Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.4, Aug. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.4.921

보안취약점 중요도 정량 평가 체계 연구

안 준 선,[1†] 창 병 모,[2] 이 은 영[3‡]
[1]한국항공대학교, [2]숙명여자대학교, [3]동덕여자대학교

Quantitative Scoring System on the Importance of Software Vulnerabilities

Joonseon Ahn,[1†] Byeong-Mo Chang,[2] Eunyoung Lee[3‡]
[1]Korea Aerospace University, [2]Sookmyung Women's University
[3]Dongduk Women's University

요 약

본 논문에서는 소프트웨어 보안취약점의 중요도를 정량적으로 산출할 수 있는 중요도 정량 평가 체계를 제안한다. 제안된 평가 체계는 국내 소프트웨어 이용 환경을 고려한 보안취약점의 파급도, 위험도, 소프트웨어 점유율, 시스템에서의 사용 정도 등을 복합적으로 반영하여 보안취약점에 대한 심각성을 적절히 평가할 수 있는 평가 척도와 이를 기반으로 한 중요도 계산식으로 구성된다. 논문에서는 제안된 소프트웨어 보안취약점 평가 체계를 국내의 보안취약점에 시범적으로 적용하고 그 효용성을 CVSS 및 CWSS 등과 비교, 분석하였으며, 제안된 평가 체계의 활용 방안을 제시하였다.

ABSTRACT

We proposed a new scoring system on software vulnerabilities, which calculates quantitatively the severity of software vulnerabilities. The proposed scoring system consists of metrics for vulnerability severity and scoring equations; the metrics are designed to measure the severity of a software vulnerability considering the prevalence of the vulnerability, the risk level of the vulnerability, the domestic market share of the software and the frequency of the software. We applied the proposed scoring system to domestically reported software vulnerabilities, and discussed the effectiveness of the scoring system, comparing it with CVSS and CWSS. We also suggested the prospective utilization areas of the proposed scoring system.

Keywords: software security; software weakness; software vulnerability; scoring system

I. 서 론

소프트웨어 소스코드 내에 존재하는 보안약점(weakness)으로 인한 보안취약점(vulnerability)이 정보 시스템의 보안 침해 예방에 있어 중요한 문제로 간주되고 있으며, 이러한 보안취약점에 대한 효과적이면서도 신속한 대응에 대한 관심이 증가하고 있다[1]. 또한 소프트웨어를 개발하는데 있어서도 향후 보안취약점을 유발할 수 있는 코드 형태가 프로그램에 포함되는 것을 사전에 예방하기 위한 다양한 기술적 및 정책적 노력들이 이루어지고 있다[2-4].

소프트웨어의 보안취약점을 예방하기 위한 연구 및 활동의 기반 데이터를 제공하기 위하여 보안취약점과 보안약점에 대한 전반적인 정보를 축적하고 이를 효과적으로 제공하고자 하는 연구가 활발히 진행되어 지금까지 상당한 진척과 성과를 보이고 있으며,

접수일(2015년 6월 17일), 수정일(2015년 8월 3일), 게재확정일(2015년 8월 4일)
† 주저자, jsahn@kau.ac.kr
‡ 교신저자, elee@dongduk.ac.kr(Corresponding author)

대표적인 사례로 CWE(Common Weakness Enumeration), CVE(Common Vulnerability Enumeration), NVD(National Vulnerability Database) 등이 있다[5,6,7]. 또한 다양한 보안약점 중에 중요 보안약점을 선별하여 대응하고자 하는 노력으로 CWE/SANS Top 25, OWASP Top 10 등이 발표되고 있으며, 국내에서도 행정자치부 보안약점 기준이 정보 소프트웨어 개발보안 정책을 위하여 발표된 바 있다[8,9,10]. 또한 신규 보안취약점에 대한 신속한 대응을 위하여 2012년부터 한국인터넷진흥원에서는 신규 취약점 발굴 포상제를 시행하고 있다[11].

소프트웨어 개발 및 운용 과정에서 발견되는 보안약점이나 결과적으로 야기되는 보안취약점의 심각성을 판단하여 중요 약점 및 취약점에 대한 우선적인 대응을 수행하는 것이 중요하여, 이를 위하여 보안약점이나 보안취약점에 중요도를 평가하기 위한 평가척도가 필요하다. 대표적인 사례로 보안약점 평가를 위한 CWSS (Common Weakness Scoring System)와 보안취약점 평가기준인 CVSS (Common Vulnerability Scoring System)가 발표되어 사용되고 있다[12,13].

소프트웨어의 보안취약점의 영향 및 심각성은 그 자체의 본질적인 특성 뿐 아니라 사용되는 환경이나 응용 분야의 특성, 소프트웨어 점유율, 시간 및 지역에 따라 변화하는 정보 시스템 환경 등에 영향을 받게 되므로, 국내의 소프트웨어 환경 특성을 고려한 새로운 소프트웨어 보안취약점 평가 체계가 필요하며, 같은 평가 척도에 대해서도 국내 상황에 맞는 적절한 객관적 판단 기준이 설정되어야 할 것이다. 따라서 국내 환경에 적합한 정량적 및 정성적 기준을 제시하고 이에 기반 하여 소프트웨어 보안취약점의 중요도를 판별할 수 있는 소프트웨어 보안취약점 평가 체계를 마련하여 활용하는 것이 필요하다.

