Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.6, Dec. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.6.1449

공개 취약점 정보를 활용한 소프트웨어 취약점 위험도 스코어링 시스템*

김 민 철,[1†] 오 세 준,[1] 강 현 재,[1] 김 진 수,[2] 김 휘 강[1‡]
[1]고려대학교 정보보호대학원, [2]국방과학연구소

Risk Scoring System for Software Vulnerability Using Public Vulnerability Information*

Min Cheol Kim,[1†] Sejoon Oh,[1] Hyunjae Kang,[1] Jinsoo Kim,[2] Huy Kang Kim[1‡]
[1]Graduate School of Information Security, Korea University,
[2]Agency for Defense Development

요 약

소프트웨어 취약점의 수가 해마다 증가함에 따라 소프트웨어에 대한 공격 역시 많이 발생하고 있다. 이에 따라 보안 관리자는 소프트웨어에 대한 취약점을 파악하고 패치 해야 한다. 그러나 모든 취약점에 대한 패치는 현실적으로 어렵기 때문에 패치의 우선순위를 정하는 것이 중요하다. 본 논문에서는 NIST(National Institute of Standards and Technology)에서 제공하는 취약점 자체 정보와 더불어, 공격 패턴이나 취약점을 유발하는 약점에 대한 영향을 추가적으로 고려하여 취약점의 위험도 평가 척도를 확장한 스코어링 시스템을 제안하였다. 제안하는 스코어링 시스템은 CWSS의 평가 척도를 기반으로 확장했으며, 어느 기업에서나 용이하게 사용할 수 있도록 공개된 취약점 정보만을 활용하였다. 이 논문에서 실험을 통해 제안한 자동화된 시스템을 소프트웨어 취약점에 적용함으로써, 공격 패턴과 약점에 의한 영향을 고려한 확장 평가 척도가 유의미한 값을 보이는 것을 확인하였다.

ABSTRACT

As the number of software vulnerabilities grows year by year, attacks on software are also taking place a lot. As a result, the security administrator must identify and patch vulnerabilities in the software. However, it is important to prioritize the patches because patches for all vulnerabilities are realistically hard. In this paper, we propose a scoring system that expands the scale of risk assessment metric by taking into consideration attack patterns or weaknesses cause vulnerabilities with the vulnerability information provided by the NIST(National Institute of Standards and Technology). The proposed scoring system is expanded based on the CWSS and uses only public vulnerability information to utilize easily for any company. In this paper, we applied the automated scoring system to software vulnerabilities, and showed the expanded metrics with consideration for influence of attack pattern and weakness are meaningful.

Keywords: scoring system, risk based prioritization, CVE, CWE, CVSS

Received(09. 10. 2018), Modified(10. 12. 2018), Accepted(10. 30. 2018)
* 본 연구는 국방과학연구소 연구과제(UD180011ED) 지원을 받아 수행하였습니다.
† 주저자, gomcheol@korea.ac.kr
‡ 교신저자, cenda@korea.ac.kr(Corresponding author)

I. 서 론

보안 업체 Risk Based Security가 발표한 연말 취약점 보고서에 의하면 2017년 한 해 동안 발견된 취약점 수는 20,832개로 전년대비 31% 증가하였다[1]. 이처럼 취약점의 수가 증가함에 따라 소프트웨어에 대한 공격 역시 많이 발생하고 있다. 미국의 시장 조사 전문 업체인 Forrester는 2017년 세계 보안 연구조사를 통해 2017년에 발생한 데이터 유출 사고의 41%가 소프트웨어 취약점에서 시작되었다고 발표하였다[2]. 그렇기 때문에 보안 사고를 미리 예방하기 위해서 소프트웨어에 대한 안정성을 파악해야 한다. 또한, 보안 업체인 RAPID7의 모의 침투 테스터들에 의한 보고서인 "Under the Hoodie: 2018"에 의하면, 2017년 9월 초부터 2018년 6월 중순까지 계약한 178개의 기업 중 소프트웨어 취약점에 안전한 기업은 4% 이하라고 발표하였다. 다시 말하면 나머지 96%의 기업은 적어도 하나의 소프트웨어 취약점으로 공격받을 수 있다는 것을 의미한다. 이처럼 대부분의 기업들은 소프트웨어 취약점에 노출되어 있다는 것을 알 수 있다. 게다가 모의 침투 테스터들은 제로 데이 취약점의 희소성에 대해 강조하였다[3]. 즉, 실제로 공격자는 완전히 새로운 소프트웨어 취약점을 찾지 않고 기존에 알려진 취약점을 이용하여 공격을 한다는 것이다. 따라서 알려진 소프트웨어 취약점의 위험도를 인지할 필요가 있다.

보안을 설계단계에서부터 고려하여 개발했다 하더라도, 일반적으로 소프트웨어에는 알려진 또는 알려지지 않은 많은 취약점이 존재한다. 그러나 인력 및 비용 등의 문제가 있기 때문에 현실적으로 모든 취약점을 패치 할 수 없다. 게다가 취약점이 공개된 이후 패치되기 전까지 약 30일이 가장 위험한 기간이기 때문에 어느 취약점을 가장 우선적으로 패치 할 것인지 신속하게 정하는 것이 중요하다[4]. 취약점의 우선순위를 정하기 위한 쉬운 방법으로, 위험도를 수치화하는 방법이 있다. 이런 점에서 표준화된 위험도 수치화 프레임워크로 취약점에 대한 CVSS(Common Vulnerability Scoring System)와 약점에 대한 CWSS(Common Weakness Scoring System)가 많이 사용된다[5].

약점(weakness)은 취약점의 원인이 되는 소프트웨어의 구조, 설계, 코드 또는 구현 단계에서 발생할 수 있는 결함, 버그 등의 오류를 의미한다. 취약점(vulnerability)은 해킹 등의 외부 공격으로 보안 사고의 실제 원인이 되는 시스템 상의 보안 허점을 의미한다. 쉽게 말해서 약점을 발생시킬 수 있는 구체적인 사례라고 생각할 수 있다. 따라서 소프트웨어 약점은 취약점의 근본적인 원인과 관련이 있으며, 취약점을 평가할 때 취약점을 유발한 약점 자체의 심각성이나 영향도를 고려하여 더욱 다양한 정보를 이용할 수 있는 계기가 된다.

