 분류의 중요성을 보이고 있다.

ABSTRACT

It is required to design and implement secure web applications to provide safe web services. For this reason, there are several scoring frameworks to measure vulnerabilities in web applications. However, these frameworks do not classify according to seriousness of vulnerability because these frameworks simply accumulate score of individual factors in a vulnerability. We rate and score vulnerabilities according to probability of privilege acquisition so that we can prioritize vulnerabilities found in web applications. Also, our proposed framework provides a method to score all web applications provided by an organization so that which web applications is the worst secure and should be treated first. Our scoring framework is applied to the data which lists vulnerabilities in web applications found by a web scanner based on crawling, and we show the importance of categorizing vulnerabilities according to privilege acquisition.

Keywords: web vulnerability, scoring system, privilege acquisition, rating

접수일(2012년 2월 3일), 수정일(2012년 4월 30일), 게재확정일(2012년 5월 4일)

* 이 연구는 대한민국 지식경제부 정보통신진흥기금으로 수행되었으며, 정보통신산업진흥원(NIPA)의 관리로 진행된 사이버보안연구센터 지원사업임. (과제코드 NIPA-H0701-12-1001)

† 주저자, sungyoung.cho@kaist.ac.kr
‡ 교신저자, p4ssion@gmail.com

I. 서 론

최근 국내에서 7.7 DDoS 공격과 3.4 DDoS 공격, 국내 금융회사와 온라인 쇼핑몰의 개인정보 유출 사고와 전산시스템 마비 사고 등과 같은 크고 작은 보안 사고가 발생하고 있다. 이러한 보안 사고의 원인 중 상당수는 웹 애플리케이션에서 발견되는 취약점(vulnerability)을 이용한 공격이 주를 이루고 있다. 미국 카네기멜론 대학교 소프트웨어 공학 연구소에서 조사한 바에 따르면, 소프트웨어의 보안 취약점의 70% 정도가 설계 과정에서 발생하는 오류로 인하여 발생하고 있다 [1]. 따라서 보안 사고를 예방하기 위한 방법으로 소프트웨어 개발 주기(software development life cycle)의 설계(design) 및 구현(implementation) 단계에서부터 보안을 고려하여야 한다. 개발자들은 상용화된 스캐너 도구를 이용하여 웹 애플리케이션의 취약점을 진단하고, 일부 도구는 취약성(degree)을 수치화하고 있지만 구체적이고 표준화된 기준이나 절차가 마련되어 있지 않다. 이러한 맥락에서 CVSS (common vulnerability scoring system)와 CWSS (common weakness scoring system) 등은 표준화되고 구체적인 기준과 절차를 통하여 취약성을 수치화(scoring)할 수 있는 프레임워크(framework)이다. 이러한 프레임워크를 사용하여 도출된 취약성 수치는 보다 안전한 프로그램을 만들어야 하는 개발자뿐만 아니라, 소비자에게도 보다 안전한 프로그램을 선택할 수 있는 기회를 제공하는 점에 의의가 있다. 하지만 이러한 프레임워크들은 단지 하나의 웹 애플리케이션에 적용할 수 있으며, 웹 애플리케이션에 기반을 둔 여러 서비스들의 전반적인 취약성을 나타내는 데 한계가 있다.

우리는 CWSS를 보완한 취약성 수치화 프레임워크를 제안하고자 한다. 이 프레임워크에서는 CWSS를 보완하여 웹 애플리케이션에서 발견된 취약점을 권한 획득의 가능성에 따라 등급을 나누고 분류함으로써 우선순위에 따른 등급을 도출한다. 이렇게 도출된 등급을 통하여 우선적으로 처리해야 할 취약점이 무엇인지, 우선적으로 대응해야 하는 서비스가 무엇인지 파악할 수 있어 비용과 인력의 투입을 조정함으로써 위협 요소에 대해 선별적으로 대응할 수 있다.

이 논문은 다음과 같이 구성된다. 2장에서는 현재 웹 애플리케이션을 통한 공격 및 피해 실태를 살펴봄으로써 웹 애플리케이션에 존재하는 취약점을 점검해야 할 필요성을 제기하고, 3장에서 CVSS, CWSS, CWRAF, 그리고 다른 여러 취약성 측정 프레임워크를 알아본다. 4장에서는 3장에서 살펴본 기존의 취약성 측정 프레임워크가 가지고 있는 문제점과 우리가 제안하는 프레임워크를 제시하고, 5장에서는 4장에서 언급한 프레임워크를 실제 취약점 점검 결과에 적용한 결과를 나타낸다. 6장에서는 이러한 연구의 결과를 정리한다.

II. 웹 애플리케이션을 통한 공격 및 피해 실태

2.1 통계 자료

한국인터넷진흥원(KISA) 인터넷침해사고대응센터(krCERT)에서 분석한 인터넷 침해사고 동향에 따르면2011년 11월을 기준으로 인터넷을 통한 침해사고는 10,599건이 발생하였다 [2]. [표 1]을 보면, 스팸릴레이가 3,450건, 단순한 침입 시도가 2,734건, 홈페이지 변조가 1,640건의 순으로 나타났다. 이는 KISA에서 접수·처리한 침해사고만을 나타낸 수치임을 고려한다면, 실제 발생한 인터넷 침해사고는 이보다 더 많은 것으로 파악할 수 있다.

특히, 홈페이지 변조로 인한 침해사고는 인터넷 침해사고의 상당한 비중(2010년 18.7%, 2011년 15.5%)을 차지하고 있다. 피해 시스템 수는 IP의 수를 기준으로 나타난 것이며 하나의 IP를 갖는 단일 시스템에 많게는 수십 개에서 수백 개에 이르는 홈페이지가 운영되고 있는 것을 감안한다면 피해 홈페이지 수는 피해 시스템의 수보다 많다고 볼 수 있다 [2].

홈페이지 변조 등을 통한 웹 공격은 홈페이지 등의 웹 애플리케이션의 취약점으로 인하여 발생한다. [표 2]는 웹 애플리케이션의 취약점 분포를 나타내는데 [3], 40.9%(입력부 22.4%, 출력부 18.5%)의 비율로 적절하지 못한 입출력부 처리에서 취약점이 존재하며, 웹 애플리케이션에 대한 공격은 이러한 취약점을 이용하여 이루어지게 된다.