본 논문에서는 국내의 소프트웨어 환경을 고려한 새로운 소프트웨어 보안취약점 평가 체계를 제안하고자 한다. 본 논문은 다음과 같은 형식으로 구성되어 있다. 2장에서는 기존의 보안약점과 보안취약점 평가 방식에 대해 간단히 설명하고 이들의 한계점에 대하여 논의한다. 3장에서는 국내 소프트웨어 이용 환경을 고려한 보안취약점의 파급도, 위험도, 소프트웨어 점유율, 시스템에서의 사용 정도 등을 복합적으로 반영하여 보안취약점에 대한 심각성을 적절히 평가할 수 있는 기준과 이를 기반으로 한 평가 방법을 제안하다. 또한 4장에서는 제안된 소프트웨어 보안취약점 평가 체계를 국내의 보안취약점에 시범적으로 적용하고 그 효용성을 CVSS 및 CWSS 등과 비교, 분석하였으며, 결론에서는 제안된 보안취약점 평가 체계의 활용방안에 대하여 논의하고자 한다.

II. 관련연구

현재 가장 대표적인 보안약점 및 보안취약점 평가 체계로는 CWSS와 CVSS를 들 수 있다. 본 절에서는 소프트웨어 보안약점과 보안취약점의 정량적 평가를 위한 기반 연구로서 CWSS와 CVSS를 소개하고, 그 특성 및 개선 방향을 분석하고자 한다.

2.1 Common Weakness Scoring System

CWSS는 보안약점의 중요도를 평가하는 체계로서 총체적인 소프트웨어 보안약점 명세데이터를 구축하는 CWE 프로젝트의 일환으로 추진되었다. CWE와 CWSS의 특징은 안전한 소프트웨어의 개발과 보안 유지에 책임이 있는 당사자들인 정부, 학계, 산업체들이 모여서 만드는 커뮤니티 형태의 협업이라는 점에 있다. 현재 이 프로젝트는 미국 NCSD (National Cyber Security Division)과 미국 DHS (Department of Homeland Security)의 지원을 받아서 진행되고 있다. CWSS는 소프트웨어에 일반적으로 발생하는 다양한 약점에 대하여 제거의 우선순위를 줄 수 있는 정량적인 기준을 제시한다. 정량적인 기준을 제시하기 위한 다양한 평가 기준을 약점 자체의 심각성 (Base Finding Metric Group), 공격 측면의 심각성 (Attack Surface Matric Group), 환경적 측면의 심각성 (Environment Matric Group)으로 분류하여 그 정량적 기준과 함께 제시하고 있으며, 아울러 소프트웨어가 사용되는 도메인에 적용하여 중요성을 조정할 수 있는 방법론인 CWRAF (Common Weakness Risk Analysis Framework)를 제시하고 있다. 현재 CWSS는 버전 0.8이 2011년 6월에 발표되었으며, 최신 버전은 2014년 9월에 발표된 1.0.1이다. CWSS의 일부 평가 척도는 2011 SANS Top 25의 선정에 활용되었다.

2.2 Common Vulnerability Scoring System

CVSS는 보안약점으로부터 실제 발생한 보안취약점의 중요성을 평가하는 연구결과로 보안취약점 평가를 위한 일반적인 프레임워크를 제공한다. CVSS는 실제 사용되고 있는 소프트웨어에서 공격에 침해될 수 있는 실제적인 보안취약점을 대상으로 하기 때문에 CWSS와 차별성을 갖는다. CVSS는 보안취약점을 본질적인 기본 척도(Base Metric), 시간에 따른 척도 (Temporal Metric), 환경적인 척도 (Environmental Metric)에 따라 평가하도록 하여 보안취약점의 심각성을 다양한 관점에서 평가하기 위한 방법을 제공하고 있다. 현재 CVSS는 어느 정도 정착 단계에 이르러 버전 2 계열이 주로 활용되고 있으며, 최신 버전인 버전 3.0은 2015년 5월에 공개되었다.

2.3 기존 평가 방법론의 특징 분석

보안약점과 보안취약점에 대한 정량 평가 방법으로 CWSS, CVSS가 대표적인 평가 체계로 사용되고 있다. 그렇지만, 현재 NVD 구축에서 사용된 CVSS의 적용 사례를 살펴보면 CVSS의 다양한 척도 중에서 기본 척도만 사용하고 있고, 보안취약점의 파급도나 대상 소프트웨어의 점유율 등은 충분히 반영하고 있지 않은 것을 볼 수 있다. 이는 보안 취약점의 평가에 있어서 환경적 및 시간적 요소가 중요하나 해당 평가 척도가 특정 사용 환경에 종속적이어서 NVD에 적용하기에는 적합하지 않았음을 반증하고 있는 것으로 판단된다.

CWSS는 보안약점에 대한 평가 척도라는 점에서 보안취약점 평가에 그대로 사용하기에는 적합하지 않은 것으로 판단된다. 또한 보안 약점 평가척도에 대한 세부 평가 기준이 충분히 객관적이지 못하여, CWSS 버전 0.8이 적용된 SANS Top 25 사례에서도 CWSS의 여러 가지 척도 중에서 중요도(importance), 유행도(prevalence), 침해가능성(likelihood of exploit)만을 적용하고 있다. 그러나 CWSS는 CVSS와 비교하여 좀 더 최근에 개발된 평가 척도로서 CVSS보다 다양한 평가 척도를 제시하고 있다는 강점을 가지고 있다.

3장에서는 CVSS와 CWSS의 개별 척도에 대한 보다 구체적인 분석을 제시하고 이를 반영한 보안취약점 중요도 평가 체계를 제시한다.

III. 보안취약점 중요도 평가 체계 설정

3.1 기존 평가체계 평가척도 분석

기존의 보안약점 및 보안취약점 평가체계인 CWSS와 CVSS는 다면적인 평가 척도를 사용한 평가방법을 제시하고 있으나, 국내 보안취약점 평가에 바로 활용하기에는 부족한 점이 존재한다. 그 주요 내용은 다음과 같다.

