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.2015.25.4.961

SIEM을 이용한 소프트웨어 취약점 탐지 모델 제안*

전 인 석,[1†] 한 근 희,[2] 김 동 원,[1] 최 진 영[2‡]
[1]고려대학교 정보보호대학원 , [2]고려대학교 융합SW대학원

Using the SIEM Software vulnerability detection model proposed*

In-seok Jeon,[1†] Keun-hee Han,[2] Dong-won Kim,[1] Jin-yung Choi[2‡]
[1]Graduate School of Information Security, Korea University
[2]Graduate School of Convergence Software, Korea University

요 약

ESM에서 SIEM으로의 발전은 더 많은 데이터를 기반으로 상관분석을 할 수 있게 되었다. 취약점 진단에서 발견된 소프트웨어 취약점을 CWE와 같은 분류 표준으로 수집을 한다면, 로그 분석 및 취합, 보안관제 및 운용 과정 등에서 통일된 유형의 메시지를 활용함으로써 초기대응단계에서의 귀중한 시간절약으로 신속하게 대응할 수 있고, 모든 대응 단계에서 일관성을 유지하여 처리할 수 있게 된다. 취약점 진단과 모니터링 단계에서 CCE, CPE, CVE, CVSS 정보를 공유하여, 사전에 정의된 위협에 대해서만 탐지하지 않고, 각 자산이 가지고 있는 소프트웨어 취약점을 유기적으로 반영할 수 있도록 하고자 하였다. 이에 본 논문에서는 SIEM의 빅데이터 분석 기법을 활용하여 소프트웨어 취약점에 대한 위협을 효과적으로 탐지하고 대응할 수 있는 모델을 제안하고 적용해본 결과 기존의 방법으로 탐지할 수 없었던 소프트웨어 취약점을 탐지함으로서 효과적임을 확인하였다.

ABSTRACT

With the advancement of SIEM from ESM, it allows deep correlated analysis using huge amount of data. By collecting software's vulnerabilities from assessment with certain classification measures (e.g., CWE), it can improve detection rate effectively, and respond to software's vulnerabilities by analyzing big data. In the phase of monitoring and vulnerability diagnosis Process, it not only detects predefined threats, but also vulnerabilities of software in each resources could promptly be applied by sharing CCE, CPE, CVE and CVSS information. This abstract proposes a model for effective detection and response of software vulnerabilities and describes effective outcomes of the model application.

Keywords: Risk Management, Software Vulnerability, Secure Coding, Managed Security Service, SIEM, ESM

I. 서 론

침해사고를 조사하다 보면 대부분의 침해사고는 소프트웨어의 취약점을 이용한 공격이 대부분을 차지하고 있다. 대다수의 기업은 이런 소프트웨어 취약점을 제거하기 위해 취약점 진단을 수행하고 소프트웨어 개발사는 소프트웨어 취약점을 제거하기 위한 패치를 배포한다. 기업은 서비스 가용성이나 호환성을 이유로 패치를 적용하지 못하는 경우도 많고, 수시로 발견되는 모든 취약점에 대한 대응을 하지 못하는 경우가 많다. 취약점 진단 단계에서 발견된 수많은 자산과 소프트웨어 취약점 정보는 모니터링 업무를 수행하는 부서로 전달되어 소프트웨어 취약점을 이용한

접수일(2015년 6월 22일), 수정일(2015년 7월 29일), 게재확정일(2015년 7월 30일)
* 본 연구는 인터넷진흥원의 소프트웨어개발보안확대방안연구로 수행되었음.
† 주저자, wilcois@ahnlab.com
‡ 교신저자, choi@formal.korea.ac.kr(Corresponding author)

위협에 대하여 신속하게 대응할 필요가 있다. 자산과 소프트웨어 취약점은 계속해서 변화하기 때문에 주기적으로 점검을 하고 그 결과는 모니터링 부서와 지속적으로 공유되어야 한다. 이 공유된 정보를 모니터링 시스템에서 효율적으로 활용할 수 있어야 소프트웨어 취약점을 이용한 위협에 효과적으로 대응할 수 있다.

ESM(Enterprise Security Management)에서 SIEM(Security Information and Event Management)으로의 발전은 빅데이터를 활용한 상관 분석을 가능하게 하였다.

이에 본 논문에서는 취약점 진단에서 수집된 정보를 활용하여 자산마다 가지고 있는 소프트웨어의 취약점을 반영하여 효과적으로 탐지가 가능한 모델을 제안하고자 한다.

II. 취약점 진단 및 모니터링 업무 분석

2.1 취약점 진단 업무 분석

취약점 진단의 주요 업무는 table 1과 같이 서비스 환경 분석, 자산 식별 및 취약점 진단, 보안계획 수립, 보안시스템 구축 및 사후관리의 절차로 진행하게 된다.

정보보호조치에 관한 지침에 의하면, "취약점 점검"이라 함은 컴퓨터의 하드웨어 또는 소프트웨어의 결함이나 체계 설계상의 허점으로 인해 사용자에게 허용된 권한 이상의 동작이나 허용된 범위 이상의 정보 열람·변조·유출을 가능하게 하는 약점에 대하여 점검하는 것을 말한다고 정의되어 있다[4].

Table 1. Vulnerability assessment process[15]

Inception / Initiation	- Analysis of present condition of business and its requirements - Confirm the scope of the vulnerability assessment and methodology - Discuss the agenda of the vulnerability assessment
Diagnosis	- Identification and valuation of information based assets - System vulnerability and threat assessment - Assess the condition of the security management
Planning	- Onsite asset analysis - Selection of critical tasks - Establishment of security master plan - Establishment of security countermeasures in each phase
Execution & Completion	- Documentation of Security Management System - Security system construction - Propose major restriction areas - Conduct security education for executives and employees with further follow-ups

일반적인 소프트웨어의 취약점은 Table 2와 같이 해당 벤더에서 자체적으로 분류하여 공유되고 있다. 대규모 소프트웨어 벤더의 경우는 자체의 취약점 분류와 CVE정보를 함께 제공하기도 하고, CVE정보만을 제공하기도 한다.