CVSS는 CVSSv3을 기준으로 공격 벡터, 필요한 권한, 보고서 신뢰도 등을 포함한 15가지의 평가 척도로 구성되어 있고 CWSS는 획득한 권한 및 계층, 기술적 영향, 내부 및 외부 통제 효과 등을 포함한 16가지의 평가 척도로 구성되어 있다. 모든 평가 척도를 사용하는 것도 가능하나, 일부는 특정 사용 환경에 종속적이거나 충분히 객관적이지 않기 때문에 일반적으로 사용하기에는 어려운 부분이 많다. 이에 따라 많은 사람들은 취약점에 대한 위험도 점수로 NVD(National Vulnerability Database)에서 제공하는 CVSS 점수를 참고한다. 그러나 이것은 기본 평가 척도만 사용한 점수이기 때문에 다양한 부분을 평가할 수 없고 취약점에 국한된 점수이기 때문에 취약점을 유발한 약점과 연관된 부분은 평가할 수 없다.

본 논문에서는 NIST에서 제공하는 취약점 자체 정보와 더불어, 공격 패턴이나 취약점을 유발하는 약점에 대한 영향을 추가적으로 고려하여 취약점의 위험도 평가 척도를 확장한 스코어링 시스템을 제안하였다. CWSS는 검증된 평가 시스템이며 약점에 대한 영향을 평가하기에 적합하다. 따라서 제안하는 스코어링 시스템은 CWSS의 평가 척도를 기반으로 확장했으며, 어느 기업에서나 용이하게 사용할 수 있도록 공개된 취약점 정보만을 활용하였다. 공개된 취약점 정보로 CVE(Common Vulnerabilities and Exposures), CWE(Common Weakness Enumeration), CAPEC(Common Attack Pattern Enumeration and Classification), CVSS, Exploit-DB[1]정보를 사용하였다.

본 논문은 다음과 같이 구성된다. 2장에서는

1) CVE: https://cve.mitre.org/
CWE: https://cwe.mitre.org/
CAPEC: https://capec.mitre.org/
CVSS: https://www.first.org/cvss/
Exploit-DB: https://www.exploit-db.com/

Table 1. CVSSv2 Metrics

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

CVSS와 CWSS에 대해 소개한 후, 취약점 위험도 평가 방법으로 CVSS와 CWSS를 활용한 연구를 살펴보고, 3장에서는 우리의 취약점의 위험도 스코어링 시스템을 제안한다. 이어서 4장에서는 상용 소프트웨어 취약점에 제안하는 스코어링 시스템을 적용한 결과를 분석하고, 마지막으로 5장에서 연구 결과를 정리한다.

II. 관련 연구

본 장에서는 취약점 및 약점에 대한 표준화된 점수화 프레임워크인 CVSS와 CWSS에 대해 소개한 후 이를 이용한 연구들을 살펴본다.

2.1 CVSS(Common Vulnerability Scoring System)

CVSS는 미국의 비영리조직인 FIRST(Forum of Incident Response and Security Teams)에 의해 관리 되고 있으며 전반적인 소프트웨어 및 하드웨어 취약점에 대한 주요 특징을 파악하고 심각성을 반영하는 수치 점수 산출 방법을 제공한다[6]. 이를 이용하여 미국의 국립표준기술연구소인 NIST는 NVD를 통해 CVE가 발견되면 CVSS 점수를 산출하여 제공한다. 현재 CVSS 점수는 두 가지 버전으로 제공된다. CVSSv2는 오랫동안 정착되어 사용되었기 때문에 CVSSv2 점수를 계속해서 제공하고 있고, CVSSv3은 2015년 3월에 공개되어 2015년 12월 20일 이전에 분석된 CVE에 대한 CVSSv3 점수는 제공하지 않는다.

Table 2. CVSSv3 Metrics

Metric Group	Metric
Base	Attack Vector
	Access Complexity
	Privileges Required
	User Interaction
	Confidentiality Impact
	Integrity Impact
	Availability Impact
	Scope
Temporal	Exploit Code Maturity
	Remediation Level
	Report Confidence
Environmental	Modified Base Metrics
	Confidentiality Requirement
	Integrity Requirement
	Availability Requirement

CVSS는 취약점의 위험도를 측정하기 위해 구성요소를 세 가지 영역으로 나누었다. 첫 번째는 기본(base) 평가 척도 그룹으로, 시간과 사용자 환경에 영향을 받지 않는 취약점의 본질적인 특성을 다룬다. 그 다음은 시간(temporal) 평가 척도 그룹으로, 사용자 환경에 영향을 받지 않고 시간이 지남에 따라 변경될 수 있는 요소를 다룬다. 마지막으로 환경(environmental) 평가 척도 그룹은 특정 사용자 환경에 관련된 특성을 다룬다. CVSS의 두 가지 버전 모두 동일한 평가 척도 그룹으로 구성되어있지만 세부적인 평가 척도에 있어 약간의 차이가 있다. Table 1과 Table 2는 두 버전에 대한 세부적인 평가 척도를 나타낸다.

2.2 CWSS(Common Weakness Scoring System)

CWSS는 미국의 비영리조직인 MITRE가 진행하는 CWE 프로젝트로 소프트웨어의 약점에 대해 우선순위를 정하는 메커니즘을 제공한다[7]. CWSS는 CVSS와 유사하게 소프트웨어의 약점을 기본 발견(base finding), 공격 표면(attack surface), 환경(environmental)의 세 가지 평가 척도 그룹으로 구분하여 평가한다. Table 3은 CWSS의 세부적인 평가 척도를 보여준다.

CVSS와 달리, CWSS를 이용하여 소프트웨어 약점에 대한 점수를 산정하는 특정 기관이 없다. 다만, SANS Institute, MITRE 및 미국과 유럽의

Table 3. CWSS Metrics

Metric Group	Metric
Basic Finding	Technical Impact (TI)
	Acquired Privilege (AP)
	Acquired Privilege Layer (AL)
	Internal Control Effectiveness (IC)
	Finding Confidence (FC)
Attack Surface	Required Privilege (RP)
	Required Privilege Layer (RL)
	Access Vector (AV)
	Authentication Strength (AS)
	Level of Interaction (IN)
	Deployment Scope (SC)
Environmental	Business Impact (BI)
	Likelihood of Discovery (DI)
	Likelihood of Exploit (EX)
	External Control Effectiveness (EC)
	Prevalence (P)

수많은 소프트웨어 보안 전문가들에 의해 만들어진 CWE/SANS Top 25 Most Dangerous Software Errors 목록이 있다. 이 목록은 가장 위험한 소프트웨어 약점 25개에 대해서만 한정적으로 CWSS 점수를 제공하고 있다. 이 점수들은 단지, 유행도(prevalence), 중요도(importance), 침해 가능성(likelihood of exploit)만 고려하여 평가되었다.