- 취약점으로 인한 침해의 시도 또는 발생이 얼마나 빈번한가를 측정함에 있어서, CVSS에서는 대상 분포(TD)를 제시하고 있으나, 그 의미가 특정 설치 환경 내에서의 침해의 영향 범위에 국한하고 있어 소프트웨어 취약점의 전역적인 영향도를 충분히 반영하지 못하며, 특히 관련 소프트웨어의 보급 정도를 반영하지 못하고 있다. CWSS는 일반적인 약점 형태의 프로그램 출현 빈도를 평가하고 있어 취약점의 출현도 평가에 정확히 부합하지 않는다.
- 취약점으로 인한 침해의 기술적인 영향에 있어서, CVSS는 보안의 기본 3요소(기밀성, 무결성, 가용성)에 기반하고 있어서 그 구분이 세밀하지 못하고 악성 코드 수행과 같은 침해의 기술적 형태를 적절히 반영하지 못하고 있다. CWSS는 기술적 영향(TI) 척도에 대한 구체적인 기준을 제시하고 있지 않아 평가의 객관성을 확보하기 어려운 단점이 있다.
- CWSS는 일반적인 약점에 대한 중요성 척도를 제시하고 있기 때문에 특정 소프트웨어의 특성이나 용도 등을 반영한 평가 기준을 제시하지 못하고 있으며 척도에 대한 평가 기준의 객관성이 부족하다.

본 연구에서는 이러한 점을 고려하여 취약점의 중요도를 결정하기 위하여 평가하여야 할 요소를 설정하고, 각 평가 요소에 대하여 국내 상황을 고려하여 중요도를 평가할 수 있는 평가 척도를 설정하였다.

3.2 평가 요소의 선정

신규 취약점 발굴의 중요도를 평가하기 위한 평가 척도를 선정하기 위하여 먼저 중요도를 구성하는 6개의 주요 요소를 다음과 같이 도출하였다.

- 출현도: 해당 취약성이 얼마나 많은 시스템에서 발견될 수 있는지를 나타낸다.
- 시스템 중요도: 해당 취약성으로 인한 침해의 결과가 경제적 및 사회적으로 얼마나 심각한 영향을 끼치는지 여부를 판정한다.
- 기술적 영향: 해당 취약성으로 인해 해당 시스템에 발생할 수 있는 기술적인 파급의 범위 및 심각성을 판정한다.
- 공격 난이도: 공격자가 해당 취약성을 가진 시스템을 발견하여 이에 대한 공격을 시도할 경우, 실제적인 침해를 발생시키기 위한 기술적, 절차적 어려움 및 성공 가능성을 평가한다.
- 대응 난이도: 해당 취약성에 대한 대응 방법의 난이도를 평가한다. 대응 난이도는 대응기술의 발전에 따라 가변적인 척도이다.
- 발굴 수준: 신규 취약점 발굴 작업과 관련한 척도로서 취약점 발굴 작업의 난이도와 문서화의 정도 등 발굴 작업 전반에 대한 수준을 평가하며 신규 취약점 발굴의 평가에 활용할 수 있다.

설정된 평가요소에 대하여 각 요소를 평가하기 위한 평가 척도를 선정하였다. 평가 척도의 선정에 있어서 주요 고려 사항은 다음과 같다.

- 용이성: 모든 평가 요소를 포괄하되, 용이성을 위하여 과다한 척도 설정을 지양한다.
- 독립성: 평가 척도간의 중복성 또는 의존성이 발생하지 않도록 독립적인 척도를 설정한다.
- 객관성: 평가 척도별로 객관적인 기준을 제시하여 평가의 객관성을 확보한다.

3.3 평가 척도의 선정

각 평가 요소별로 CWSS와 CVSS의 평가 척도들을 참고하여 기존의 평가 척도의 기준을 객관화 하거나 신규의 평가 척도를 개발하여 각 평가 요소를 균형 있게 평가하도록 구성하였다. 다음은 평가 요소별로 선정된 평가 척도를 나타낸다.

- 출현도: 파급 범위, 대상 분포
- 시스템 중요도: 피해의 심각성
- 기술적 영향: 침해 형태
- 공격 난이도: 접근 벡터, 권한 요구도, 상호작용 정도, 침해 가능성
- 대응 난이도: 교정 난이도, 외부 제어의 효과
- 발굴 수준: 발굴 난이도, 문서 완성도

선정한 12개 평가척도에 평가 내용과 평가 기준은 다음과 같다.

1) 파급 범위

- 평가 내용: 보급률이 높고 사용자가 많은 소프트웨어에서 발견되는 취약점일수록 높은 값을 가진다. 또한 해당 소프트웨어의 특성에 따라 침해의 파급 정도가 달라지므로 소프트웨어의 종류와 보급도를 복합적으로 평가하도록 하였다.
- 등급별 기준: Widespread(W), High(H), Common(C), Limited(L), Rare(R)의 5가지 단계로 평가하며, 각 등급별 점수는 1, 0.9, 0.8, 0.6, 0.3으로 부여한다. Table 1은 각 소프트웨어 종류에 따른 보급도 단계에 대한 예시를 보여 준다.

Table 1. Prevalence classification of software

Prevalence / kind	Repre-sentative	Generlly known	Rarely Known
Operating Systems	W	W	H
Infrastructure (Router etc) Management, Server SW, System SW	W	H	C
Web Site Security SW (portal, bank etc)	W	H	L
basic application program (word processor, messenger security program etc)	H	C	L
general application	C	L	L
open source program	C	L	R
system control web page	L	L	R

2) 대상 분포

- 평가 내용: 해당 보안취약점에 의하여 침해가 가능하도록 설치되는 소프트웨어 버전 및 설치 환경의 비율을 평가한다.
- 등급별 기준 : 대상 분포 척도의 등급별 평가 기

준은 다음과 같다. 괄호 안은 해당 등급에 대한 점수값이다.