보안취약점은 보안정책 위반이나 소프트웨어 취약점, 설정 오류로 인하여 발생할 수 있는 결점으로 보안사고의 근원이다. 미국은 체계적인 취약점 관리를 위해 사이버보안 정보공유 프로젝트(CSISP:Cyber Security Information Sharing Project)를 추진하여 정부와 민간부분의 정보공유 모델을 제공하고 있고, 국토안보부의 책임 하에, MITRE社, NIST를 중심으로 국가 취약점 DB인 NVD(National Vulnerability Database), CWE(Common

Table 2. Management and deployment of vulnerability patches by vendors

Vendors	Vulnerability management
Microsoft	Released every months in a form of "MSYY-XXXX " that applies some urgent patches [10]
Adobe	Released every months in a form of "APSBYY-XX" that applies some urgent patches[11]
JAVA	No categorization code exist, distributed randomly including CVE [12]
PHP	Categorized and managed by #66048 code, distributed randomly including CVE[13]
Apache	No categorization code exist, distributed randomly including CVE [14]

Fig. 1. NVD Vulnerability management system[23]

Weakness Enumeration) 체계를 구축하여 효율적으로 통합 관리하고 있다[5]. NVD의 정보가 많아지고, 같은 취약점에 대한 정보가 상이할 경우를 대비하여 취약점 식별자 체계인 CVE를 구축하였다[22]. 또한, 취약점들을 효과적으로 분류하기 위하여 약점 분류체계인 CWE를 구축하였다[23].

SCAP(Security Content Automation Protocol)은 소프트웨어 벤더간에 서로 다른 취약점 분류기준을 통일하고 자동화하고 표준화하기 위한 구성요소이다.

과거 한국은 국가 취약점 DB인 KNVD의 구축을 추진하였지만 미국이나 일본에 비해 취약점 등의 정보가 부족한 실태이다.[9].

소프트웨어 취약점을 가장 쉽고 빠르게 인지하기 위해서는 취약점 식별자인 CVE를 활용하면 된다.

취약점 진단 도구를 제공하는 대다수의 벤더에서는 발견된 취약점에 대해서 CVE정보를 함께 제공해 주고 있다. 하지만 CVE정보를 제외한 정보는 벤더별 각자의 기준에 의해 제공되기 때문에 표준화가 필요하다.

Table 3. Components of the SCAP

Component	Description
CVE (Common Vulnerabilities and Exposures)	A dictionary of publicly known information security vulnerabilities and exposures. [25]
CCE (Common Configuration Enumeration)	Unique identifiers for common system configuration issues. [26]
CPE (Common Platform Enumeration)	A structured naming scheme for IT systems, platforms, and packages [27].
CVSS (Common Vulnerability Scoring System)	An open framework for communicating the characteristics and severity of software vulnerabilities [28]
XCCDF (Extensible Configuration Checklist Description Format)	A specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target systems. [29]
OVAL (Open Vulnerability and Assessment Language)	An information security community effort to standardize how to assess and report upon the machine state of computer systems. [30]

표준화된 양식으로 취약점을 관리하고자 하는 필요에 의해 여러 가지 시도가 되고 있으나, Table 4와 같이 취약점 진단 툴은 CVE나 CVSS를 제외한 다른 정보들에 대해서는 표준화된 정보를 제공하는 것에 대해 미흡함이 있다.

취약점 진단 없무는 자산의 위험을 측정하고 취약점의 잠재 위협수준을 파악하는 것에 목적이 있다. 따라서 취약점 진단을 수행하는 부서에서 소프트웨어 취약점 결과를 공유할 때는 아래와 같은 사항[9]이 고려되어야 한다.

- 취약점 분류체계의 수립 : 미국의 CWE를 그대로 수용하거나 새로운 분류체계를 도입할 것인지 결정
- 취약점 DB에 유지할 정보 : 미국의 NVD, CVE, CWE를 참조하여 결정하여야 한다.
- 국산/해외 취약점 DB 구분 : 국산과 해외 제품의 취약점 DB를 구분하여 해외기관과의 호환성을 유지하면서 국내 취약점 DB를 효과적으로 관리해야 한다.
- 사용자에 따른 설명 수준 : 개발자의 경우는 소프트웨어 취약점의 위치나 API(Application Programming Interface)를 알려 주어야 하지만, 일반 사용자는 단순히 패치정보만을 공유하면 된다.
- 공개/비공개 여부 결정 : 악용 될 소지가 있는 Zero-day 취약점은 패치가 될 때까지 공개되지 않아야 하며, 바로 악용될 수 있는 공격코드의 공개여부를 결정해야 한다.
- 유연한 DB구축 : 새로운 소프트웨어 취약점은 계속해서 나오고 분류방법 또한 변화하게 된다. 새로운 취약점 분류체계를 반영할 수 있도록 유연하게 구축되어야 한다.
- 언어의 선택 : 소프트웨어 취약점 정보를 국문 및 영문으로 서비스 하여, 정보공유의 활용도를 높여야 한다[9].

현재의 최종 산출되는 보고서는 산업표준이나 규정별로 생성해 주고 있으나, 통일된 산출물에 대한 분류가 없다. 따라서 각각 산출된 보고서를 기준으로 소프

Table 4. Features of vulnerability scanners

Vendors	Features
Nessus	- Commercial Use. - Scans Server's Vulnerabilities. - Supports WINDOWS/LINUX. - Supports CVE standards
Nexpose	- Commercial Use. - Scans web/network vulnerabilities.. - Synchronized with Metasploit for penetrain tests. - Supports WINDOWS/LINUX. - Supports CVE/CVSS standards
Acunetix	- Partial free - Specialized in Web Vulnerability Scanning - Supports CVE standards
AppScan	- Commercial Use - Specialized in Web Vulnerability Scanning - Supports WINDOWS/LINUX - Supports CVE standards
OpenVAS	- Open Source - Scanning network Vulnerabilities - Supports LINUX only - Supports CVE standards

트웨어 취약점을 제거하거나 보고 하는 업무를 수행하게 된다.