[표 1] 인터넷 침해사고 현황[2]

구 분	2010년	2011년 (11월까지)
스팸릴레이	5,216	3,450
피싱 경유지	891	338
단순침입시도	4,126	2,734
기타 해킹	3,019	2,437
홈페이지 변조	3,043	1,640
합계	16,295	10,599

〔표 2〕 웹 애플리케이션의 취약점 분포〔3, 재구성〕

취약점 항목	비 율
부적절한 입력값 처리	22.4%
부적절한 출력값 처리	18.5%
충분하지 않은 안티-자동화	15.2%
알려지지 않은 취약점	14.5%
충분하지 않은 인증	9.7%
애플리케이션의 잘못된 구성	6%
충분하지 않은 승인	5%
기타 취약점	8.7%
합 계	100%

홈페이지 변조를 포함한 인터넷 침해사고의 대부분은 〔표 3〕〔3〕에서 보이는 것처럼 DoS(32%)와 SQL 인젝션(21%), 크로스 사이트 스크립트(9%) 등에 의하여 발생된다. 이러한 SQL 인젝션이나 크로스 사이트 스크립트 공격방식은 홈페이지를 포함한 웹 애플리케이션의 취약점을 이용하여 이루어진다. 또한 이러한 웹 공격을 통하여 〔표 4〕〔3〕에서 보이는 것처럼 시스템 장애(33%), 홈페이지 변조(15%), 정보유출(13%), 악성코드 유포(9%) 등이 발생하는 것으로 분석되고 있다. 최근, 공격자는 악성코드를 다운로드할 수 있는 경로를 삽입하는 등의 방식으로 웹 애플리케이션을 변조하고 여기에 접속하는 컴퓨터가 악성코드에 감염되도록 함으로써, 개인정보를 탈취하거나 좀비PC로 만드는 등 공격자의 의도에 따라 컴퓨터를 통제할 수 있다. 이와 같이 웹 애플리케이션을 통한 공격으로 인하여 여러 가지 피해가 복합적으로 발생하게 된다.

2.2 웹 애플리케이션의 취약점으로 인한 보안사고 사례

웹 애플리케이션의 취약점으로 인한 보안사고의 한 예로 현대캐피탈 해킹사고를 들 수 있다. 지난 2011

〔표 3〕 웹을 통한 공격의 유형 분석〔3, 재구성〕

공격 유형	비 율
서비스 거부 (DoS)	32%
SQL 인젝션	21%
크로스 사이트 스크립트 (XSS)	9%
전수공격 (Brute-force)	4%
크로스 사이트 요청 위조 (CSRF)	4%
알려진 취약점 공격	4%
알려지지 않은 공격	10%
기타 공격 유형	16%
합 계	100%

〔표 4〕 웹을 통한 공격의 피해 발생 분포〔3〕

공격 피해 유형	비 율
시스템 장애	33%
홈페이지 변조	15%
정보 유출	13%
악성코드 전파	9%
허위정보 공표	9%
웜(worm)	5%
금전적 손실	4%
기타 피해	12%
합 계	100%

년 4월 발생한 현대캐피탈 해킹 사고는 웹 사이트의 취약점을 이용한 공격으로 인하여 175만여 명의 계좌 비밀번호와 신용등급을 포함한 개인정보가 유출된 사고이다〔4〕. 현대캐피탈과 같은 금융기관이나 대기업들은 데이터베이스(DB) 서버에 대한 보안을 강화하지만, 상대적으로 웹 서버에 대한 보안은 이용자들의 불편을 감안하여 상대적으로 보안에 취약한 면들이 존재한다. 따라서 공격자들이 이와 같은 점을 이용하여 SQL 인젝션이나 블라인드 SQL 인젝션 등과 같은 공격 방법을 이용하여 고객의 정보를 유출한 것으로 추측되고 있다〔5〕. 따라서 웹 애플리케이션에 대해 개발 및 구현 단계에서 보안을 강화한 프로그래밍이 이루어진다면 이와 같은 사고를 방지할 수 있을 것이다. 이러한 맥락에서 안전한 웹 애플리케이션의 구현을 위하여 개별 웹 애플리케이션 기반 서비스에 대해서 CVSS와 CWSS와 같이 취약성을 수치화할 수 있는 다양한 프레임워크가 제안되고, 폭넓게 활용되고 있다. 하지만 여러 서비스에 대해 다양한 취약점들이 발견될 때, 인력과 비용이 한정되어 있다면 어떤 서비스를 우선 보완해야 하는지에 대한 결정이 요구될 수밖에 없다.

III. 관련 연구

웹 애플리케이션의 취약점을 이용한 공격으로 다양한 피해가 발생한다는 사실을 2장에서 확인하였다. 이에 따라 개발자들은 안전한 웹 애플리케이션을 개발하기 위하여 노력하지만, 그럼에도 불구하고 웹 애플리케이션에서 발생할 수 있는 여러 취약점들은 항상 존재할 수밖에 없다. 웹 애플리케이션의 취약점을 진단하는 웹 스캐너(web scanner)와 같은 진단 도구가 개발되고 있고, 실제로 상용화된 제품도 존재한다. 하지만 이러한 개별적인 진단 도구에서 취약성을 수치화

할 수 있는 구체적이고 표준화된 기준과 절차가 마련되어 있지 않은 것이 현실이다. 따라서 CVSS와 CWSS와 같이 취약성을 수치화함에 있어 구체적이고 표준화된 기준과 절차가 제안되어 많은 곳에서 사용되고 있다.

3.1 CVSS (Common Vulnerability Scoring System)

CVSS [6] [7]는 미국 표준기술연구소(NIST: National Institute for Standards and Technology)의 지원을 받아 FIRST(Forum of Incident Response and Security Teams)에서 관리하고 있으며, 소프트웨어와 하드웨어 플랫폼에서 발견된 취약점에 대해 취약성을 수치화할 수 있는 방법을 제시한다.

CVSS에서 취약성 측정을 위한 구성 요소는 크게 세 가지 영역으로 분류될 수 있다. 기본 메트릭(base metric group)은 시간과 사용자 환경에 구애받지 않는 취약점의 본질적인 특성을 나타내고, 시간적 메트릭(temporal metric group)은 시간에 따라 변화하는 취약점의 특성을 나타내며, 환경 메트릭(environmental metric group)은 특정 사용자 환경에 고유한 취약점의 특성을 나타낸다. 보다 자세한 항목은 [표 5]에 요약되어 있다.

CVSS에서 기본 메트릭에 해당하는 각 요소들에 대해 적절히 수치화되면, 기본 메트릭의 수치가 0점에서 10점 사이로 계산되며, 이는 시간적 메트릭과 환경 메트릭에 단계적으로 반영되어 최종 수치가 도출된다. 시간적 메트릭과 환경 메트릭은 CVSS를 사용하는 목적과 환경에 따라 선택적이므로 반드시 고려될 필요는 없다.