- All(1): 해당 보안취약점이 모든 플랫폼과 설정에 존재하는 경우이다.
- Moderate(0.9): 해당 보안취약점이 일반적인 플랫폼과 일반적으로 인스톨되는 설정에 존재하는 경우이다.
- Rare(0.6): 해당 보안취약점이 사용빈도가 낮은 플랫폼이나 설정에 존재하는 경우이다.
- Default(0.8): 해당 보안취약점으로 인한 침해 가능성이 해당 소프트웨어가 설치된 시스템의 특성에 따라 다양할 경우에 중간 값을 부여한다.

3) 피해의 심각성

- 평가 내용: 취약점을 이용한 공격으로 발생 가능한 비즈니스/임무의 운영에 대한 피해의 정도에 따라 평가를 수행하며, 실제적으로 침해의 피해 정도는 특정 시스템의 사용 목적 및 운용 환경에 의존적이므로 국내 평균적인 사용 환경을 고려하여 영향도를 평가한다. 또한 해당 소프트웨어가 공공의 안전 등에 영향을 미칠 경우 한 등급을 상향 조정한다.
- 등급별 기준 : 피해의 심각성 척도의 등급별 평가 기준은 다음과 같다.
  - Critical(0.7): 비즈니스/임무가 완전히 실패할 수 있다.
  - High(0.5): 비즈니스/임무의 운용이 크게 영향 받을 수 있다.
  - Medium(0.3):비즈니스/임무의 운용이 크게 영향 받을 수 있으나 정상적인 운용에 대규모의 피해는 없다.
  - Low(0.1):비즈니스/임무에 최소한의 영향이 있다.
  - None(0): 비즈니스/임무에 최소한의 영향이 없다.

4) 침해 형태

- 평가 내용: 취약점에 대한 공격이 성공할 경우 발생하는 침해 형태의 종류를 평가한다. CWE에서 제시한 기술적 영향(Technical Impacts) 항목을 기준으로 관련성을 파악하여 점수를 합산한 후, 이를 기준으로 등급을 부여한다. 항목별 침해도에 따른 점수는 Table 2와 같다.

Table 2. Scopes of technical impacts

Technical Impact	Complete	Partial	None
Modify data	2	1	0
Read data	2	1	0
DoS: unreliable execution	2	1	0
DoS: resource consumption	2	1	0
Execute unauthorized code or commands	4	2	0
Gain privileges / assume identity, Bypass protection mechanism	2	1	0
Hide activities	2	1	0

- 등급별 기준: 해당 침해 형태들의 점수를 합산하여 이를 기반으로 최종 점수를 산정한다. Table 3은 침해 형태 척도의 등급별 평가 기준을 보여준다.

Table 3. Metrics for technical impcts

Rating	Metic Point	Scope
Critical	1	6~
High	0.9	4~5
Medium	0.8	2~3
Low	0.7	1
None	0	0

5) 접근 벡터

- 평가 내용: 취약점의 주요 공격 형태를 참고하여, 침해를 위하여 주로 사용되는 접근의 제한 정도를 평가한다. CWSS는 내부의 네트워크 접속에 대하여 Intranet, Private, Adjacent Network의 세분화된 분류를 제공하고 있으나, 평가의 용이성을 위하여 이 세 등급을 Adjacent Network로 통합하였다.
- 등급별 기준 : 접근 벡터 척도의 등급별 평가 기준은 다음과 같다.

- Internet(1):일반적인 인터넷을 통하여 취약점을 침해할 수 있다.
- Adjacent Network (0.9): 방화벽 등으로 차단된 사업체의 인트라넷이나, 신뢰되는 그룹만이 접근할 수 있는 개별 네트워크 또는 물리적으로 연결된 지역 이더넷 세그먼트 등의 인터페이스를 통한 접근이 필요하다.
- Local(0.8): 쉘 계정과 같이 운영체제에 대하여 직접 명령어를 수행하는 접근이 필요하다.
- Physical(0.7): 시스템에 대하여 USB, 키보드, CD, 마우스 등을 사용한 직접적인 물리적 접근이 있어야 침해가 가능한 경우이다.
- Default(0.85): 평가할 수 없는 경우 부여한다.

6) 권한 요구도

- 평가 내용: 공격자가 취약점에 대한 공격을 수행하기 위하여 필요한 접근 권한을 평가한다. CAPEC[14] 등의 취약점의 공격방법 예시를 참고하여, 침해를 위하여 필요한 권한을 판단한다.
- 등급별 기준 : 권한 요구도 척도의 등급별 평가 기준은 다음과 같다.
  - None(1): 취약점을 가진 코드에 접근하기 위하여 아무 권한도 필요하지 않음 경우를 말한다. 공개되어 있는 웹 페이지를 위한 웹 응용프로그램에서 발생되는 보안취약점이나 이메일 등을 통한 공격은 None으로 평가한다.
  - Guest(0.9): 특정한 관리자의 허락을 요구하지 않고, 불특정 다수에게 허용되는 회원가입 등을 통하여 접근할 수 있는 프로그램 코드의 경우에 해당된다.
  - Regular User(0.8): 특별한 관리자 권한이 없는 정규 사용자 권한을 필요로 하는 경우를 말한다.
  - Adminis-trator(0.7): 해당 소프트웨어와 운영체제 전체에 대한 접근 권한을 가진 시스템 관리자 권한이 필요한 경우를 말한다.
  - Default(0.85): 해당 취약점에 대한 공격이 시스템의 환경에 따라 다양한 권한을 요구하는 경우 Default 등급으로 하며, 점수는 Guest와 Regular User의 중간값을 부여한다.