문제는 취약점 분석의 결과가 모니터링 업무를 수행하는 부서로 정형화된 정보를 가지고 전달 될 수 없는 것이다. 다수의 취약점 분석 도구를 이용했을 경우에 각자 다른 기준에 따라 결과가 나오기 때문에 근무자의 판단에 의해 가공하여 모니터링 시스템에 활용 되고 있다. 또한, 추후 소프트웨어 취약점이 제거되었다 하더라도 어떤 자산에 어느 취약점이 있었는지가 SIEM에서 관리되고 있지 않기 때문에, 변화하는 취약점에 대하여 유연하게 대처할 수 없는 것이다.

2.1 모니터링 업무 분석

모니터링 부서의 주요업무는 예방, 탐지, 분석, 대응, 보고로 구분되며, 탐지에 가장 많은 리소스를 투입하고 있다.

탐지는 사전에 알려진 공격들을 분석, 특징을 추출하여 탐지패턴을 만드는 것부터 시작된다. 전반적인 트래픽 급증, 급감 및 내부 정보를 절취하기 위한 해킹 시도 및 악성코드 유포와 같은 공격 시도를 SIEM을 이용하여 사전에 탐지하는 업무와 보호자산 서비스의 단절, 지연, 오류 등의 현상을 파악하고 악성코드 유포지로 사용되지 않도록 보안취약점을 찾는 것이다.

분석은 경유지 악용, 해킹메일 유포, 홈페이지 위변조, 정보자료 절취 등과 같은 해킹시도를 탐지한 뒤,

로그정보를 기반으로 공격자 정보, 시간, 방법 등을 알아내고 피해규모를 파악하는 것을 말한다. 대응은 해킹시도를 대상기관에 전달하고 분석단계에서 분석한 공격자 정보, 시간, 방법과 취약점 정보를 전달하고 피해확산을 막기 위한 조치를 진행한다[16][17].

보안장비나 시스템에서 발생하는 모든 이벤트에 대하여 근무자가 모니터링 할 수 없기 때문에, SIEM을

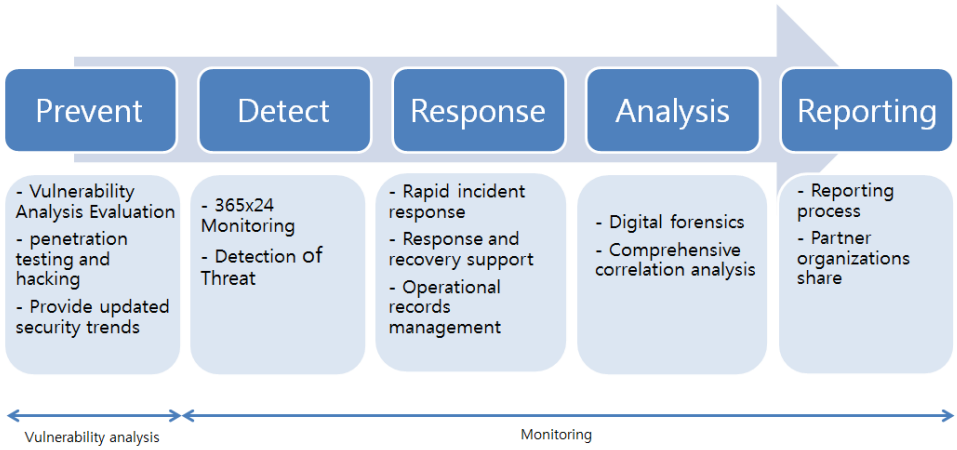

Fig. 2. security management process [15]

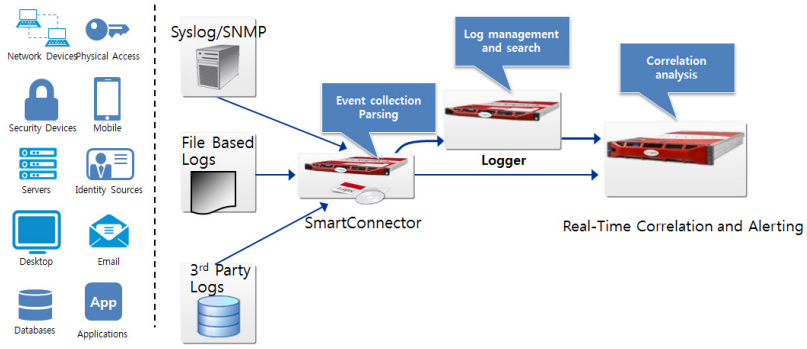

Fig. 3. Alert systems using SIEM examples (1)

이용하여 Alert이 발생하는 이벤트에 대해서 선별적으로 실시간 모니터링을 진행하고 특이사항이나 침해사고 발생시에 모든 이벤트를 전수검사 하는 방법으로 업무를 진행하고 있다.

기본적으로 SIEM은 이벤트를 SmartConnector에서 다양한 장비의 로그 정보를 수집하고 정규화 된 패턴으로 parsing을 진행한다. 이후 분석시스템에서 상관관계를 분석하여 Alert을 발생시킨다. 과거 ESM(Enterprise Security Management)은 다양한 장비의 로그를 통합하여 관리하는 것이었으며, SIEM은 각종 보안장비로부터 데이터를 수집하고, 현재의 데이터 뿐만 아니라 과거 데이터까지 상관분석을 하는 빅데이터 개념이 추가되었다.