[표 5] CVSS의 취약성 측정 항목 분류

Metric Group	Factors
Base	• Access Vector • Access Complexity • Authentication • Confidentiality Impact • Integrity Impact • Availability Impact
Temporal	• Exploitability • Remediation Level • Report Confidence
Environmental	• Collateral Damage Potential • Target Distribution • Confidentiality Requirements • Integrity Requirements • Availability Requirements

CVSS를 이용하여 취약성을 수치화함에 있어, 개별 취약점을 대상으로 측정하며, 대상 호스트에 미치는 직접적인 영향만을 고려한다. 개별 취약점에서 [표 5]에서 언급된 요소들은 기밀성(confidentiality), 무결성(integrity), 그리고 가용성(availability)의 관점에서 측정된다. 이를 통하여 취약점의 어느 부분이 취약성이 강한지 판단할 수 있다.

하지만 CVSS는 이미 알려진 개별 취약점에 대하여 취약성을 측정할 수 있다는 점에서 하나의 웹 애플리케이션에서 발견된 모든 취약점들에 대해 이 취약점들을 종합한 취약성을 측정하고 수치화할 수 없다. 즉, 웹 애플리케이션의 취약점을 점검하기 위해 자동화된 점검 도구를 이용하여 코드를 검사할 경우, 많게는 수천 가지의 취약점이 발견될 수 있지만 CVSS는 발견된 모든 취약점을 고려한 하나의 웹 애플리케이션의 전반적인 취약성을 수치화할 수 없다. 또한, 한 기업이나 조직에서 운영하고 있는 모든 웹 애플리케이션이 가지고 있는 취약성을 비교·평가할 수 없다는 한계가 존재한다.

3.2 CWSS (Common Weakness Scoring System)

CWSS [8]는 미국 국토안보부(Department of Homeland Security) 사이버보안부서(National Cyber Security Division)의 지원을 받아 보안 실무자들이 협력하여 만들고 MITRE 사가 관리하고 있는 웹 애플리케이션 취약성 수치화 프레임워크이다. CVSS는 하나의 웹 애플리케이션에서 발견된 개별 취약점에 대한 취약성을 수치화하는 것이라면, CWSS는 CVSS처럼 하나의 웹 애플리케이션에서 발견된 개별 취약점에 대한 취약성을 수치화할 뿐만 아니라, 서로 다른 여러 종류의 취약점들에 대해 수치화된 취약성을 비교·평가하는 데 유용하게 사용되고 있다.

CWSS에서 언급하고 있는 취약성 수치화 방법은 세 가지이다. 타깃(targeted) 방법은 하나의 웹 애플리케이션에서 발견된 개별 취약점에 대하여 취약성을 수치화하는 방법이다. 이는 기존의 CVSS와 같지만, CVSS와 CWSS에서 취약성을 수치화하기 위하여 사

용되는 항목은 조금씩 다르다. 일반적(generalized) 방법은 웹 애플리케이션에 무관하게 취약점들의 취약성을 수치화하고 이를 비교하여 우선순위를 마련하기 위하여 사용된다. CWE (Common Weakness Enumeration) / SANS Top 25 [9]와 OWASP(The Open Web Application Security Project) Top 10 [10] 등과 같이 가장 위험한 취약점의 목록을 정렬할 때 일반적 방법을 통하여 취약성을 수치화한다. 집계된(aggregated) 방법은 한 소프트웨어에서 여러 취약점들이 발견되었을 때 그 취약성을 수치화하는 방법이다.

CWSS는 CVSS와 비슷하게 세 가지 영역으로 나누어 분석한다. 기본 발견 영역(basic finding group)은 취약점에 내재된 위험, 취약점의 발견에 대한 신뢰도, 통제 강도 등을 포함하며, 공격 표면 영역(attack surface group)은 공격자가 취약점을 이용하여 공격하기 위하여 반드시 통과해야 하는 장벽(barrier)을 말한다. 한편 환경 영역(environmental group)은 업무 영향, 악용 가능성, 그리고 외부 통제의 존재와 같은 다양한 요소를 포함한다. 보다 자세한 항목은 [표 6]으로 요약될 수 있다. CWSS에서는 [표 6]에 나타낸 각 요소들에 대한 가중치(weighting factor)들을 제시하고 있으며, 취약성을 수치화할 때 각 요소들에 대하여 적절한 가중치를 선택하여 주어진 공식에 대입한 후 계산함으로써 수치가 산출된다.

[표 6] CWSS의 취약성 측정 항목 분류

Metric Group	Factors
Base Finding	• Technical Impact • Acquired Privilege • Acquired Privilege Layer • Internal Control Effectiveness • Finding Confidence
Attack Surface	• Required Privilege • Required Privilege Layer • Access Vector • Authentication Strength • Authentication Instance • Level of Interaction • Deployment Scope
Environment	• Business Impact • Likelihood of Discovery • Likelihood of Exploit • External Control Effectiveness • Remediation Effort • Prevalence

CWSS는 몇 가지 한계가 존재한다. 첫 번째, CVSS가 내재된 조정 장치가 있어 요소들 간의 영향으로 인하여 취약성 수치가 크게 변동하지 않는 반면, CWSS는 취약성을 수치화하기 위해 측정되는 요소들은 독립적이지 않고 서로 영향을 받는 부분이 존재한다.

두 번째, 하나의 웹 애플리케이션 내에서 발견된 취약점들 간에 관계와 조합(chain and composites)이 있을 경우 이에 대한 취약성을 수치화할 때 이중계산(double-counted)의 문제가 발생할 수 있거나, 여러 취약점들의 관계와 조합이 개별 취약점들의 단순한 합보다 큰 영향을 발생할 수 있음에도 불구하고 이를 반영한 취약성 수치화 방법에 대하여 제시하지 못하고 있다. 또한, 이렇게 한 웹 애플리케이션에서 발견된 여러 취약점들에 대해 이를 모두 반영한 취약성 수치화 방법을 CWSS에서 명확하게 제공하고 있지 않다. 이 부분에 대해서 CWSS에서는 차후 반영할 예정이라고 한다 [8].