7) 상호작용 정도

- 평가 내용: 취약점을 공격하는데 필요한 피공격자의 협조적인 행동의 요구 수준을 평가한다. 시스템의 환경에 따라 여러 기준에 모두 해당하는 경우 해당 값들의 중간(median)을 부여한다.
- 등급별 기준 : 상호작용 정도 척도의 등급별 평가 기준은 다음과 같다.
  - Automated(1): 희생자의 협조적인 행동이 필요 없다.
  - Limited(0.9): 희생자의 일반적인 행동(이메일 열람, 웹페이지 접근)이 동반되어야 침해가 가능하다.
  - Moderate(0.8): 희생자가 경고 메시지를 무시하는 것과 같은 어느 정도 위험할 수 있는 작업을 수행하여야 해당 보안약점에 대한 공격이 이루어진다.
  - High(0.7): 희생자가 잘못된 행동을 하도록 희생자에 대한 직접적인 접근을 포함한 복잡한 사회적 작업을 수행하여야 한다.

8) 침해 가능성

- 평가 내용: 공격에 필요한 권한과 접근 및 희생자의 협조적인 행동을 획득한 공격자가 수행한 공격이 실제적인 침해로 연결될 가능성을 기법의 난이도 및 취약점의 성격, 실제적인 피해의 발생 가능성 등을 반영하여 평가한다. 본 평가는 취약점 자체 및 공격 시도 과정의 기술적 난이도만 반영하며, 특정 소프트웨어 설정에 대한 의존성, 접근 권한의 획득, 접근 벡터, 사용자와의 상호작용 요구 정도 등의 특성은 고려하지 않는다.
- 등급별 기준: 침해 가능성 척도의 등급별 평가 기준은 다음과 같다.
  - High(1): 해당하는 약점에 대한 표준적인 공격 기법이 존재하거나 해당 소프트웨어의 취약점에 해당하는 공격 방법이 알려져 있으며, 침해 연계 가능성이 높은 경우
  - Medium(0.9): 해당 소프트웨어의 취약점에 해당하는 공격 방법의 개발이 실제적으로 가능하고 침해 연계 가능성이 높은 경우
  - Low(0.8): 해당 소프트웨어의 취약점에 해당하는 공격 방법이 개발이 실제적으로 어렵거나, 하

여도 보안 침해의 발생 가능성이 낮은 경우이다.
- Very Low(0.7): 해당 소프트웨어의 취약점에 해당하는 공격 방법이 개발이 실제적으로 어려우며, 보안 침해의 발생 가능성이 낮은 경우
- None(0): 해당 취약점에 대한 공격으로 인한 보안 침해가 발생할 가능성이 없을 경우이다.

9) 교정 난이도

- 평가 내용: 취약점을 제거하는데 필요한 난이도를 평가하며, 이는 코드 수정의 난이도와 함께 공식적인 패치의 존재 여부 및 패치 적용의 난이도도 함께 평가한다.
- 등급별 기준: 교정 난이도 척도의 등급별 평가 기준은 다음과 같다.
  - Extensive(1): 공식적인 패치는 존재하지 않으며, 교정을 위하여 설계와 전체 시스템 구조의 수정과 같은 전체적인 수정이 필요하여 상당한 작업과 시간이 필요하다.
  - Moderate(0.9): 소스 파일의 복수개의 모듈 수정과 같은 중간 정도의 수정이 필요하며, 설계와 구조에 대한 수정은 필요 없다. 또는, 공식적인 패치가 존재하나, 해당 패치를 적용하기 위해서는 관련 시스템이 일정기간 중단되어야 하는 등 전체 서비스에 지장을 초래할 수 있다.
  - Limited(0.8): 한 모듈 내의 적은 수의 라인의 코드에 대한 수정을 요구하며, 일정한 수준의 노력과 시간이 필요하다. 또는 공식적인 패치가 존재하며, 관련 시스템 서비스의 운영에 대한 어려움 없이 패치의 적용이 가능하다.
  - Default(0.9): 소스코드의 미확보와 취약점 특성 등으로 인하여 필요한 난이도를 평가할 수 없을 경우 중간값(Modearte)과 동일한 점수를 부여한다.

10) 외부 제어의 효과

- 평가 내용: 소프트웨어 외부의 추가적인 시스템을 통하여 해당 취약점을 제어하는 방법의 효과를 평가한다. 환경에 따라 적용할 수 있는 기법이 달라 여러 등급에 해당할 때에는 가장 높은 점수를 부여한다.
- 등급별 기준 : 외부 제어의 효과 척도의 등급별 평가 기준은 다음과 같다.
  - None(1): 외부적으로 제어할 수 있는 방법이 없다.
  - Limited(0.9): 간단한 방법이나, 부분적인 제한만이 가능하며, 초보적인 공격에 대해서만 방어가 가능하다.
  - Moderate(0.8): 일반적으로 사용되는 방어 방법이 존재하나, 지식을 가진 공격자에 의하여 필요한 노력이 동반될 경우 침해될 수 있다.
  - Indirect(0.7): 해당 침해를 전적으로 방어하지는 못하나, 공격의 피해를 줄이는 방법이 존재한다. 예를 들어 ASLR 방법은 잘못된 코드의 수행은 막을 수 있으나 프로그램의 중단되는 결과는 감수하여야 한다.
  - Best Available(0.6): 적용 가능한 방어 방법이 존재하나, 숙련된 공격자가 다른 취약점을 함께 사용하여 공격할 경우 침해가 발생할 수 있는 가능성이 존재한다.
  - Complete(0.4): 약점에 대하여 전적으로 효과적인 방법이 존재한다. 예를 들어 sandbox 방법을 통하여 파일 접근을 제어할 수 있다.