2.3 CVSS 및 CWSS를 활용한 연구들

NVD에서 제공하는 CVSS 점수는 기본 평가 척도에 대한 점수이기 때문에 이것만으로 소프트웨어 취약점의 우선순위를 정하기에는 제한적이다. 이에 따라 시간 및 환경 평가 척도의 일부분을 사용하는 연구가 많이 이루어졌으며 새로운 평가 척도를 개발하는 연구도 계속해서 진행되고 있다. 최근에는 취약점의 위험도를 수치화하는 방법으로 CWSS를 참고하는 연구가 이루어지고 있다.

Stefan Frei 등[8]은 알려진 취약점에 대한 데이터를 수집하고 발견 날짜, 공개 날짜, 악용 및 패치 가능 날짜에 대한 정보를 분석하여 파레토, 베이불 분포와 같이 일반적으로 사용되는 다양한 통계적 분포를 데이터에 적용하였다. 이 연구에서 제시한 통계적 모델을 통해 많은 연구에서 CVSS의 시간 평가 척도 중에 악용 가능성과 교정 수준을 점수화하였다.

다음 연구들은 CVSS의 평가 척도 중 환경 평가 척도 부분에서 차별성을 두었다. Christian Frühwirth와 Tomi Männistö[9]는 환경 평가 척도를 평가하기 위해 기업의 보안정책을 고려해야 한다고 제안하였다. 보안정책 평가는 실제 여러 기업의 보안 관리자와 인터뷰를 통해 이루어졌다. Laurent Gallon[10]은 환경 평가 척도 중 보안 요구조건을 분석하여 CVSS 점수를 계산할 경우 조직의 보안 수준을 고려해야 한다고 주장하여 앞선 연구에 힘을 실었다. 또한, Ruyi Wang 등[11]은 환경 평가 척도와 같은 주관적인 요소를 제거하고 기본 평가 척도에 호스트 환경을 반영할 수 있는 평가 척도로 서버와 운영체제 유형을 추가하였다.

시간 및 환경 평가 척도를 그대로 이용하는 대신에 세부 평가 척도를 새로 개발하는 연구도 진행되었다. Anshu Tripathi와 Umesh Kumar Singh[12]는 NVD에서 제공하는 CVSS 점수에 추가로 수정 가능한 패치에 대한 존재 여부와 취약점의 나이를 고려하여 CVSS 점수를 계산하였다. 이를 발전시켜 Umesh Kumar Singh와 Chanchala Joshi[13]는 NVD에서 제공하는 CVE 정보와 CVSS 평가 척도를 활용하여 시간 평가 척도로 취약점의 성숙도와 패치의 가용성을 이용하였고 환경 평가 척도로 사용자 환경에 대한 빈도수를 적용하였다. 또한, 추가 연구에서 사용자 환경에서 효과적으로 공격 확률을 식별하는 평가 방법을 제안하였다[14]. 이 방법에서 CVSS 기본 평가 척도와 환경 평가 척도를 이용하여 5가지 항목을 측정하였다.

Siv Hilde Houmb와 Virginia N.L. Franqueira[15]는 기존의 CVSS의 평가 척도를 두 가지로 나누어 빈도와 영향 측면에서 조건부 확률을 이용하여 대상의 위험도를 평가하는 모델을 제안하였다. 빈도 측면에는 사용 가능한 공격 도구, 기존의 보안 척도, 보고서 신뢰도 요소를 추가하여 평가가 이루어졌다.

Candace Suh-Lee와 Juyeon Jo[16]는 신뢰할 수 없는 네트워크에 대한 근접성과 인접 호스트의 취약점 개수 및 유형을 파악하여 취약점의 위험도를 계산하였다. 인접한 호스트와 네트워크의 정보를 활용하여 정적인 CVSS 평가 방법을 보완하였다.

Young Hoon Moon 등[17]은 CVSS 스코어를

Table 4. Metrics in our scoring system

Metric Group	Metric
Impact	Technical Impact (TI)
	Severity (S)
	Acquired Privilege (AP)
Appearance	Finding Confidence (FC)
	Deployment Scope (OC/LC)
Level of Difficulty	Required Privilege (RP)
	Access Vector (AV)
	Authentication (A)
	Interaction (I)
	Likelihood of Exploit (EX)

기본으로 하되, OSVDB에 공개된 취약점 정보를 바탕으로 운영체제, 소프트웨어, 인증기법별 평판점수를 산정하여 attack graph를 생성하는 알고리즘과 시스템을 제안하였다.

지금까지 살펴본 연구들은 모두 취약점에 대한 위험도를 수치화하기 위해 CVSS를 바탕으로 하였다. 그러나 Joonseon Ahn 등[18]은 CVSS와 CWSS의 평가 척도를 분석하여 국내 활용에 적합하도록 12가지의 평가 척도로 구성된 취약점 중요도 정량평가 시스템을 제시하였다.

지금까지 살펴본 많은 연구에서 취약점의 위험도를 평가할 때 취약점 자체에 대한 정보를 활용하여 새로운 평가 척도를 개발하였다. 이와 달리, 본 논문에서는 취약점과 관련된 공격 패턴과 약점의 영향을 추가적으로 고려하였다.

III. 제안하는 취약점 스코어링 시스템

본 장에서는 기존의 취약점 및 약점에 대한 평가 방법론인 CVSS와 CWSS의 특징을 분석한 후, 우리가 개발한 취약점 점수화 시스템을 설명하였다.

3.1 기존의 취약점 평가 방법론 특징 분석

CVSS 점수는 취약점의 위험도를 측정하기 위해 사용하는 대표적인 평가 체계이며, 기본, 시간, 환경적 측면에서 다양한 척도를 제시하고 있다. 그러나 NVD에서는 CVSS의 기본(base) 평가 척도 점수만 사용하고 있다. 또한 취약점으로 획득 가능한 권한이나 공격 코드의 발견 가능성 등이 고려되고 있지 않다.

반대로 약점의 위험도를 측정하는 CWSS는 CVSS에서 부족한 특성을 폭넓게 다루고 있으나, NVD와 같이 대외적으로 공개된 데이터베이스가 없다. 따라서 CWSS를 활용하려면 약점들을 관리자가 직접 분석하여 점수를 계산해야 한다. 하나의 기업에서도 발견되는 약점은 매우 다양하며 시간이 흐름에 따라 새로운 약점이 계속해서 발견되기 때문에, CWSS 체계를 그대로 도입하기에 어려움이 따른다.

3.2 제안하는 취약점 스코어링 시스템

본 연구에서는 기존 취약점 평가 방법론을 개선하기 위해, 다음과 같은 관점을 고려하여 새로운 취약점 스코어링 시스템을 개발하였다.