SIEM에서 좀 더 다양한 정보를 수집하고 분석하기 때문에 어떤 정보를 획득, 분석할 것이고 그것으로 어떤 위협을 탐지할 수 있는지에 대한 고민을 해야 한다. ESM 단계에서 탐지할 수 없었던 위협을 탐지하기 위해서 유용한 정보를 더 많이 전송해야한다.

보안장비의 벤더는 대부분의 경우 매달 시그니처를 배포하고 있으며, Zero-day 취약점과 같이 빠른 시간내에 배포해야 하는 시그니처가 있을 경우에는 비정기적으로 배포하고 있다. IBM社의 Proventia의 경우는 2015년 5월 기준으로 총 5,350EA의 시그니처가 배포되었다. 해당 시그니처를 모두 Enable 설정하면 과도한 이벤트가 탐지되어 장비부하로 인한 장애가 발생할 수 있고, 모니터링 측면에서도 효율적이지 않다. 따라서 벤더에서 배포한 시그니처를 서비스 환경에 따라 Enable/Disable 설정하여, 서비스 환경에 맞는 이벤트를 탐지하고 있다.

이는 취약점 진단에서 확인된 자산의 종류와 OS, 그리고 소프트웨어에 따라 설정하게 되며, 각 시그니처 마다 영향 받는 소프트웨어와 시스템정보를 함께 제공하고 있다. 장비에 설정된 시그니처에 의해 탐지된 이벤트는 SIEM으로 전송되어 이벤트처리 알고리즘에 따라 최종적으로 모니터링 근무자가 확인해야 하는 사항에 대하여 Alert을 주게 된다.

Rules관점에서 SIEM을 바라보면 장비에서 탐지된 이벤트는 SIEM의 Filter를 통하게 된다. Filter

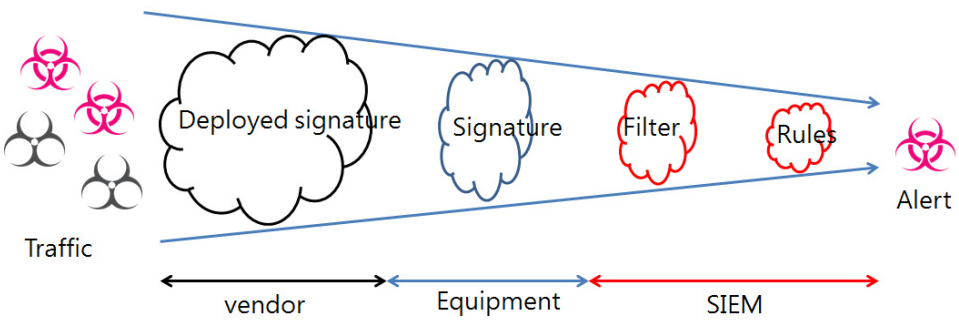

Fig. 4. Rules customize process

는 공격유형을 분류하고 각종 이벤트간에 AND, OR 과 같은 조건으로 위협이벤트를 선별한다. Filter에서는 이벤트에 특정 문자열이 있는지 여부나, 이벤트 카운트 등 장비에서 제공하는 수많은 정보를 기준으로 필터링하게 되고, 다수의 서로 다른 장비의 이벤트를 서로 상관분석을 할 수 있다.

Filtering 된 이벤트는 Rules를 통해 최종적으로 모니터링 근무자에게 Alert을 주게 된다. Rules은 Filtering 된 이벤트들을 연결하고 분석하여 모니터링 근무자에게 Alert을 하기 위한 부분이며, 최종적으로 근무자에게 보여줄 정보들을 설정하게 된다.

이런 일련의 과정을 통해 벤더가 배포한 시그니처를 서비스 환경에 맞도록 일부 시그니처를 Enable하여 그 수를 줄이고, 많은 이벤트를 발생시키는 트래픽에서 Filter를 통해 위협을 구분하고, 최종적으로 Rules을 통해 최종적으로 모니터링 근무자에게 Alert을 발생시키는 것이다. 이런 과정이 없다면 모니터링 근무자가 처리할 수 없는 양의 이벤트가 발생되기 때문에 볼 수 있는 수준으로 이벤트 발생량을 낮추고 무의미한 이벤트들을 Filtering 하는 것이다.

이와 같이 SIEM에서는 위협을 탐지하기 위한 IDS, 탐지 후 차단까지 진행하는 IPS, 그리고 웹공격에 특화되고 HTTPS 트래픽의 위협을 탐지하는 WAF(Web Application Firewall)에 이르기 까지 다수의 보안장비의 이벤트를 분석하여 위협에 대응 하고 있다.

하지만 여기서 두가지 문제점이 도출될 수 있다. 하나는 SIEM으로 넘어가면서 각종 이벤트를 상관관계 분석을 하면서 분명 과거보다 더 많은 위협을 탐지하고 있지만, 자산별로 가지고 있는 소프트웨어 취약점을 대응을 하지 못하고 있다. 전체적인 서비스에 따라서 Filter를 설정하여 전체적인 위협에는 대응하지만 일부 자산과 소프트웨어에 대해서는 대응을 할 수 없는 것이다.

예를 들어 모든 자산이 oracle을 사용하고 있다고 가정을 하면 모니터링근무자는 Rules customize process 단계에서 oracle 관련 취약점의 위험도를 높이고, 관련 취약점에 대해서 Filter를 추가하여 위협을 탐지할 것이다. 신규 서비스가 추가되면서 DB가 MS-SQL로 도입된다면 해당 MS-SQL의 위협에 대한 대응 수준이 낮아질 수 밖에 없다. 대다수의 시스템이 oracle을 쓰고 있는 상황에서 소수 도입된 MS-SQL로 인해서 MS-SQL의 모든 위협에 대하여 위험도를 높이고 Filter를 추가하게 되면, oracle 시스템에 MS-SQL 취약점을 이용한 위협이 높게 평가되는 문제가 있다. Filter 단계에서 사용하는 DB별로 설정할 수 있으나, 빈번하게 서비스가 추가되고 삭제되는 환경에서는 적절하지 않다.