마지막으로, CWSS가 공격 기술의 관점에서 취약점들을 평가하고 취약성을 수치화하고 있어, 한 기업이나 조직에서 제공하고 있는 웹 애플리케이션 기반의 모든 서비스들에 대해 취약성을 수치화하고 문제를 보완하기 위한 우선순위가 필요한 서비스를 결정하는 방법이 제시되어 있지 않다. 웹 애플리케이션에서 발견된 취약점을 이용한 공격으로 인하여 발생할 수 있는 피해를 입는 기업이나 조직의 입장에서 취약점을 평가하고 취약성을 수치화하는 방법이 필요하다. 이러한 방법은 웹 애플리케이션 기반의 서비스를 제공하고 있는 기업이나 조직에서 어떤 서비스, 그리고 한 서비스 내에서 어떤 부분이 보안 관점에서 우선적으로 처리해야 하는지를 파악하는 데 도움을 줄 것이다.

3.3 CWRAF (Commom Weakness Risk Analysis Framework)

CWRAF [11]은 CWSS와 함께 CWE 프로젝트의 한 부분으로서 업무 영역에 맞는 웹 애플리케이션 취약점을 점검하고 취약성을 수치화하는 프레임워크를 제공한다. CWRAF는 업무 영역에서 발생할 수 있는 다양한 시나리오(vignette)를 가정하여 각각의 시나리오마다 주어지는 업무 맥락가치 (BVC; business value context)와 기술적 영향 점수표 (TIS; technical impact scorecard)를 바탕으로 취약성

을 측정하여 수치화하는데 도움을 준다.

BVC는 시나리오별로 웹 애플리케이션에서 발견된 취약점이 공격자에 의해서 악용될 경우 발생할 수 있는 잠재적인 업무상 위험을 설명하고 있다. TIS는 취약점을 통한 공격이 이루어졌을 때 발생하는 저단계(low-level) 효과를 설명하고 시나리오에 의해서 정의된 업무 영역에 미치는 영향에 대하여 기술하고 있다. 이를 바탕으로 CWRAF는 시나리오에 맞는 적절한 입력값(input)을 CWSS에 제공한다.

TIS에서 크게 분류하고 있는 잠재적인 기술적 영향은 데이터 수정, 데이터 읽기, 서비스 거부(신뢰할 수 없는 실행, 리소스 점유), 승인되지 않은 코드 및 명령의 실행, 권한 획득과 신원 가정, 보호 기법 우회 그리고 활동 숨김 등이 있다.

이러한 기술적 영향이 미치는 계층(layer)은 시스템 계층, 응용프로그램 계층, 네트워크 계층, 엔터프라이즈 계층으로 나누어 분석한다. 기술적 영향과 기술적 영향이 미치는 계층은 업무 영역과 시나리오마다 부여된 수치가 다르며, 이를 CWSS에 반영하여 취약성을 수치화하는 데 도움을 준다. 구체적으로는 다음과 같다.

웹 애플리케이션의 소스코드를 웹 스캐너와 같은 점검 도구를 사용하여 정적 분석을 할 경우 발견되는 취약점(CWE entry)에 해당하는 common_consequence 항목을 살펴본다. common_consequence는 CWE 프로젝트의 일환으로 종합된 CWE 취약점 목록에서 각 항목마다 상세하게 설명되어 있으며, 특히 많이 발생하게 되는 취약점 25가지를 모아놓은 CWE/SANS Top 25 [9]가 매년 발표되고 있다. common_consequence를 참조하면 취약점을 이용한 공격으로 인한 기술적 영향이 설명되어 있으며, 업무 영역과 시나리오별로 기술적 영향과 계층에 따라 부여되는 점수가 다르므로 이를 참조하여 CWSS에서 취약성을 측정할 때 부분 점수에 반영할 수 있다.

그러나 웹 스캐너와 같은 자동화된 점검 도구들은 업무 영역과 시나리오를 자동적으로 인식하여 이에 맞는 취약성 수치화를 도출할 수 없다. 점검 도구를 이용한 웹 애플리케이션의 취약점 점검 이전에 업무 영역과 시나리오가 설정이 되어야 하며, 이러한 업무 영역과 시나리오는 매우 많은 경우의 수를 가지고 있어 해당 조직에서 이러한 업무 영역과 시나리오를 설정하고 BVC와 TIS를 일일이 확인해야 하는 불편함이 따른다. 웹 애플리케이션의 취약점을 통한 공격과 이로 인한 피해는 조직 또는 업무 영역에 따라 달라질 수 있지만, 일반적인 웹 애플리케이션의 취약성을 점검하고 취약성을 수치화함으로써 취약점의 우선순위를 정하고 위험(risk)을 줄일 수 있을 것이다.

3.4 다른 방법들

이 밖에도 다른 여러 가지 웹 애플리케이션의 취약성 수치화 방법이 제시되어 있다. CERT/CC [12]는 다양한 기준에 의하여 0점부터 180점까지의 범위 내에서 취약성을 수치화한다. 하지만 이러한 취약성 측정 기준들은 취약점에 대한 정보가 유용하거나 알려져 있는가와 같은 질문들로 이루어지고 있으며, 이러한 질문들은 구체적이지 못하고 추상적인 내용들로 구성되어 있어 실제 취약성을 수치화하는 데 주관적인 판단이 개입될 수밖에 없다. 또한 이러한 취약성 수치화 방법은 개별적으로 발견된 취약점에 한하여 유용할 뿐 하나 이상의 웹 애플리케이션의 취약성을 수치화하기에는 어려움이 따른다. Microsoft 보안대응 센터(MSRC)에서 제공하는 보안 심각성 측정 시스템 [13]에서는 Microsoft 제품군을 사용하는 시스템과 네트워크에 대한 보안 취약성을 측정할 수 있다.

이러한 기존의 취약성 수치화 프레임워크는 모든 개인과 조직에서 발생할 수 있는 취약점에 대한 영향이 고정적이라는 점에서, 개인과 조직별로 발생할 수 있는 취약점의 종류와 정도는 다양할 수 있는 현실에 비추어 볼 때 비현실적인 프레임워크라는 평가를 받고 있다.

IV. 취약성 수치화 프레임워크의 제안

4.1 기존 프레임워크의 문제점

이미 상용화되어 많은 기업이나 조직에서 채택·사용되고 있는 웹 스캐너와 같은 웹 애플리케이션 점검 도구들의 경우, 여기에 사용되는 취약성 수치화 프레임워크는 개별 점검 도구마다 서로 다른 수치 측정 기준과 공식에 따라 취약성 수치가 산출되고 있어 구체적이고 표준화된 취약성 수치화 방법이 존재하지 않는다.