- CVSS에서 다루지 않는 특성인 획득 가능한 권한, 공격 코드의 발견 가능성, 취약점의 침해 범위 등 6가지의 평가 척도를 추가적으로 고려하였다. 기본적으로 CWSS를 기반으로 설계하였으며, CVSS의 기본 척도를 포함하여 위에서 나열한 척도를 추가적으로 개발하였다.
- 기업에서 쉽게 적용할 수 있도록 공개된 데이터베이스(NVD, MITRE의 CWE와 CAPEC, Exploit-DB) 만을 이용하였다. 공개된 데이터베이스들에서 취약점 정보를 수집하고 있다는 가정 하에, 관리자가 취약점을 추가로 분석하지 않아도 위험도를 평가할 수 있는 시스템을 목표로 하였다. 실제로 4장에서 본 연구에서 제안한 스코어링 시스템을 실험할 때, 자동화된 시스템을 개발하여 활용하였다.

3.2.1 평가 척도 선별

제안하는 시스템에서는 두 가지 기준을 적용하여 Table 4와 같이 평가 척도를 선별하였다. 첫째, 제안하는 점수화 시스템은 자동화가 용이하도록 공개된 취약점 정보만 이용하였다. 따라서 선별된 평가 척도는 공개된 취약점 정보를 이용하여 평가 할 수 있어야 한다. 각 평가 척도의 평가 방법은 평가 척도 설정에서 설명하였다. 둘째, 선별된 평가 척도는 취약점의 위험도를 평가하기에 충분한 의미가 있어야하며 약점에 대한 영향을 고려할 수 있어야 한다. 선별된 평가 척도는 CWSS를 참고하였기 때문에 약점에 대한 영향을 고려할 수 있다. Table 4에 있는 평가 척도 중 획득 가능한 권한은 취약점으로 인해 발생하는 영향과 직접적으로 관련이 있기 때문에 취약점을

평가하기 위한 요소로 적합하다[19]. 또한, 취약점을 이용한 공격 코드가 공개되면 해당 취약점을 이용할 가능성이 매우 높기 때문에 공격 코드의 발견 가능성을 선별하였다[20]. 마지막으로 침해 범위는 취약점의 출현도를 평가할 수 있고 피해 심각성은 취약점의 영향도를 평가할 수 있기 때문에 선별된 평가 척도들은 충분한 의미를 갖는다[18].

CWSS에서 나머지 평가 척도는 다음과 같은 이유로 제외되었다.

- Acquired/Required Privilege Layer (AL/RL): 공개된 취약점 정보의 한계성에 의해 제안하는 시스템에서 평가할 수 없어 제외하였다.
- Likelihood of Discovery (DI): 본 연구의 스코어링 시스템은 공격자가 이미 취약점을 발견했다는 가정으로 패치할 취약점에 대한 우선순위를 정하는 것이기 때문에 적합하지 않다.
- Internal/External Control Effectiveness (IC/EC): 소프트웨어가 설치된 특정 환경에 의존하여 다르게 평가되기 때문에 제외하였다.
- Business Impact (BI): 해당 취약점이 적용될 수 있는 소프트웨어마다 미칠 수 있는 영향이 다르기 때문에 제외하였다.
- Prevalence (P): CWSS 권고사항에 따라 자동화된 환경에서 개별 취약점을 점수화하는 경우 적합하지 않기 때문에 제외하였다.

3.2.2 평가 척도 설정

공개된 취약점 정보를 이용해서 평가하기 때문에 그에 맞도록 평가 척도의 세부 등급을 설정하였다. 각 평가 척도의 등급 별 점수 값은 CWSS와 CVSS에서 제시하는 값을 참고하여 부여하였다[6,7]. 또한, 시스템 특성상 공개된 정보가 부족하여 평가할 수 없는 상황이 발생하는 평가 척도에 대해서 'Unknown' 등급을 추가하였다. 해당 등급의 값은 등급의 최댓값과 최솟값의 평균으로 부여하였다.

1) Technical Impact

취약점을 통해 성공적으로 공격이 이루어졌을 경우 발생할 수 있는 잠재적인 영향의 다양성을 파악한다. 평가를 위해 먼저, CVSS 벡터를 통해 해당 취약점의 CIA 영향 영역을 확인하였다. 그 다음 해당 취약점과 관련된 CWE와 CAPEC에서 제시하는 기술적 영향을 앞서 파악한 CIA 영역에 맞게 통합하여 개수에 따라 점수를 부여하였다. 기술적 영향은 메모리 변조, 데이터 변조, DoS 공격, 인가되지 않은 명령어 실행 등을 포함한 총 20가지로 구성되어 있다.

- Critical(C/1): 개수가 10개 이상인 경우
- High(H/0.75): 개수가 7개 이상 10개 미만인 경우
- Medium(M/0.5): 개수가 4개 이상 7개 미만인 경우
- Low(L/0.25): 개수가 1개 이상 4개 미만인 경우
- None(N/0): 개수가 0개인 경우

Table 5. Metric value of Severity

Metric Value	Score	Pre-Score
High (H)	1	4.0~5.0
Medium (M)	0.7	2.0~3.9
Low (L)	0.4	0~1.9

2) Severity

해당 취약점으로 인해 발생할 수 있는 공격 패턴을 통해 피해 심각성을 파악한다. 공격 패턴은 다섯 가지 등급으로 심각성을 나타내기 때문에 이를 1부터 5점까지 부여하여 Table 5와 같이 평균 점수에 따라 등급을 부여하였다.

3) Acquired Privilege

공격자가 취약점을 이용한 공격이 성공한 경우 획득할 수 있는 권한을 식별한다. NVD에서 제공하는 CVE 정보의 획득한 권한 정보를 참고하여 등급을 부여하였다.

- All Privilege(AP/1): 시스템의 모든 권한을 획득하는 경우
- Regular User(RU/0.8): 단순한 사용자 계정 권한을 획득하는 경우
- Other(O/0.6): 제한된 계정 권한 또는 게스트 계정 권한을 획득하는 경우
- None(N/0.4): 어떤 권한도 획득할 수 없는 경우
- Unknown(UK/0.7): 정보를 알 수 없는 경우

4) Finding Confidence

취약점이 실제 이용될 수 있는지에 대한 신뢰도를

평가한다. Exploit-DB 정보를 통해 실제 취약점을 이용하는 소스코드의 유무를 파악하여 등급을 부여하였다.