Table 5. Signature management of Security solution vendors

Vendors	Feature
IBM	- CVE, CVSS, XFID Information provided - Additional information provided from ISS X-Force - Providing “Systems affected” information with lists of targeted software.[18]
Paloalto	- CVE Code provided - Provides “CVE” and “exploit codes” information with Threat names - Provides “Bugtraq IDs” Information.[19]
Wins	- CVE Code provided - Categorized events with self-standard, not support for further standard information. [20]
HP	- CVE Code provided - Manages signatures with 11 categories. - Provides “Bugtraq IDs” Information.[21]

두 번째 문제는 이처럼 다수의 보안장비가 SIEM을 통해서 보안 위협에 대하여 대응하고 있지만, 취약점 분석 도구와 마찬가지로 탐지하고자 하는 시그니처의 취약점에 대한 표준이 서로 다르다는 것이다. 따라서 특정 자산에 존재하는 소프트웨어 취약점에 대한 위협을 인지하거나 취약점이 제거된 자산에 대해서 자동으로 탐지를 제외하는 기능을 제공할 수 없다.

결국 모니터링 근무자가 서비스 환경을 이해하고, 취약점 진단 결과를 가지고 수작업으로 SIEM에 등록을 해줘야 한다. 취약점 진단 결과가 SIEM에 유연하게 적용 될 수 없는 것이다.

III. SIEM을 이용한 소프트웨어 취약점 탐지 모델

3.1 소프트웨어 취약점 탐지 모델의 개요

본 논문에서는 제시하는 모델은 보안위협 전체에서 효과적으로 위협을 탐지하는 것이 아니며, 빅데이터를 이용하여 소프트웨어 취약점을 효과적으로 탐지하고 대응하는 것으로 범위를 제한하였다. 취약점 분석 도구와 침입탐지 시스템은 표준화된 취약점 관리체계의

각 컴포넌트를 제공하고 있지 않는다. 따라서 현재 상황에서 비교적 빠르게 적용할 수 있고 소프트웨어 취약점에 대한 탐지 및 대응에 효과적일 수 있는 CCE, CPE, CVE, CVSS 정보를 이용하여 효과적인 소프트웨어 취약점 탐지 모델을 제안하고자 한다.

기존의 탐지방법론을 변경하는 것이 아니며, 기존의 탐지방법론에 자산이 가지고 있는 소프트웨어 취약점을 이용하여 Risk Management 하는 것으로서 자산에 대한 형상관리가 잘 이루어졌을 경우에 매우 효과적이다.

가장 중요한 요소는 취약점 진단의 결과가 취약점 DB로 전달되고, 해당 DB를 SIEM에서 참조하여 각 자산의 소프트웨어 취약점에 대한 정보를 인지하는 것이다. 위험도를 평가함에 있어서 기존의 방법은 각 취약점의 영향도를 기준으로 평가하였지만, 취약점 DB를 공유함으로써 해당 자산에 소프트웨어 취약점 여부를 기준으로 위험도를 평가할 수 있다. 그로인해 실제 취약점을 가지고 있는 자산에 그 취약점을 이용한 위협이 탐지되었을 경우 위험도가 높게 산정되고, 해당 취약점이 없는 자산에 위협이 탐지되었을 경우 위험도를 낮게 선정하여 실제 Vulnerability + Threat = Risk 를 기반으로 탐지할 수 있다.

취약점 DB는 취약점 진단과정에서 확인된 정보를 저장하는 DB로서 Fig 5와 같이 Identification, Vulnerability Analysis, Vulnerability collection 과정으로 구분할 수 있다.

Identification 단계에서는 각 자산의 제품과 솔루션을 파악하게 된다. 이는 추후 SIEM과 정보 공유에 있어서 고유한 식별자 역할을 하게 된다. 추가적으로 SIEM에서 수집된 이벤트와 상관분석을 하기 위해 nmap과 같은 자동화된 도구를 이용하여 IP, Port, Service를 파악한다.

Vulnerability Analysis 단계에서는 취약점 분석도구를 이용하여 각 자산이 가지고 있는 소프트웨어 취약점을 진단을 한다. Identification 단계에서 획득한 정보를 기반으로 취약점 진단을 진행하며, 장비 설정상의 취약점인 CCE와 CVE 취약점을 진단한다.

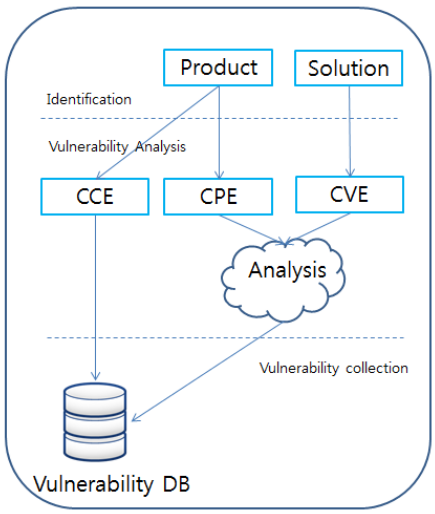

Fig. 5. Vulnerability collection Model

CVE취약점은 CPE(관련 플랫폼)에 따라 영향이 다르기 때문에 영향도를 분석하기 위하여 취약점 DB에 저장하게 된다.

최종적으로 취약점 DB에 저장되는 정보는 Table 6과 같다. 취약점 DB는 소프트웨어 취약점에 대한 탐지와 분석을 위한 목적으로 생성된 것으로 모든 위협에 효과적으로 대응하기 위해서는 더 많은 정보가 취약점 DB에 포함될 필요가 있다.