표준화되고 구체적인 기준에 따른 취약성 수치화에 대한 요구에 대응하여 제안된 CVSS와 CWSS 등의 취약성 수치화 프레임워크는 웹 애플리케이션 내에서 발견된 취약점에 대한 요소들에 부분점수를 측정한 후

적절한 공식에 의하여 수치를 산출한다. 하지만 CVSS는 하나의 웹 애플리케이션에서 발견된 개별 취약점에 대한 취약성 수치화 방법을 제시하고 있지만, 발견된 모든 취약점들을 고려하여 종합한 하나의 수치로 도출할 수 있는 방법에 대하여 제시하지 않고 않다 [2]. CWSS 또한 앞에서 이미 언급한 것처럼, 여러 종류의 취약점이 한 개 이상 발견되었을 경우 한 웹 애플리케이션 내에서 이를 반영한 종합적인 취약성 수치로 산출할 수 있는 방법에 대하여 제시하지 못하고 있다 [3].

또한, CVSS나 CWSS는 공격 기술의 관점에서 취약점들을 평가하고 있기 때문에, 한 기업이나 조직에서 제공하고 있는 웹 애플리케이션 기반의 모든 서비스들에 대해 취약성을 수치화하고 이를 비교하는 방법이 제시되어 있지 않다. CVSS나 CWSS에 의하여 도출된 취약성 수치는 전반적인 수치에 불과하기 때문에, 공격으로 인한 피해자인 서비스 제공 기업이나 조직의 입장에서 어떤 피해를 입을 수 있는지 파악하기 어렵다. 게다가 CVSS나 CWSS에서는 모든 위협을 동등하게 간주하고 있으므로 기업이나 조직의 입장에서는 단순히 발견된 취약점의 개수가 많은 웹 애플리케이션에 대해 우선적으로 조치해야 한다고 생각하기 쉽다. 하지만 실제로는 취약점들마다 존재하고 있는 위협의 심각성은 다르며, 이러한 위협의 심각성에 따라 취약점들을 분류하고 이에 따라 웹 애플리케이션에서 발견된 취약점들에 대해서 대응해야 하는 우선순위 또한 달라져야 한다. 이러한 우선순위에 기반하여 웹 애플리케이션 기반의 서비스를 제공하고 있는 기업이나 조직에서 보안 관점에서 바라보았을 때 어떤 서비스가 우선적으로 처리되어야 하는지, 그리고 그 서비스 내에서 어떤 부분이 우선적으로 처리되어야 하는지를 파악할 수 있어야 한다.

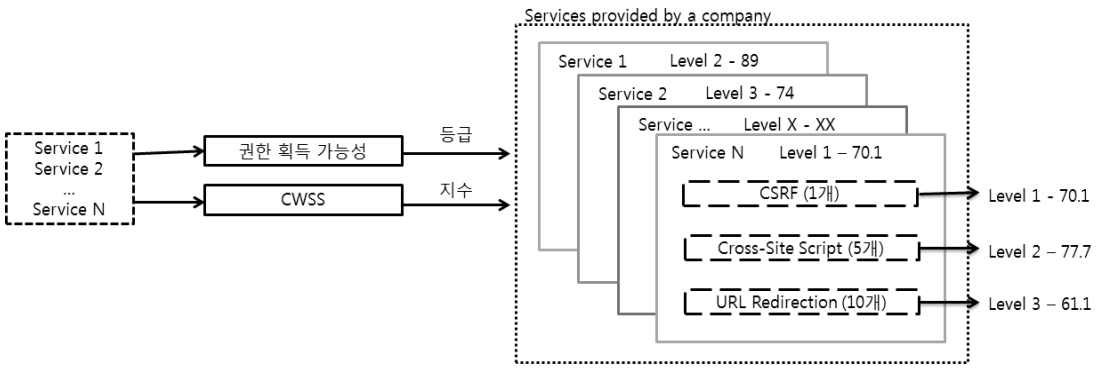
〔표 7〕 권한 획득 여부에 따른 등급의 분류

등급	시스템 정보 획득	개인정보 획득	시스템 통제
1 (High)	○	○	○
2 (Medium)	○	○	×
3 (Low)	○	×	×

4.2 프레임워크 구조

기존의 취약성 측정 프레임워크는 공격 기술의 관점에서 바라본 것이다. CVSS와 CWSS의 평가 항목(〔표 5〕, 〔표 6〕)에 따르면 웹 애플리케이션에서 발견된 취약점에 대해 기술적 영향(technical impact)이나 접근 경로(access vector) 등과 같은 기술적 항목을 통하여 취약성을 수치화한다. 이러한 기술적인 항목은 개발자 또는 공격자의 관점에서 접근한 것이다. 하지만 피해자, 즉 웹 애플리케이션 기반의 서비스를 제공하고 있는 기업이나 조직의 관점에서 취약성을 수치화할 필요가 있다. 취약점을 이용한 공격으로 인한 피해가 어느 정도인지 수치를 통하여 파악할 수 있도록 한다면, 기업이나 조직의 입장에서는 서비스의 어떤 부분이, 그리고 어느 서비스가 보안 관점에서 우선적으로 처리되어야 하는지 판단할 수 있다.

웹 애플리케이션에서 피해의 정도는 "권한 획득(privilege)"과 깊은 연관이 있다. 취약점을 이용한 공격을 통하여 권한을 획득할 수 있는 정도에 따라 공격자가 활용할 수 있는 범위는 다양하다. 취약점을 통



〔그림 1〕 연구에서 제안하는 취약성 수치화 프레임워크의 구조와 예시

하여 시스템의 정보 및 웹 애플리케이션에서 보유하고 있는 개인정보, 그리고 웹 애플리케이션이 동작하는 시스템에 대한 권한 획득이 이루어진다면 공격자는 그의 의도에 따라 시스템을 통제할 수 있으며, 개인정보를 이용하여 금전적 이득을 얻을 수 있다(Level 1, high). 하지만 어떤 취약점을 통하여 시스템에 대한 통제를 할 수 없다면 공격자는 단지 개인정보의 획득을 통하여 금전적인 이득을 볼 수 있다(Level 2, medium). 만약 어떤 취약점이 단지 시스템에 대한 정보만을 확인할 수 있다면 공격자는 제한적인 정보만을 획득할 수 있으며, 앞에서 언급한 다른 취약점들보다는 덜 심각하다고 볼 수 있다(Level 3, low). 이를 〔표 7〕로 정리하였다. 취약점이 웹 애플리케이션 내에서 발견되었을 때, 취약점을 통한 권한 획득의 여부에 따라 Level 1, Level 2, 또는 Level 3 등의 세 가지 등급(level)을 부여할 수 있다. 이렇게 등급을 정의한 후, 기존의 CWSS를 이용하여 개별 취약점에 대한 취약성 수치를 산출할 수 있다. 따라서 제안된 프레임워크에 따라 도출되는 취약성은 "Level 1 - 93.3"과 같이 표현될 수 있다.