- Proven True(PT/1): Exploit-DB에 소스코드가 있으며 검증된 코드인 경우
- Waiting Verification(WV/0.7): 소스코드가 Exploit-DB에 있으며 아직 검증되지 않은 경우
- None(N/0.4): Exploit-DB에 소스코드가 없는 경우

5) Deployment Scope

취약점이 어떤 플랫폼에 존재할 수 있는지 파악한다. Exploit-DB에서 제공하는 플랫폼 정보와 CWE에서 제공하는 플랫폼 정보를 통합하여 평가하였다. 플랫폼 정보는 크게 운영체제와 컴퓨터 언어로 나누어 평가하였고 각각의 가중치는 동일하게 설정하였다. 'Common' 등급의 경우, 사용자의 컴퓨터 사용 환경을 파악하여 통계를 내는 StatCounter에서 제공하는 운영체제 순위[21]와 소프트웨어 품질 평가를 전문으로 하는 TIOBE에서 제공하는 컴퓨터 언어 순위[22]를 참고하여 부여하였다.

- All(A/0.5): 특정 운영체제 또는 컴퓨터 언어에 국한되지 않고 적용될 수 있는 경우
- Common(C/0.4): Android, Windows, iOS, MacOS, Linux 운영체제 또는 Java, C/C++, C#, Python, PHP, JavaScript, .NET 컴퓨터 언어에서 적용되는 경우
- Rare(R/0.25): 'Common'에서 언급되지 않은 운영체제 또는 컴퓨터 언어에서 적용되는 경우
- Unknown(UK/0.38): 적용되는 운영체제 또는 컴퓨터 언어를 알 수 없는 경우

6) Required Privilege

공격자가 해당 취약점을 이용하여 공격하기 위해 필요한 권한을 식별한다. CVSSv3의 Privileges Required 평가 척도를 이용하여 등급을 부여하였다. 해당 취약점에 대한 CVSSv3 정보가 없는 경우, 필요한 권한을 파악할 수 없기 때문에 'Unknown' 등급을 부여하였다.

- None(N/1): 공격을 하는데 어떤 권한도 필요하지 않은 경우로 CVSSv3의 PR이 'None'인 경우
- Regular User(RU/0.8): 특별한 권한 없이 단순한 사용자 계정의 권한만 필요한 경우로 CVSSv3의 PR이 'Low'인 경우
- Administrator(A/0.4): 소프트웨어나 운영체제에 대한 완전한 통제를 할 수 있는 권한이 필요한 경우로 CVSSv3의 PR이 'High'인 경우
- Unknown(UK/0.7): 해당 취약점에 대한 CVSSv3 정보가 없는 경우

7) Access Vector

공격자가 취약점을 이용하기 위해 통해야하는 채널을 식별한다. 접근 벡터를 평가하기 위해 CVSSv2의 'Access Vector', CVSSv3의 'Attak Vector'를 이용하였다. CVSSv3의 정보를 우선으로 하며 CVSSv2에는 'Physical'을 구분하지 않고 'Local'로 통합하여 평가했기 때문에 등급을 따로 부여하였다.

- Internet(I/1): 취약점을 이용하기 위해서 일반적인 인터넷에 접속해야하는 경우로 CVSS의 AV가 'Network'인 경우
- Adjacent Network(A/0.7): 취약한 소프트웨어의 브로드캐스트 또는 충돌 도메인과 같은 네트워크에 대한 물리적 인터페이스에 액세스해야 하는 경우로 CVSS의 AV가 'Adjacent Network'인 경우
- Local(L/0.6): 취약점을 이용하기 위해 운영체제와 상호작용하는 쉘 계정이 필요한 경우로 CVSSv3의 AV가 'Local'인 경우
- Local Physical(LP/0.4): CVSSv2 정보만 있으며 AV가 'Local'인 경우
- Physical(P/0.2): 소프트웨어가 실행되는 시스템에 물리적으로 액세스하거나 USB, CD, 키보드, 마우스 등을 사용하여 시스템과 상호작용이 필요한 경우로 CVSSv3의 AV가 'Physical'인 경우

8) Authentication

공격자가 취약점을 이용하기 위해 인증해야하는 횟수를 파악한다. CVSSv2의 'Authentication'을 이용하여 평가하였다.

- None(N/1): 인증이 필요하지 않은 경우로 CVSSv2의 Au가 'None'인 경우
- Single(S/0.7): 한 번의 인증만 필요한 경우로 CVSSv2의 Au가 'Single'인 경우
- Multiple(M/0.4): 두 번 이상 인증이 필요한 경우로 CVSSv2의 Au가 'Multiple'인 경우

9) Interaction

Table 6. Metric value of Likelihood of Exploit

Metric Value	Score	Pre-Score
High (H)	1	5.0~6.0
Medium (M)	0.7	3.0~4.9
Low (L)	0.4	0~2.9

공격이 성공하기 위해 공격 대상에게 요구되는 상호작용의 필요여부를 다룬다. NVD에서 제공하는 CVE 정보를 통해 평가하였다.

- None(N/1): 공격 대상과의 상호작용 없이 공격이 이루어지는 경우
- Required(R/0.6): 공격 대상과의 상호작용이 필요한 경우
- Unknown(UK/0.8): 상호작용 필요여부 정보가 없는 경우

10) Likelihood of Exploit

취약점이 발견될 경우 공격을 위해 필요한 능력을 가진 공격자가 성공적으로 공격을 할 가능성을 평가한다. 그러나 취약점에 대한 직접적인 공격 가능성에 대한 정보가 없기 때문에 해당 취약점으로 인해 발생할 수 있는 약점과 공격 패턴에 대한 공격 가능성을 통합하여 반영하였다. CWE와 CAPEC에서 제공하는 세 가지의 공격 가능성 등급에 대해 1부터 3까지 점수를 부여한 후, 약점과 공격 패턴의 관계성을 고려하여 각 점수를 합하고 평균을 내어 Table 6과 같이 등급을 부여하였다.

3.2.3 취약점 위험도 산정

최종적으로 각 평가 척도 점수를 취합하여 취약점의 위험도를 산정하기 위한 공식을 만들었다. 공식은 Ahn 등[18]이 사용했던 방법을 참고하여 크게 영향도, 출현도, 공격 난이도로 분류하여 점수를 계산할 수 있도록 하였으나, 식 구성과 가중치는 우리가 제시한 평가 척도에 맞게 조정하였다. 취약점의 위험도를 나타낼 때 취약점이 미칠 수 있는 영향이 가장 중요하기 때문에 CVSS와 CWSS의 설정과 같이 영향도에 중점을 두어 점수를 구성하였다. 그리고 본 연구의 스코어링 시스템은 기본적으로 취약점 단위로 위험도를 평가하기 때문에, 적용할 수 있는 플랫폼에 대한 정보를 포함하는 출현도보다 취약점 자체에 대한 정보로 구성된 난이도에 가중치를 더 부여하였다. 따라서 위와 같은 이유로 (1)과 같이 영향도 5점, 출현도 2점, 공격 난이도 3점의 합으로 10점 만점이 되도록 구성하였다.