취약점 DB는 정기적/비정기적으로 취약점 진단을 진행하면서 계속해서 업데이트 될 필요가 있다. 침입탐지 시스템에서 신규 취약점에 대한 시그니처가 지속적으로 배포되기 때문에 항상 최신의 취약점 DB를 유지하여야 한다.

Table 6. Sample of Vulnerability DB

ID	IP	CPE	OS	CCE	Service	CVE
1	123.12.1.1	cpe:/h:hp:apache-based_web_server:2.0.43.00	CentOS Linux release 7.0.1406	CCE-27911-7 CCE-27667-5 CCE-27273-2 CCE-26965-4	HTTP	CVE-2013-2251 CVE-2015-0251 CVE-2015-0227
2	123.12.1.1	cpe:/a:microsoft:sql_server:2000:sp2	Windows server 2012	CCE-19831-7 CCE-19930-7 CCE-19855-6	MS-SQL	CVE-2014-4061 CVE-2009-2503 CVE-2008-3014

Table 7. NIST Base Metric Calculation[24]

BaseScore = (0.6 * Impact + 0.4 * Exploitability - 1.5)*f(x) ConfImpact=IntegImpact=AvailImpact=0=(X)=0, != 1.176 Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact)) Exploitability=20*AccessVector*AccessComplexity*Authentication

또한 취약점 진단의 결과로서 해당 취약점이 제거되었을 경우 취약점 DB에도 제거되어야 불필요한 Alert 발생을 방지할 수 있다. OS설치와 같이 DB에 저장되지 않은 취약점들이 추가될 수 있고, 파악하지 못한 취약점에 의한 위험이 있을 수 있기 때문에 자산에 대한 형상관리는 매우 중요하다.

소프트웨어 취약점 탐지 모델에서 추가된 개념은 CVSS를 산정함에 있어서 사전에 정의된 산술식에 실제 해당 자산의 취약점을 반영하여 점수를 산정하는 것이다.

Base Metic은 취약점 자체의 점수로서 상황의 변화에 따라 변화하지 않는다. 소프트웨어 취약점 탐지 모델에서는 기존의 BaseScore에서 해당 자산에 해당 취약점이 있을 경우에 가중치를 줘서 증가시키고, 반대로 해당 자산에 취약점이 없을 경우에는 감소시킴으로서 실제의 위험이 반영된 CVSS를 산정할 수 있다.

이는 SIEM이 Alert을 하는 방법 중에 Risk Scoring 방법을 개선시킬 수 있다.

예를 들어 Table 8과 같이 HTTP헤더의 Content-Type 항목 값을 변조가 가능한 Apache 취약점 CVE-2014-3581 취약점의 CVSS가 7.0을 산정 되었으며, SIEM에서 8.0 이상에 대하여 Alert

Table 8. CVSS applied Vulnerability

CVSS	Vulnerability exist		Vulnerability non-exist	
	Weighting	Modified CVSS	Weighting	Modified CVSS
7	1.5	10.5	0.7	4.9

을 발생시키도록 설정되었을 경우 실제 취약점이 있는 자산에 대해서만 Alert을 발생시킬 수 있다.

가중치를 취약성이 있을 경우와 없는 경우에 각각 1.5와 0.7로 적용했을 경우 실제 자산에 소프트웨어 취약점이 있는 경우에만 Alert이 발생하고, Alert 임계치인 8.0 이하인 7.0의 취약점도 탐지할 수 있게 된다.

Risk Scoring 방식 외에 Filter와 연계된 Rules을 이용한 Alert 방식에도 적용할 수 있다. Zero-day와 같이 긴급하게 탐지해야 하는 이벤트의 경우와 신규 Worm의 등장으로 글로벌하게 특정 취약점을 이용한 공격이 급증하는 경우가 있을 수 있다. 자산에 해당 소프트웨어 취약점이 있고, 해당 취약점을 이용한 모든 공격에 대하여 Alert이 필요한 경우에는 보안장비의 시그니처와 CVE정보를 연계에서 탐지할 수 있다. 이를 위해서는 사전에 보안장비의 시그니처와 CVE 정보를 Table 9와 같이 DB에 저장해야 한다. 앞서 해당 내용을 조사해본 결과 대다수의 보안장비 벤더에서는 시그니처를 배포하면서 관련 CVE정보를 제공해 주고 있지만, 이벤트 탐지로그에는 해당 CVE정보를 제공해 주지 않고 있었다.

따라서 SIEM의 DB에 시그니처와 CVE정보의 매핑정보를 수집하고 해당 탐지로그에서 시그니처명과

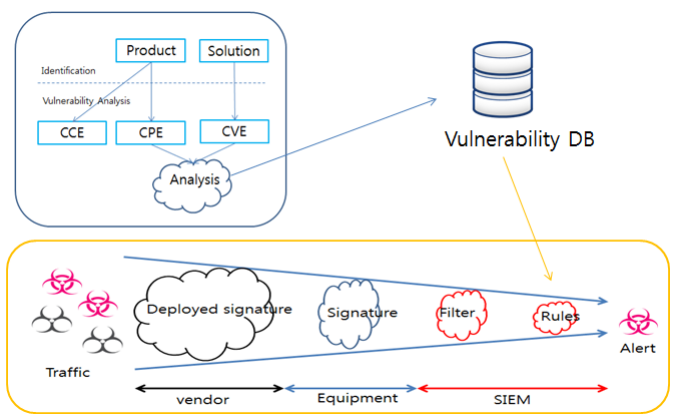

Fig. 6. Software vulnerability detection model

Table 9. Sample of CVE Mapping Table

Signature NAME	CVE
BIND denial-of-service attack	CVE-2012-5166
CompoundFile_Nested_SWF	CVE-2011-0609
Symantec_AMS_Component_Overflow	CVE-2011-0110
MHTML_Handler_Detected	CVE-2011-0096
MHTML_Script_Detected	CVE-2011-0096
Kerberos_Weak_Encryption	CVE-2011-0091
HTTP_IE_rpawinet_DLL_Hijacking	CVE-2011-0038
SMB_IE_rpawinet_DLL_Hijacking	CVE-2011-0038

취약점 DB의 CVE정보의 비교가 필요하다.