이러한 등급 분류에 따라 우리가 제안하는 프레임워크의 구조는 〔그림 1〕과 같다. 자동화된 웹 스캐너 도구를 이용하여 웹 애플리케이션을 점검하여 취약점을 발견하면, 이를 이용하여 취약성을 판단할 수 있다. 이러한 프레임워크에 의하여 서비스에 대한 취약성이 등급과 수치로써 산출되면 프로그램 개발자 및 기업의 경영진은 어느 취약점이 우선적으로 수정·보완되어야 하는지 판단할 수 있다. 즉, 〔그림 1〕과 같이 한 서비스 내에서 취약점들이 발견되었을 때, 등급이 높을수록 권한 획득의 정도가 크므로 입을 수 있는 피해가 크다고 볼 수 있으며, 다른 취약점보다도 우선적으로 처리되어 수정·보완되어야 한다. 〔그림 1〕의 예에서 Service N에서 CSRF(cross-site request forgery) 취약점이 1개, 크로스 사이트 스크립트 취약점이 5개, 그리고 URL 리디렉션 취약점이 10개 발견되었다고 한다면, 우리가 제안한 수치화 프레임워크에 의하면 각각 "Level 1 - 70.1", "Level 2 - 77.7", "Level 3 - 61.1"로 취약성 수치가 도출될 수 있다. 기존의 취약성 수치화 프레임워크에 의하면, 가장 많이 발견된 취약점인 URL 리디렉션 취약점을 먼저 처리해야 하지만, 그 서비스에서 발견된 CSRF 취약점을 이용한 공격에 의하여 시스템의 권한 획득이 이루어지게 되므로 CSRF 취약점이 Level 1로 파악되고, 이렇게 도출된 수치를 바탕으로 우선적으로 대응·처리해야 할 취약점이 CSRF임을 확인할 수 있다.

이렇게 개별 서비스에서 발견된 취약점에 대한 등급과 수치가 도출할 수 있다. 이 때 취약성 등급은 발견된 취약점들이 가진 등급 중 가장 높은 등급으로 결정되고, 결정된 등급에 속하는 개별 취약점들의 취약성 수치 중 최대값이 이 서비스의 취약성 수치가 된다. 이는 보안 관점에서 공격자들은 시스템이나 서비스 내에 존재하는 많은 취약점 중 가장 취약한 부분을 찾아 그 부분을 집중 공격함으로써 그들의 목표를 달성하려 하기 때문이다. 개별 서비스마다 위와 같은 취약성 지수가 산출된다면 그 서비스들을 제공하고 있는 기업에서는 어떤 서비스가 가장 취약한지 판단할 수 있게 되고 그 서비스 중에서 가장 취약한 부분을 찾아 우선적으로 조치를 취하게 함으로써 발생할 수 있는 보안 사고를 신속하고 유연하게 대처할 수 있다. 〔그림 1〕의 예에서는 가장 취약하다고 평가된 서비스 N을 우선적으로 처리하여야 하며, 서비스 1과 서비스 2 순으로 처리하면 된다.

4.3 CWE/SANS Top 25에의 적용

우리가 제안한 프레임워크에 대하여 가장 위험한 소프트웨어 에러를 정리하여 목록화한 CWE/SANS Top 25 [9]를 〔표 9〕에 적용·비교하여 정리하였다.

적용한 결과 CWE/SANS Top 25에서 가장 위험하다고 여겨지는 취약점들의 상당수는 권한 획득이 이루어지는 것들이기 때문에 우리의 프레임워크에서도 위험한 것으로 나타났으나, 일부 취약점들은 순위에 변동이 발생함을 알 수 있었다. 예를 들어, 크로스 사이트 스크립팅(XSS)은 CWE/SANS Top 25에서 4위로 결정될 정도로 매우 취약한 것으로 나타났지만, XSS 공격만으로는 권한 획득이 이루어지지 못하기 때문에 Level 2로 나타났으며, 순위도 19위로 밀려난 것을 알 수 있다. 즉 CWE/SANS Top 25에서 경로 순회(path traversal) 취약점이 XSS 취약점보다 덜 취약한 것으로 분석됨에도, 우리의 프레임워크 기준인 "권한 획득" 기준에서 보면 심각한 취약성을 가지고 있다는 것을 볼 수 있다.

V. 프레임워크의 적용 및 분석

5.1 식의 사용 예

우리가 제안한 프레임워크를 실제에 적용하고 이를

〔표 8〕 취약점 분석 데이터 개요

취약점 검사	2011년 1월	
취약점 분석 소요 시간	456.36분	
취약점 검사 도메인	취약점 존재함	23개
	취약점 존재하지 않음	91개
	연결 안됨	26개
	외부 연결 도메인	6개
	총 검사 도메인	146개

분석하기 위하여, 지난 2011년 1월에 국내 모 대학의 도메인을 포함하고 있는 웹 사이트들을 대학 사이트의 메인 웹 페이지부터 단계적으로 크롤링(crawing)하는 방식으로 웹 사이트들이 가지고 있는 취약점을 분석하였다. 〔표 8〕은 이 데이터들의 간략한 개요를 나타낸다.

취약점 검사 기준은 〔표 10〕에 나타내었다. 모든 취약점을 검사하지 않는 이유는 실제 보안 웹코딩에 관련된 부분은 CWE에서 제시하는 보안 취약점 중 극히 일부분이며, 웹 애플리케이션의 설계 및 구현 단계에서 발생하는 코딩 오류가 아닌 보안 구성, 통신, 로직과 같이 보안 코딩 이외의 요소로 인한 웹 보안의 취약점이 주류를 이루고 있기 때문이다. 한편, 등급에 따라 취약점을 분류한 후에 적용한 취약성 지수는 CWE/SANS Top 25 2011 〔9〕을 이용하였다.

5.2 적용 및 분석 결과

웹사이트들의 취약점을 분석한 데이터를 이용하여, 〔표 9〕와 본 논문에서 제안한 프레임워크에 적용한 결과를 〔표 12〕과 같이 정리하였다. 여기서 사이트 취약성 지수는 가장 높은 등급을 차지한 개별 취약점들의 취약성 수치 중 최대값으로 결정된다.