$$VulnerabilityScore = ImpactScore + AppearanceScore + LevelOfDifficultyScore \quad (1)$$

영향도 점수는 (2)와 같이 기술적 영향, 피해 심각성, 획득한 권한으로 구성하였다. 기술적 영향은 다양성에 초점이 맞추어져 있다. 따라서 기술적 영향 점수가 높지만 피해 심각성 점수가 낮은 경우 점수가 낮아질 수 있도록 기술적 영향과 피해 심각성을 곱연산으로 구성하였다. 마지막에 5점 만점이 될 수 있도록 2.5의 가중치를 부여하였다.

$$ImpactScore = 2.5 \times (TechnicalImpactScore \times SeverityScore + AcquiredPrivilegeScore) \quad (2)$$

출현도 점수는 (3)과 같이 발견 신뢰도, 배포 범위로 구성하였다. 발견 신뢰도는 취약점을 이용할 수 있는 코드의 존재성을 나타내기 때문에 배포 범위보다 가중치를 높게 주었다. 또한, 2점 만점으로 구성하기 위해 각 요소에 가중치를 부여하였다.

$$ApperanceScore = 1.4 \times FindingConfidenceScore + 0.6 \times DeploymentScopeScore \quad (3)$$

공격 난이도 점수는 3점 만점으로 (4)와 같이 필요한 권한, 접근 벡터, 인증 횟수, 상호작용 필요여부, 침해 가능성으로 구성하였다. 침해 가능성을 제외한 요소들의 산술평균으로 구성하였으며 마지막에 침해 가능성을 곱하여 침해 가능성에 따라 점수가 반영되도록 하였다.

$$LevelOfDifficultyScore = 0.75 \times (RequiredPrivilegeScore + AccessVectorScore + AuthenticationScore + InteractionScore) \times LikelihoodOfExploitScore \quad (4)$$

IV. 적용 및 분석

영국의 뉴스 및 정보제공기업인 Reuters의

“2017 세계적 기업의 소프트웨어 시장에 대한 경향 및 2022년까지의 예상 보고서”[23]에서 공급업체 중 1위를 차지한 Microsoft의 두 소프트웨어에 우리의 스코어링 시스템을 적용하였다. 두 소프트웨어는 Windows 10 버전 1803과 Microsoft Office 2016으로, 각 소프트웨어의 공급 업체, 제품명, 버전 정보를 이용하여 NVD에서 취약점(CVE) 목록을 추출하였다. 추출된 취약점의 개수는 각각 11개, 60개였으며, Windows 10은 최신버전으로 상대적으로 취약점이 적게 확인되었다.

각 소프트웨어의 취약점에 대해 벡터 정보, 스코어링 시스템 점수, ‘Unknown(UK)’ 개수, CVSS 점수와 함께 Table 7과 Table 8에 나타내었으며, 시스템 점수를 기준으로 내림차순으로 정리하였다. UK의 개수는 동일한 점수를 갖는 취약점에 대한 우선순위를 정하는 추가 지표로 활용하였다. 그리고 표의 CVSS 점수는 v3 기준으로 기재하였으며, v3 점수가 없는 경우 v2 점수로 대체하였다.

먼저, Windows 10에 적용한 결과를 살펴보면, CVSS 점수가 가장 높은 9번 취약점의 경우 우리의 시스템에서는 낮은 우선순위를 보였다. 9번 취약점은 CVSS를 반영하는 필요한 권한, 접근 벡터, 인증 횟수, 상호작용 필요여부 평가 척도 모두 가장 높은 등급을 가졌다. 이와 같이 공격에 필요한 요구조건은 낮으나 침해 가능성이 ‘Medium’이기 때문에 공격 난이도 점수가 상대적으로 낮게 부여되었다. 또한 심

Table 7. Results of applying the proposed system to Microsoft Office 2016

No.	CVE ID	Vector											Score (UK)	CVSS
		TI	S	AP	FC	OC	LC	RP	AV	A	I	EX		
1	CVE-2015-2468	C	H	N	PT	C	C	UK	I	N	R	H	7.86(1)	9.3(v2)
2	CVE-2016-3357	C	H	N	PT	C	C	N	L	N	R	H	7.78(0)	7.8
3	CVE-2017-11882	C	H	N	WV	C	C	N	L	N	R	H	7.36(0)	7.8
4	CVE-2017-8550	C	H	N	WV	C	A	RU	I	S	R	H	7.35(0)	8.0
5	CVE-2016-7277	C	H	N	N	R	C	N	I	N	R	H	7.15(0)	9.6
6	CVE-2018-0792	C	H	N	N	R	C	N	I	N	R	H	7.15(0)	8.8
⋮														
21	CVE-2015-2477	C	H	N	N	R	C	UK	I	N	R	H	6.93(1)	9.3(v2)
22	CVE-2015-6093	C	H	N	N	R	C	UK	I	N	R	H	6.93(1)	9.3(v2)
23	CVE-2017-0261	C	H	N	N	R	A	N	L	N	R	H	6.91(0)	7.8
24	CVE-2016-0010	C	H	N	N	R	C	N	L	N	R	H	6.85(0)	7.8
⋮														
46	CVE-2016-0025	C	H	N	N	R	A	RU	L	N	R	H	6.76(0)	7.3
47	CVE-2018-1028	M	H	N	N	R	R	N	I	N	R	H	5.81(0)	8.8
48	CVE-2016-0012	M	M	N	N	R	A	N	I	N	R	H	5.59(0)	4.3
49	CVE-2016-0141	M	M	N	N	R	A	N	I	N	R	H	5.59(0)	6.5
50	CVE-2018-0950	M	M	N	N	R	A	N	I	N	R	H	5.59(0)	6.5
51	CVE-2018-1007	M	M	N	N	R	A	N	I	N	R	H	5.59(0)	5.3
52	CVE-2017-11939	M	M	N	N	R	A	RU	I	S	N	H	5.51(0)	6.5
53	CVE-2017-8676	M	M	N	N	R	A	RU	L	N	N	H	5.44(0)	3.3
54	CVE-2017-11934	M	M	N	N	R	A	N	L	N	R	H	5.29(0)	5.5
55	CVE-2018-0853	M	M	N	N	R	A	N	L	N	R	H	5.29(0)	3.3
56	CVE-2018-0919	M	M	N	N	R	A	N	L	N	R	H	5.29(0)	3.3
57	CVE-2018-8163	M	M	N	N	R	A	N	L	N	R	H	5.29(0)	5.5
58	CVE-2017-0199	L	H	N	PT	C	R	N	L	N	R	M	5.09(0)	7.8
59	CVE-2017-0260	L	H	N	N	C	R	N	L	N	R	M	4.16(0)	7.8
60	CVE-2018-0819	N	H	N	N	C	R	N	I	N	R	M	3.75(0)	6.5