11) 발굴 난이도

- 평가 내용: 취약점 발굴의 노력을 평가하기 위하여 채택한 특징적인 척도로서 취약점을 발굴에 필요한 노력 및 난이도를 직접적으로 평가한다.
- 등급별 기준 : 발굴 난이도 척도의 등급별 평가 기준은 다음과 같다.
  - Very High(1): 기존에 알려지지 않은 방식의 취약점으로 발굴난이도 높음
  - High(0.9): 기존에 알려지지 않은 방식의 취약점이나 발굴난이도 낮음
  - Medium(0.7): 기존에 알려진 방식의 취약점으로 이를 활용하기 위한 발굴 난이도 높음
  - Low(0.6): 기존에 알려진 방식의 취약점으로 발굴 난이도 낮음

12) 문서 완성도

- 개요: 해당 보안취약점에 대한 보고서의 완성도와 신뢰성을 평가한다. 일반적이 취약점의 중요

도 보다는 신규 취약점 포상 등에 활용하기 위한 평가 척도이다.

- 등급별 기준 : 문서 완성도 척도의 등급별 평가 기준은 다음과 같다.
  - High(1): 해당 보안취약점에 대한 침해 방법이 재현 할 수 있는 형태로 명확히 제시되어 있으며, 동작 환경 등에 대한 설명이 존재하는 경우이다.
  - Medium(0.8): 해당 보안취약점에 대한 침해 방법이 재현 할 수 있는 형태로 명확히 제시되어 있으나, 동작 환경 등에 대한 설명이 존재하지 하는 경우이다.
  - Low(0.6): 해당 보안취약점에 대한 침해 방법이 일정 수준 설명되어 있으며, 동작 환경 등에 대한 설명이 존재하는 경우이다.
  - Very Low(0.4): 취약점에 대한 침해 방법의 설명이 미흡하거나, 동작환경 등에 대한 설명의 부재로 인하여 침해의 재현이 어려울 것으로 판단되는 경우이다.

3.4 중요도 점수의 산정

각 척도별 평가 등급을 취합하여 전체 중요도 점수를 정량적으로 산정한다. 본 연구에서는 개발된 평가체계의 활용을 고려하여, 취약점 DB 구축 시 취약점의 본질적인 중요도를 포함시키기 위한 취약점 DB 점수와, 신규 취약점 발굴 작업에 대한 적절한 포상 수행을 위한 점수인 취약점 발굴 점수의 두 가지 산출식을 제시하였다.

취약점 발굴 점수의 경우에는 해당 취약점의 대응은 시스템 개발자나 사용자가 수행하여야 하는 부분이므로 대응 난이도 요소를 평가에서 제외하였으며 취약점 DB 점수는 발굴자의 보고에 대한 신뢰성 및 노력을 평가하기 위한 발굴 수준 요소를 평가에서 제외하였다. 따라서, 취약점 발굴 점수는 취약점 자체의 특성을 반영한 취약점 점수와 발굴 수준 점수를 사용하여 계산되며, 취약점 DB 점수는 취약점 점수와 대응 난이도 점수를 사용하여 계산된다. 취약점 점수와 발굴 수준 점수, 대응 난이도 점수는 0~1의 범위를 가지며, 총점은 10점 만점으로 다음과 같이 계산된다.

- 취약점 발굴 점수 =
  취약점 점수 * 8 + 발굴 수준 점수 * 2
- 취약점 DB 점수 =
  취약점 점수 * 8 + 대응 난이도 점수 * 2

취약점 점수, 발굴 수준 점수, 대응 난이도 점수의 산출 방법은 다음과 같다.

▷ 취약점 점수
= 영향도 점수 * 출현도 점수 * 공격 난이도 점수

- 영향도 점수
  = 침해 형태+(1-침해 형태)*피해의 심각성 *(침해 형태가 0이 아닌 경우)*
  0 *(침해 형태가 0인 경우)*
- 출현도 점수 = 파급 범위*0.8 + 대상 분포*0.2
- 공격 난이도 점수
  = (접근 벡터 + 권한 요구도 + 상호작용 정도) * 침해 가능성 / 3

▷ 발굴 수준 점수 = (발굴 난이도 + 문서 완성도) / 2

▷ 대응 난이도 점수
= (교정 난이도 + 외부 제어의 효과) / 2

취약점 자체의 심각성은 영향도, 출현도, 공격 난이도를 평가하여 계산된다. 영향도 점수에 있어서 기술적 영향 척도의 점수가 기본적으로 사용되며, 피해의 심각성은 부수적으로 반영되어 기술적 영향이 다양하지 않더라도 피해의 심각성이 크면 점수가 1에 가까워지도록 하여 구체적인 심각성에 해당되는 요소의 평가가 높아지도록 설정하였다. 출현도에 있어서는 일반적인 출현도의 심각성 인식에 부합하도록 소프트웨어의 파급 범위에 중점을 두어 점수를 부여하였다. 공격 난이도의 경우에는 공격의 방해 및 필요 요소와 관련된 세 가지 척도의 산술 평균과 해당 공격으로 인한 침해의 발생 가능성을 곱하여 계산함으로써, 공격으로 인한 실제 침해가 없을 경우에 이를 전반적으로 반영할 수 있도록 하였다. 발굴 수준, 대응 난이도 점수는 관련 척도의 평균을 사용하도록 하였다.

IV. 보안취약점 시범 평가

본 절에서는 국내에서 보고된 40개 신규 취약점 사례[15]에 대하여 본 연구에서 개발한 취약점 평가 방법을 사용한 평가 결과를 제시한다.