이 모델이 소프트웨어 취약점 탐지 및 대응에 효과적인 이유는 Risk Scoring 팀지 방법에서는 취약점의 영향도에 가중치를 줌으로서 고정된 소프트웨어 취약점의 영향도만으로 탐지하는 것이 아닌 실제의 영향도를 반영할 수 있다는 것이다. 또한 Rules 기반 탐지에서는 취약점 진단에 의해 특정 소프트웨어 취약점이 발견되었으나 서비스 가용성이나 안정성을 이유로 패치를 진행하지 못하는 경우에 해당 소프트웨어 취약점에 대한 이벤트를 탐지할 수 있다는 것에 있다.

서비스 환경에서는 취약점 진단에 의해 여러 가지 소프트웨어 취약점이 발견되어도 각 소프트웨어간의 호환성 문제나 패치에 대한 안정성을 이유로 패치를 진행하지 못하는 경우가 많다.

때문에 패치하지 못한 소프트웨어 취약점을 관리하고 탐지하는 부분에 있어서 매우 효과적일 수 있다.

3.2. 소프트웨어 취약점 탐지 모델 적용

본 논문에서 제안한 소프트웨어 취약점 탐지 모델의 일부를 실제 서비스에서 적용해 보았다. 적용범위는 Rules 기반의 탐지에서 취약점 DB에서 CVE를 기준으로 확인된 소프트웨어 취약점이 있을 경우 Alert을 발생시키도록 하였다. 이로 인하여 기존에 탐지하지 못한 소프트웨어 취약점을 탐지할 수 있는지 여부와 해당 모델을 사용하지 않고 해당 소프트웨어 취약점을 탐지하려고 하였을 경우 발생되는 Alert의 양을 파악하여 효율적인지를 보고자 하였다.

먼저 취약점 진단을 통하여 취약점 DB에 IP, OS, 서비스, CVE, 취약점정보를 수집하였다. 취약점 진단도구에 따라 DB에 저장되는 형태가 서로 다르기 때문에 별도의 변환과정이 필요하다.

각 취약점 진단도구를 개발한 벤더에서 제공하는 정보를 기반으로 서로 다른 취약점 진단도구의 결과물을 동일한 형태의 표준으로 수집을 진행하였다.

이후 보안장비에서 탐지된 이벤트를 취약점 DB와 비교하기 위한 정규화를 진행하여야 한다. 대부분의 보안장

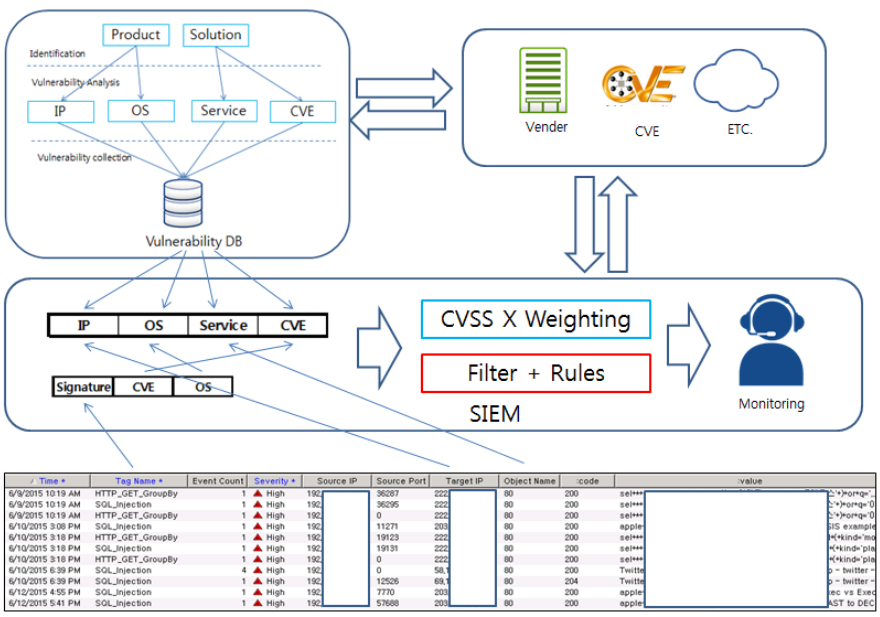

Fig. 7. Apply vulnerability DB to SIEM

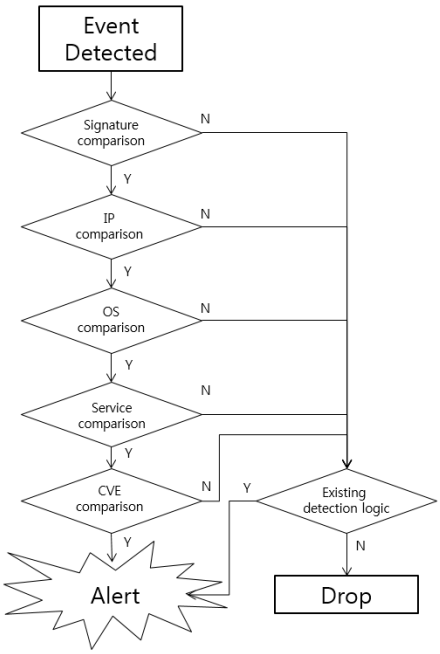

Fig. 8. Software vulnerability detection model diagram

비는 탐지로그에 CVE정보와 OS정보를 포함하지 않기 때문에, 벤더에서 제공하는 시그니처별 CVE 정보와 영향 받는 OS 정보와 연동이 필요하다.