각성은 높으나 침해 영향 범위가 인가되지 않은 명령어 실행으로 한정적이기 때문에 영향도 점수가 낮게 평가되었다. 또한, CVSS와 달리 실제로 거의 유사한 취약점은 동일한 점수로 평가되었다. 4, 5번 취약점은 호스트 서버의 하이퍼 V에서 발생하는 원격 코드 실행 취약점으로, 우리의 시스템과 CVSS 모두 동일한 점수로 평가되었다. 그러나 6, 7번 취약점은 커널 상에서 발생할 수 있는 정보 노출 취약점으로 우리의 시스템에서는 동일한 점수로 평가되었지만, CVSS에서는 다른 점수로 평가하였다. 최악의 경우 CVSS로는 유사한 취약점의 점수 차이로 인해 취약점 패치를 놓칠 수 있다. 반대로 1, 2, 10, 11번 취약점은 각각 VBScript Engine에서 발생할 수 있는 원격 코드 실행 취약점, Windows Media Foundation에서 발생할 수 있는 메모리 충돌 취약점, Microsoft COM에서 발생할 수 있는 원격 코드 실행 취약점, .NET과 .NET Core에서 XML 문서를 처리할 때 발생할 수 있는 DoS 취약점으로, CVSS에서는 같은 점수로 평가하였지만 우리의 시스템에서는 다른 점수로 평가하여 다른 취약점에 대한 우선순위를 명확하게 정할 수 있다.

그 다음 Microsoft Office 2016에 대한 결과는 취약점의 수가 많기 때문에, Table 8과 같이 유사한 취약점의 일부를 생략하여 나타내었다. 6~20번 취약점은 메모리에서 개체를 처리하는 방식으로 인한 원격 코드 실행 취약점으로 모두 동일한 점수로 평가되어 7~20번 취약점을 생략하였다. 비슷한 이유로 25~45번 취약점을 생략하였다. 앞선 결과와 달리, CVSS에서 가장 높은 우선순위를 갖는 5번 취약점은 영향도와 난이도 측면에서 높은 등급을 받았기 때문에 우리 시스템에서도 높은 우선순위를 보였다. 또한, 60번 취약점은 다른 취약점들과 다르게 명확한 기술적 영향에 대한 정보가 없었다. 따라서 기술적 영향에 대한 점수가 낮게 평가되어 전체적으로 위험도가 낮게 평가되었다. 그리고 Windows에서의 실험과 같이, 49, 50, 51번 취약점은 유사한 취약점으로 모두 5.59로 동일하게 평가되었다. 추가적으로 점수 분포가 10가지에서 17가지로 확장되었다. 우리의 시스템에서의 점수는 CVSS보다 폭넓은 분포를 나타내어 취약점의 개수가 많고 주어진 예산을 통해 패치할 취약점을 고려해야 하는 경우, 우선순위의 경계선을 정할 때 도움이 될 것이다.

전체적으로 두 소프트웨어에 대한 취약점은 대부분 최신의 취약점이기 때문에 Exploit-DB를 통해 알려진 공격코드를 확인할 수 없었다. 따라서 두 소프트웨어 취약점의 발견 신뢰도 평가 척도의 값은 'None'이 대부분이었다. 하지만 우리가 제안한 시스템은 자동화가 가능하기 때문에, Exploit-DB, CVE, CWE, CAPEC 정보를 주기적으로 업데이트하거나 새로운 데이터 소스를 통해 정보를 업데이트 한다면 시간이 지남에 따라 보다 취약점의 위험도를 정확하게 평가할 수 있다.

V. 결 론

본 논문에서는 취약점 자체의 특성뿐 아니라 취약점과 관련된 약점과 공격 패턴을 추가적으로 고려하여 위험도 평가 척도를 확장한 스코어링 시스템을 제안하였다. 평가 척도는 CWSS를 기반으로 CVSS에서 다루지 않는 획득 가능한 권한, 공격 코드의 발견

Table 8. Results of applying the proposed system to Windows 10

No.	CVE ID	Vector											Score (UK)	CVSS
		TI	S	AP	FC	OC	LC	RP	AV	A	I	EX		
1	CVE-2018-8174	C	H	N	WV	C	C	N	I	N	R	H	7.66(0)	7.5
2	CVE-2018-8251	C	H	N	N	R	C	N	I	N	R	H	7.15(0)	7.5
3	CVE-2018-8136	C	H	N	N	R	C	N	L	N	R	H	6.85(0)	7.8
4	CVE-2018-0959	C	H	N	N	R	A	A	A	S	N	H	6.61(0)	7.6
5	CVE-2018-0961	C	H	N	N	R	A	A	A	S	N	H	6.61(0)	7.6
6	CVE-2018-8127	M	M	N	N	R	A	RU	L	N	N	H	5.44(0)	5.5
7	CVE-2018-8207	M	M	N	N	R	A	RU	L	N	N	H	5.44(0)	4.7
8	CVE-2018-8205	L	H	N	N	R	C	RU	L	N	N	H	5.12(0)	5.5
9	CVE-2018-8225	L	H	N	N	R	R	N	I	N	N	M	4.58(0)	8.1
10	CVE-2018-0824	L	L	N	WV	C	C	N	I	N	R	L	3.79(0)	7.5
11	CVE-2018-0765	L	L	N	N	R	R	N	I	N	N	L	3.31(0)	7.5

가능성, 취약점의 침해 범위 등을 포함하여 구성했기 때문에 보다 많은 측면에서 취약점을 평가할 수 있었다. 또한, 공개된 취약점 정보만을 활용했기 때문에 자동화를 통해 어느 기업에서나 쉽게 사용할 수 있도록 하였다.

제안한 시스템을 상용 소프트웨어의 취약점에 적용함으로써 CVSS와 비교하여 확장된 평가 척도에 따라 변화된 우선순위를 확인하였다. 실제로 CVSS 점수는 높지만 취약점을 이용한 검증된 소스코드가 없거나 침해 가능성이 적은 취약점은 낮은 우선순위를 부여하였다.

취약점과 관련된 공개된 정보는 지속적으로 축적되고 있다. 따라서 우리가 제안하는 자동화된 스코어링 시스템은 시간이 지날수록 더욱 정교한 결과를 보일 것이다. 또한 다양한 평가 척도로 인해 점수의 분포가 NVD에서 제공하는 CVSS 점수 분포에 비해 5배 이상 넓어졌기 때문에 우선순위의 경계선을 보다 세밀하게 정할 수 있을 것으로 기대한다. 향후 연구에서는 공개된 취약점 정보가 부족하여 평가가 잘 이루어지지 않는 평가 척도를 보완하고, 패치의 가용성과 같은 시간적 특성을 반영하는 평가 척도를 추가로 적용할 방법을 연구할 계획이다.