4.1 보안취약점 평가 예시

Table 4는 본 연구의 취약점 평가방법을 사용한

Table 4. Example of scoring software vulnerabilities: PHP remote code execution

<table>
<tr><th>Scope</th><th>Metrics</th><th>Score</th><th>Description</th></tr>
<tr><td rowspan="2">Prevalence</td><td>Distribution Range</td><td>W</td><td>PHP is one of the popular script languages, which is widely used for web services</td></tr>
<tr><td>Target Range</td><td>A</td><td>The vulnerability exists in all the platforms and all the configuration of systems in which the target software is installed.</td></tr>
<tr><td>Business Impact</td><td>Level of Damage</td><td>M</td><td>Physical loss and/or asset loss can happen because of execution of unauthorized code.</td></tr>
<tr><td>Technical Impact</td><td>Attack Consequences</td><td>H</td><td>
<table>
<tr><th>TI Items</th><th>Score</th></tr>
<tr><td>Modify data (2/1/0)</td><td>0</td></tr>
<tr><td>Read data (2/1/0)</td><td>0</td></tr>
<tr><td>DoS: unreliable execution (2/1/0)</td><td>0</td></tr>
<tr><td>DoS: resource consumption (2/1/0)</td><td>0</td></tr>
<tr><td>Execute unauthorized code or commands (4/2/0)</td><td>4</td></tr>
<tr><td>Gain privileges / assume identity / Bypass protection mechanism (2/1/0)</td><td>0</td></tr>
<tr><td>Hide activities (2/1/0)</td><td>0</td></tr>
<tr><td>Total (6~:C, 4~5:H, 2~3:M, ~1:L, 0:N)</td><td>4</td></tr>
</table>
</td></tr>
<tr><td rowspan="4">Likelihood of Attack</td><td>Exploitability</td><td>H</td><td>It is high possible for attackers to succeed in attacks with exploiting the vulnerability, and the attack vectors are widely known.</td></tr>
<tr><td>Access Vector</td><td>I</td><td>An successful attack through Internet is possible.</td></tr>
<tr><td>User Interaction</td><td>A</td><td>An successful attack does not require any cooperation of users.</td></tr>
<tr><td>Required Privilege</td><td>N</td><td>If the vulnerable PHP program is accessible form Internet, an attacker does not need any privilege at all.</td></tr>
<tr><td rowspan="2">Level of Correction</td><td>Difficulty of Correction</td><td>L</td><td>There exist software patches to the vulnerability, and the patches are easy to apply.</td></tr>
<tr><td>External Control</td><td>None</td><td>No successful external controls are reported for this vulnerability.</td></tr>
<tr><td rowspan="2">Level of Discovery</td><td>Level of Difficulty</td><td>L</td><td>Various access vectors are already known.</td></tr>
<tr><td>Completeness of Report</td><td>VL</td><td>It is impossible to reproduce the reported attack based on the report.</td></tr>
<tr><td colspan="2">Vulnerability Discovery Score</td><td colspan="2">8.44</td></tr>
<tr><td colspan="2">Vulnerability DB Score</td><td colspan="2">9.24</td></tr>
</table>

시범평가 결과의 예로 PHP 원격코드 실행 취약점에 대한 평가 결과이다. 각 평가항목에 대해 평가결과와 평가의 근거를 제시하고 있다. 예를 들어 피해의 심각성에 대한 평가에서는 PHP 원격코드 실행 취약점으로 인하여 임의 코드를 실행하는 서버로 인해 중간 정도의 물적 손실이나 자산 손실의 결과를 낳을 수 있으므로 M으로 평가하였다. 침해 가능성의 경우에는 공격자가 발견된 해당 취약점에 접근하여 공격할 경우 침해를 성공할 확률이 높으며 안정적인 공격 방법이 알려져 있으므로 H로 평가하였다.

4.2 보안취약점 시범 평가 결과

Table 5는 신규 보안취약점 보고사례 40건에 대한 평가 결과 중 취약점 발굴 점수가 높은 상위 10개를 나열한 것이다.

첫 번째 취약점의 경우 금융 관련 온라인 보안 솔루션으로 파급도가 W로 파악되었고, 타 척도들도 위험도가 높은 것으로 판단되어 높은 점수가 부여되었다. 일반적으로 파급도가 큰 프로그램이 높은 점수를 부여받았으며, 버퍼 넘침과 관련된 취약점들이 대부분 원격코드 실행과 관련되어 기술적 영향 점수를 높

Table 5. Top 10 scores of bountied software vulnerabilities [15]

Name	Discovery Score	Discovery Rank	DB Score	DB Rank
[13-105] Execution of unauthorized code in online banking security solution	9.63	1	9.63	1
[13-162] CVE-2013-2251 Execution of unauthorized code	8.49	2	8.79	4
[12-011] Execution of unauthorized code of PHP program	8.44	3	9.24	2
[12-016] DoS of operating system of network devices	8.20	4	9.10	3
[12-094] Heap overflow of a movie player program	8.17	5	8.27	8
[13-020] CSRF XSS vulnerability of wireless routers	8.17	5	8.17	10
[12-165] Exposure of personal information of cyber payment system	8.16	7	8.46	6
[13-122] Signed integer overflow/underflow of a word processor	7.95	8	8.05	12
[12-075] Malicious document from impersonating Congress man	7.95	8	8.05	12
[12-072] Heap overflow of a word processor	7.95	8	8.05	12

게 부여 받아 다른 종류의 취약점과 비교하여 상대적으로 높은 점수가 부여되었다. 국내 워드프로세서 관련 취약점의 경우에는 해당 프로세서가 대표적인 문서 편집 소프트웨어이어서 파급도가 필수 설치 소프트웨어 중에 가장 높은 High로 평가되었으며, 버퍼 넘침 관련 취약점으로서 기술적 영향 부분의 점수도 높게 평가되어 중요도가 상위권으로 평가되었다.

버퍼 넘침 취약점 외에는 계정 탈취나 보안 우회와 같은 권한 획득과 관련한 취약점, 크로스 사이트 스크립트 및 SQL 삽입 관련 취약점, 서비스 거부 취약점 등이 있었으며 이러한 취약점은 상대적으로 낮은 중요도로 평가되었다.

V. 결 론

본 연구에서는 기존 CWSS 및 CVSS를 분석하여 이를 개선하여 국내 활용에 적합하도록 개발된 보안취약점 중요도 정량평가 체계를 제시하였다. 개발된 평가체계는 다음과 같은 장점을 가진다.