사이트 4에서 발견된 취약점은 1등급 2개, 2등급 1개이며 제안된 프레임워크의 기준과 〔표 9〕에 따라 사이트의 등급은 1등급, 취약성 지수는 93.8이다. 반면, 사이트 11는 2등급 19개, 3등급 2개이며 사이트의 등급은 2등급, 취약성 지수는 77.7이다. 사이트 4와 사

〔표 9〕 제안된 프레임워크를 CWE/SANS Top 25에의 적용과 비교

제안 프레임워크 적용			SANS Top 25 순위	CWE 번호	취약점 내용
순위	등급	CWSS 지수			
1	1	93.8	1	CWE-89	SQL Injection
2	1	83.3	2	CWE-78	OS Command Injection
3	1	79.0	3	CWE-120	Classic Buffer Overflow
4	1	76.9	5	CWE-306	Missing Authentication for Critical Function
5	1	76.8	6	CWE-862	Missing Authorization
6	1	75.0	7	CWE-798	Use of Hard-coded Credentials
7	1	75.0	9	CWE-434	Unrestricted Upload of File with Dangerous Type
8	1	74.0	10	CWE-807	Reliance on Untrusted Inputs in a Security Decision
9	1	73.8	11	CWE-250	Execution with Unnecessary Privileges
10	1	70.1	12	CWE-352	Cross-Site Request Forgery (CSRF)
11	1	69.3	13	CWE-22	Path Traversal
12	1	68.5	14	CWE-494	Download of Code Without Integrity Check
13	1	67.8	15	CWE-863	Incorrect Authorization
14	1	65.5	17	CWE-732	Incorrect Permission Assignment for Critical Resources
15	1	62.4	20	CWE-131	Incorrect Calculation of Buffer Size
16	1	61.5	21	CWE-307	Improper Restriction of Excessive Authentication Attempts
17	1	61.0	23	CWE-134	Uncontrolled Format String
18	1	60.3	24	CWE-190	Integer Overflow of Wraparound
19	2	77.7	4	CWE-79	Cross-Site Scripting
20	2	75.0	8	CWE-311	Missing Encryption of Sensitive Data
21	2	66.0	16	CWE-829	Inclusion of Functionality from Untrusted Control Sphere
22	2	64.1	19	CWE-327	Use of a Broken or Risky Cryptographic Algorithm
23	3	64.6	18	CWE-676	Use of Potentially Dangerous Function
24	3	61.1	22	CWE-591	URL Redirection to Untrusted Site
25	3	59.9	25	CWE-759	Use of a One-Way Hash without a Salt

〔표 10〕 웹 애플리케이션 취약점 점검 유형

취약점 유형	발생할 수 있는 피해 상황	등급
SQL 인젝션	• 데이터베이스에 저장된 레코드에 대한 변조, 누출, 삭제 • 악성코드 유포지로 활용 가능성 • 공격자에 대한 서버 권한 획득 및 내부망 침입 가능 • 웹서버 방문자에 대한 악성코드 설치 시도로 신뢰 상실	1
Blind SQL 인젝션	• SQL 인젝션 취약점과 동일한 피해 유발이 가능하나 공격시간 및 정보 획득에 시간이 다소 오래 걸림 • 데이터베이스의 정보 유출 • 공격 가능한 정보를 획득할 수 있으며 에러의 상태에 따라 SQL 인젝션 등의 공격이 유발될 수 있음	1
크로스 사이트 스크립팅 (XSS)	• 세션 하이재킹 • 피싱에 활용될 수 있으며, 관리자 권한 및 사용자 권한 유출 • 리디렉션 공격 • 사용자 정보의 유출 가능성	2
예외적 오류	• 실행 경로의 유출 가능성 • 구성 패키지 정보 노출 • 예외 처리가 미비할 경우 에러의 종류에 따라 SQL 인젝션이나 HTTP 500 에러 등의 공격이 가능함	3
의심스러운 오류	• 취약성 여부는 확인되지 않았지만 오류로 인식될 수 있는 문제점 • 오류 가능성이 있으며, 추가적인 검토가 필요한 오류 • 의심이 되는 상황에 대해 보고되나 에러가 아닌 경우도 있음	3

이트 11의 결과를 비교해보면, 사이트 11에서 발견된 총 취약점의 개수가 21개로 사이트 4에서 발견된 취약점의 총 개수보다 많지만 사이트의 등급에 더 우선순위를 두기 때문에 사이트 4의 취약점에 대한 처리가 먼저 이루어진다.

한편, 사이트 4와 사이트 7을 비교해보면, 사이트 7에서 발견된 취약점은 1등급 16개이며 사이트의 등급은 1등급, 취약성 지수는 93.8이다. 사이트 4와 사이트 7은 사이트 등급이 같은 1등급이다. 이와 같은 경우는 사이트 7의 총 취약점의 개수가 사이트 4의 총 취약점의 개수보다 많으므로 사이트 7에 우선순위가 있다. 그 이유는 4.2절에서 언급하듯이, 취약점의 개수가 많을수록 공격할 수 있는 지점이 많음을 의미하기 때문에 먼저 처리해야 한다.

〔표 11〕 CWSS와 제안 프레임워크의 결과 비교

취약점 대응의 우선순위	제안 프레임워크	CWSS
1	사이트 7	사이트 9
2	사이트 18	사이트 22
3	사이트 15	사이트 11
4	사이트 9	사이트 7
5	사이트 22	사이트 18
6	사이트 4	사이트 15
7	사이트 11	사이트 1
8	사이트 1	사이트 8
9	사이트 8	사이트 4
10	사이트 14	사이트 14

실험결과를 통해, 제안 프레임워크를 적용하면 사이트 4, 사이트 7, 사이트 11의 취약점 대응 우선순위는 사이트 7, 사이트 4, 사이트 11 순이다. 반면, CWSS에서 개별 사이트의 취약성 수치를 나타내는 방법은 명확하게 제공하고 있지 않다. 제안하는 몇 가지 방법 중 하나인 개별취약점 수치의 총합을 사이트의 취약성 수치로 사용하는 방법으로 사이트 4, 사이트 7, 사이트 11의 취약성 수치를 알아보면, 의심스러운 오류에 해당하는 취약점 수치의 크기에 따라 우선순위가 사이트 7, 사이트 11, 사이트 4 또는 사이트 11, 사이트 7, 사이트 4이다. CWSS방법으로 구한 두 결과 모두 사이트 4에 대한 취약점 수치가 나머지 두 사이트의 취약점 수치보다 낮게 계산되는 것을 확인할 수 있다. 〔표 11〕는 실험 데이터 중 임의의 10개의 데이터에 대해 제안 프레임워크와 CWSS를 각각 적용하여 우선순위를 판단한 결과를 비교한 것이다.