References

[1] Risk Based Security, "2017 Year End Vulnerability QuickView Report," https://pages.riskbasedsecurity.com/2017-q3-vulnerability-quickview-report, Feb. 2018.

[2] FORRESTER, "Top Cybersecurity Threats In 2018," https://www.forrester.com/report/Top+Cybersecurity+Threats+In+2018/-/E-RES137206, Nov. 2017.

[3] RAPID7, "Under the Hoodie: 2018," https://www.rapid7.com/globalassets/_pdfs/research/rapid7-under-the-hoodie-2018-research-report.pdf, July 2018.

[4] Ashish Arora, Ramayya Krishnan, Rahul Telang and Yubao Yang, "An empirical analysis of software vendors' patch release behavior: impact of vulnerability disclosure," Information Systems Research vol. 21, no. 1, pp. 115-132, Mar. 2010.

[5] Mengmeng Ge, Huy Kang Kim and Dong Seong Kim, "Evaluating security and availability of multiple redundancy designs when applying security patches," Dependable Systems and Networks Workshop (DSN-W), 2017 47th Annual IEEE/IFIP International Conference on. IEEE, June 2017.

[6] FIRST, "Common Vulnerability Scoring System(CVSS)" https://www.first.org/cvss/

[7] MITRE, "Common Weakness Scoring System(CWSS)" http://cwe.mitre.org/cwss/cwss_v1.0.1.html

[8] Stefan Frei, Martin May, Ulrich Fiedler and Bernhard Plattner, "Large-scale vulnerability analysis," Proceedings of the 2006 SIGCOMM workshop on Large-scale attack defense, ACM, Sep. 2006.

[9] Christian Frühwirth and Tomi Männistö, "Improving CVSS-based vulnerability prioritization and response with context information," Proceedings of the 2009 3rd international Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society, Oct. 2009.

[10] Laurent Gallon, "On the impact of environmental metrics on CVSS scores," Social Computing (SocialCom), 2010 IEEE Second International Conference on. IEEE, Aug. 2010.

[11] Ruyi Wang, Ling Gao, Qian Sun and Deheng Sun, "An improved CVSS-based vulnerability scoring mechanism," Multimedia Information Networking and Security (MINES), 2011 Third International Conference on. IEEE, Nov. 2011.

[12] Anshu Tripathi and Umesh Kumar Si

ngh, "On prioritization of vulnerability categories based on CVSS scores," Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on. IEEE, Dec. 2011.

[13] Umesh Kumar Singh and Chanchala Joshi, "Quantitative security risk evaluation using CVSS metrics by estimation of frequency and maturity of exploit," Proceedings of the World Congress on Engineering and Computer Science, vol. 1, Oct. 2016.

[14] Umesh Kumar Singh and Chanchala Joshi, "Quantifying security risk by critical network vulnerabilities assessment," International Journal of Computer Applications vol. 156, no. 13, pp. 26-33, Dec. 2016.

[15] Siv Hilde Houmb and Virginia N.L. Franqueira, "Estimating ToE risk level using CVSS," Availability, Reliability and Security, 2009, ARES'09, International Conference on. IEEE, Mar. 2009.

[16] Candace Suh-Lee and Juyeon Jo, "Quantifying security risk by measuring network risk conditions," Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on. IEEE, July 2015.

[17] Young Hoon Moon, Ji Hong Kim, Dong Seong Kim and Huy Kang Kim, "Hybrid attack path enumeration system based on reputation scores," In Computer and Information Technology (CIT), 2016 IEEE International Conference on, IEEE, pp. 241-248, Dec, 2016.

[18] Joonseon Ahn, Byeong-Mo Chang and EunYoung Lee, "Quantitative scoring system on the importance of software vulnerabilities," Journal of The Korea Institute of Information Security & Cryptology, Aug. 2015.

[19] Yeu-Pong Lai, Po-Lun Hsia, "Using the vulnerability information of computer systems to improve the network security," Computer Communications vol. 30, no. 9, pp. 2032-2047, June 2007.

[20] Thanassis Avgerinos, Sang Kil Cha, Alexandre Rebert, Edward J. Schwartz, Maverick Woo and David Brumley, "Automatic Exploit Generation," Communications of the ACM vol. 57, no. 2, pp.74-84, Feb. 2014.

[21] StatCounter GlobalStats, "Operating System Market Share Worldwide - July 2018," http://gs.statcounter.com/os-market-share, July 2018.

[22] TIOBE, "TIOBE Index for August 2018," https://www.tiobe.com/tiobe-index/, Aug. 2018.

[23] Reuters, "Global Enterprise Software Market Size, Share, Trends and Forecast by 2022 - Market Research Report 2017," https://www.reuters.com/brandfeatures/venture-capital/article?id=4981, Apr. 2017.

〈저자소개〉

김 민 철 (Min Cheol Kim) 학생회원
2017년 2월: 서울시립대학교 수학과 학사
2017년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 데이터 마이닝, 인공지능, 온라인게임 보안

오 세 준 (Sejoon Oh) 학생회원
2017년 2월: 서울시립대학교 수학과 학사
2017년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 온라인 게임 보안, 데이터 마이닝

강 현 재 (Hyunjae Kang) 학생회원
2012년 8월: 고려대학교 수학과 학사
2015년 2월: 고려대학교 정보보호대학원 정보보호학과 석사
2017년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 박사과정
〈관심분야〉 데이터 분석 기반 사이버 보안, 유저 행위 분석

김 진 수 (Jinsoo Kim) 정회원
1999년 8월: 전남대학교 컴퓨터공학과 학사
2002년 8월: 한국과학기술원 컴퓨터공학과 석사
2002년 9월~2005년 5월: 한국생명공학연구원
2006년 7월~현재: 국방과학연구소 선임연구원
〈관심분야〉 사이버 지휘통제, 악성코드 분석

김 휘 강 (Huy Kang Kim) 종신회원
1998년 2월: KAIST 산업경영학과 학사
2000년 2월: KAIST 산업공학과 석사
2009년 2월: KAIST 산업및시스템공학과 박사
2004년 5월~2010년 2월: 엔씨소프트 정보보안실장, Technical Director
2010년 3월~2014년 12월: 고려대학교 정보보호대학원 조교수
2015년 1월~현재: 고려대학교 정보보호대학원 부교수
〈관심분야〉 온라인게임 보안, 네트워크 보안, 네트워크 포렌식