점수를 부여함에 있어서 기존의 보안취약점 및 보안약점 평가방법에서 사용하는 평가방법의 척도 및 인터넷진흥원에서 이미 사용하고 있는 포상제 평가척도를 모두 고려하여, 가장 적합한 척도를 선택하여 사용하였다.

평가척도를 구성함에 있어 6개 필수 평가범주를 설정하고 이에 맞추어 척도를 선택함으로써 균형 있는 분석이 될 수 있도록 하였으며, 국내 실정에 맞는 적절한 평가 기준을 수립하기 위하여 국내의 파급도를 반영할 수 있도록 관련 척도를 적절히 설계하였다. 평가 공식 개발에 있어서, 보안취약점의 영향도, 출현도, 공격난이도의 취약점 특성, 발굴 수준 및 대응 난이도 등을 독립적으로 평가하고, 그 결과를 기반으로 목적에 맞는 평가 방법을 도출하도록 하여 한 가지 특성에 대한 중복되는 점수 반영을 최소화하였다. 본 취약점 평가체계는 모든 척도에 대하여 객관적 기준을 적절한 체계에 의하여 제시하였고, 중요도 산출 공식의 도출에 있어서도 논리적인 근거에 기반한 체계를 제시하였기 때문에, 설득력을 확보하면서도 향후 추가적인 환경의 변화 또는 요구조건의 변화에도 적절히 대처하여 갱신하는 것도 용이할 것으로 판단된다.

제안된 평가방법을 사용하여 최신 국내 보안취약점 보고 사례 40건에 대한 시범평가를 수행하였으며, 그 결과를 국내 전문가들의 평가 결과들과 비교,

점검하여 평가 척도 및 공식을 개선하는 과정을 거침으로써 설득력 있는 평가가 이루어질 수 있는 평가 체계를 도출하였다.

개발된 보안취약점 평가체계는 한국인터넷진흥원에서 진행하고 있는 S/W 신규 보안 취약점 신고 포상제와 같은 보안취약점 발굴 포상제의 근거로 활용이 가능하며, 또한 CVE와 같은 보안취약점에 대한 종합적인 DB 구축이 국내에서도 추진될 경우 보안취약점 중요도 점수를 객관적으로 제공하는데 사용할 수 있을 것으로 판단된다.

References

[1] Gartner, "Now is the time for security at application level," http://www.gartner.com/id=487227, December, 2005.

[2] Chan-Kyu Park, Hyong-Shik Kim, Tae Jin Lee, Jae-Cheol Ryou, "Function partitioning methods for malware variant similarity comparison," Journal of The Korea Institute of information Security & Cryptology, 25(2), pp. 321-330, Apr. 2015

[3] Min Jae Jo, Ji Sun Shin, "Study on Security Vulnerabilities of Implicit Intents in Android," Journal of The Korea Institute of information Security & Cryptology, 24(6), pp. 1175-1184, Dec. 2014

[4] Jinseok Park, Heesoo Kang, Seungjoo Kim, "How to Combine Secure Software Development Lifecycle into Common Criteria," Journal of The Korea Institute of information Security & Cryptology, 24(1), pp. 171-182, Feb. 2014

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

[6] Common Vulnerabilities and Exposures (CVE), http://cve.mitre.org

[7] National Vulnerability Database (NVD), http://nvd.nist.gov

[8] 2011 CWE/SANS Top 25 Most Dangerous Software Errors, http://cwe.mitre.org/top25/

[9] 2010 OWASP (The Open Web Application Security Project) Top 10, http://www.owasp.org

[10] Software development security, Guideline for governmental software systems, Chapter 6, http://www.law.go.kr/LSW/admRulInfoP.do?admRulSeq=2000000099405

[11] Bounty program for new SW vulnerabilities, Korea Internet & Security Agency Korea Internet Security Center (KISC), https://www.krcert.or.kr/kor/consult/consult_04.jsp

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

[13] Common Vulnerability Scoring System (CVSS-SIG), http://www.first.org/cvss

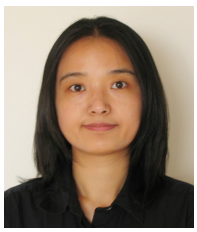
[14] CAPEC - Common Attack Pattern Enumeration and Classification, http://capec.mitre.org/

[15] Joonseon Ahn, Byeong-Mo Chang, Eunyoung Lee, "Research on Software Vulnerability Scoring Systems," Korea Internet & Security Agency, Korea, 2013

〈저 자 소 개〉

안 준 선 (Ahn, Joonseon) 종신회원
1992년 2월: 서울대학교 계산통계학과 졸업
1994년 2월: KAIST 전산학과 석사 졸업
2000년 8월: KAIST 전자전산학과 박사 졸업
2001년 9월~현재: 한국항공대학교 항공전자정보공학부 교수
〈관심분야〉 프로그래밍언어, 프로그램 분석, 소프트웨어 보안

창 병 모 (Chang, Byeong-Mo) 정회원
1988년 2월: 서울대학교 컴퓨터공학과 졸업
1990년 2월: KAIST 전산학과 석사 졸업
1994년 2월: KAIST 전산학과 박사 졸업
1995년 3월~현재: 숙명여자대학교 컴퓨터과학부 교수
<관심분야> 프로그래밍 언어, 프로그램 분석, 소프트웨어 보안



이 은 영 (Lee, Eunyoung) 종신회원
1996년 2월: 고려대학교 전산과학과 졸업
1998년 8월: 고려대학교 전산과학과 석사
2004년 1월: Princeton University 전산학 박사
2005년 3월~현재: 동덕여자대학교 컴퓨터학과 부교수
<관심분야> 소프트웨어 보안, 프로그래밍 언어, 클라우드 컴퓨팅