VI. 결 론

기존의 CWSS나 CVSS와 같은 웹 애플리케이션 취약성 수치화 프레임워크는 다양한 곳에서 채택되어

〔표 12〕 취약점 발견과 제안한 프레임워크에의 적용 결과

사이트	1 등급			2 등급		3 등급		사이트 취약성 등급	사이트 취약성 지수
	취약점	개수		취약점	개수	취약점	개수		
1	-	-		XSS	7	-	-	2	77.7
2	SQL Injection	4		XSS	20	-	-	1	93.8
3	-	-		XSS	1	-	-	2	77.7
4	SQL Injetcion	2		XSS	1	-	-	1	93.8
5	SQL Injection	2		XSS	3	-	-	1	93.8
6	SQL Injection	4		XSS	27	Suspicious Error	16	1	93.8
7	SQL Injection	12	16	-	-	-	-	1	93.8
	Blind SQL Injection	4							
8	-	-		XSS	4	-	-	2	77.7
9	SQL Injection	4		XSS	62	-	-	1	93.8
10	-	-		XSS	1	-	-	2	77.7
11	-	-		XSS	19	Suspicious Error	2	2	77.7
12	SQL Injection	4		XSS	1	-	-	1	93.8
13	-	-		XSS	1	-	-	2	77.7
14	-	-		XSS	2	-	-	2	77.7
15	SQL Injection	5		XSS	2	-	-	1	93.8
16	-	-		XSS	4	-	-	2	77.7
17	-	-		XSS	1	-	-	2	77.7
18	SQL Injection	8		XSS	2	-	-	1	93.8
19	-	-		XSS	1	-	-	2	77.7
20	SQL Injection	4		-	-	-	-	1	93.8
21	-	-		XSS	1	-	-	2	77.7
22	SQL Injection	4		XSS	60			1	93.8
23	-	-		XSS	2	-	-	2	77.7

사용되고 있지만, 웹 애플리케이션에서 발견된 개별적인 취약점에 대한 취약성을 측정하는 방법만을 제공하고 있을 뿐만 아니라 취약성이 가지고 있는 심각성을 제대로 나타내지 못하고 있는 문제점이 있다.

우리가 제안한 프레임워크는 취약점을 통하여 얻을 수 있는 권한 획득의 가능성 정도에 따라 등급을 나누고 분류할 수 있는 방법을 제공함으로써, 공격을 통하여 시스템의 권한을 획득할 수 있는 취약점을 먼저 분류하고 이러한 취약점들에 대해 우선적으로 조치할 수 있도록 한다. 단순히 발견된 취약점의 개수가 많다고 하여 그 웹 애플리케이션이 더 취약한 것이 아니라, 가장 심각한 취약점이 가지고 있는 보안 정도만큼 안전한 것이기 때문에 이러한 취약점에 대해 보다 현실적인 수치를 제공할 수 있다.

이러한 프레임워크는 웹 스캐너와 같은 웹 애플리케이션 진단 도구에 적용됨으로써 개발자는 물론이고 웹 애플리케이션을 통하여 서비스를 제공하고 있는 기업의 경영진들이 보안에 대하여 현실적인 대처를 할 수 있도록 도와 줄 수 있다. 최근 발생한 보안 사고에 대해서 현실적인 수치를 제공함으로써 최근 발생한 보안사고가 되풀이되는 것을 예방할 수 있다.

이 논문에서는 "권한 획득"을 등급을 나누는 기준으로 제시하였지만, 기업이나 조직에서 추구하는 보안 기준에 따라 다른 기준을 제시하고 이를 바탕으로 취약점을 재분류하여 취약성을 측정할 수 있을 것이다.

참고문헌

〔1〕 한국인터넷진흥원(KISA), 안전한 소프트웨어 개발 도입을 위한 보안 가이드. 2008년 12월. http://www.kisa.kr/jsp/common/libraryDown.jsp?folder=016551

〔2〕 한국인터넷진흥원(KISA) 인터넷침해대응센터(krCERT). 인터넷침해사고 동향 및 분석월보. http://www.krcert.or.kr

〔3〕 The Web Application Security Consortium, "The Web Hacking Incident Database Semiannual Report July to December 2011", Trustwave Holdings, Inc, March 2011.

〔4〕 헤럴드경제, "현대캐피탈 사태 사고 예방 소홀한 '인재', 금감원, 임직원 책임 묻기로," http://news.khan.co.kr/kh_news/khan_art_view.html?artid=201105180000035&code=920301, 2011년 5월.

〔5〕 한국일보, "현대캐피탈 해킹, 어떻게 이루어졌을까", http://news.hankooki.com/lpage/economy/201104/h20110411024033215 40.htm, 2011년 4월.

〔6〕 Forum of Incident Response and Security Teams (FORUM). Common Vulnerability Scoring System (CVSS). June 2007. http://www.first.org/cvss/

〔7〕 Mell, P., Scarfone. K. and Romanosky, S., "Common Vulnera- bility Scoring System," IEEE Security & Privacy, vol. 4, no. 6, pp. 85-89, Nov.-Dec. 2006.

〔8〕 Bob Martin, Common Weakness Scoring System (CWSS). The Mitre Corporation. June 2011. http://cwe. mitre.org/cwss

〔9〕 Bob Martin, Mason Brown, Alan Paller, and Dennis Kirby. 2011 CWE/SANS Top 25 Most Dangerous Software Errors. June 2011. http:// cwe.mitre.org/top25

〔10〕 The Open Web Application Security Project (OWASP) Top 10 - 2010. http://www. owasp.org

〔11〕 Bob Martin, Common Weakness Risk Analysis Framework (CWRAF), June 2011, http: //cwe.mitre.org/cwraf

〔12〕 United States Computer Emergency Readiness Team (US-CERT). US-CERT Vulnerability Note Field Descriptions. 2006. http://www.kb .cert.org/vuls/html/fieldhelp #metric

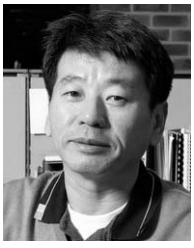
〔13〕 Microsoft Corporation, Microsoft Security Response Center Security Bulletin everity Rating System, Nov. 2002, http://technet.microsoft.com/en-us/sec urity/bulletin /rating

〈著者紹介〉

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

유 수 연 (Suyeon Yoo) 학생회원
2011년 2월: 부산대학교 산업공학과 학사 졸업
2011년 2월~현재: KAIST 산업 및 시스템공학과 석사과정
〈관심분야〉 정보보호, 네트워크 보안, 산업공학

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, 침해사고 대응, 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〉 인터넷 보안, 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