기존의 SIEM에서 사용하던 탐지로직에 추가된 로직은 이벤트가 수신되면, 별도의 시그니처-CVE-OS가 저장된 DB에서 탐지된 이벤트의 시그니처가 있는지를 확인한다. 시그니처와 CVE가 연결되지 않는 시그니처도 다수 있기 때문에 DB부하를 줄이기 위해서는 시그니처에 대한 비교를 먼저 진행하게 된다.

CVE정보가 있는 이벤트가 탐지되면 취약점 DB의 IP정보와 탐지 이벤트의 목적지 IP를 비교하게 된다. 새로 추가되어 취약점 진단이 진행되기 전인 자산은 기존의 SIEM의 탐지로직으로 진행하게 된다. 새로 추가된 자산이 아니더라도 특정 이벤트(악성코드 유포, 웹쉘, 등)는 출발지 IP와 목적지 IP가 반대로 탐지되는 경우가 있으므로, 취약점 진단을 진행하였다 하더라도 취약점 DB에 정보가 없는 경우가 일부 있을 수 있다.

많은 수의 위협이 OS취약점을 이용하고 있기 때문에 취약점 진단에서 수집된 시그니처 정보와 연결된 OS정보를 비교하여 해당 위협이 실제 위험으로 영향을 줄 수 있는지 여부를 파악한다.

취약점 DB의 서비스와 탐지된 이벤트의 서비스(목적지 Port)를 비교하게 된다. 목적지 IP와 동일하게 출발지, 목적지가 반대로 탐지되는 경우와 오탐에 의해 사용하지 않는 서비스에서 탐지가 발생할 수 있다.

그 이후에 탐지된 이벤트의 소프트웨어 취약점이 해당 자산에 있는지 여부를 확인한다. Rules 기반의 탐지에서는 CVE 취약점이 있을 경우 모니터링 요원에게 Alert을 발생시키고, Rules로 설정되어 있지 않은 이벤트인 경우 사전에 정의된 가중치를 CVSS에 반영하여 기존의 SIEM 로직을 적용하게 된다.

제안 된 모델은 기존의 탐지프로세스를 변경하는 것이 아니고 새로 추가되는 개념이다. 따라서 각 요소에서 일치하지 않으면 해당 이벤트는 Drop 되는 것이 아니고, 기존에 운영하고 있는 프로세스를 따르게 된다. 해당 자산에 소프트웨어 취약점이 없다 하더라도 추가적인 다른 위협에 의해 침해사고가 발생할 수 있고, 보안장비가 모든 위협에 대하여 탐지하는 것이 아니기 때문에 취약점 진단을 통해 해당 자산에 소프트웨어 취약점이 없는 것으로 확인 되었다 하더라도, 기존의 SIEM 프로세스에 의해 Alert이 발생되어야 하는 이벤트는 그대로 발생되어야 한다.

위 프로세스를 운영중인 SIEM에 적용하여 기존에 Rules이 설정되어 있지 않은 이벤트가 발생됨을 확인하였다.

탐지된 이벤트는 Fig 10과 같이 『HTTPS Apache ClearText DoS』로서 해당 이벤트는 443 포트로 전송되는 암호화 되지 않은 HTTP Requset를 탐지 하며, 아파치 웹서버의 유효하지 않은 Response 값을 전송 또는 중지 시킬 수 있는 공격기법이다.

해당 위협은 CVE-2005-3357 취약점을 이용하는 것이며, Apache HTTP Server 2.0.55 이하 버전에서 취약점이 있고, CVSS 점수는 3.5이다.[7][25]

해당 이벤트는 위험등급이 낮은 편이고 다수의 이벤

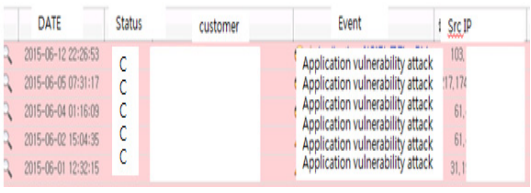

Fig. 9. Software vulnerability detection event

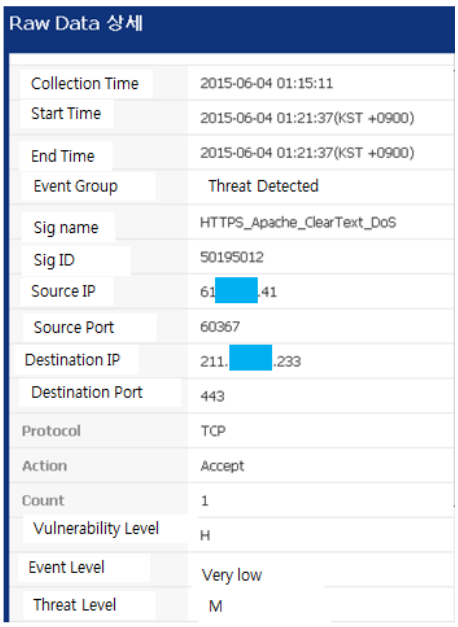

Fig. 10. Event more infomation

트가 발생하기 때문에 SIEM에서 Alert 설정하지 않는 이벤트 이다. 하지만 대상 자산의 소프트웨어 버전이 Apache HTTP Server 2.0.40 으로서 취약한 버전을 사용하고 있다. 이벤트 자체의 영향은 낮지만 취약한 버전을 사용하고 있는 자산에 해당 자산의 취약점을 악용할 수 있는 이벤트가 발생한 것이다.

해당 이벤트가 발생한 일자에 HTTPS Apache ClearText DoS 이벤트는 11개 자산을 대상으로 1,828EA의 이벤트가 발생하였다.

해당 이벤트는 영향도가 낮기 때문에 일반적으로 Rules을 생성해서 Alert을 발생시키지 않지만, CVE-2005-3357 취약점에 대응하기 위해 Rules을 생성하였다면 1,828EA의 Alert이 발생하게 된다.

취약점 DB를 이용하여 실제 취약점이 있는 자산에 대해서만 Alert을 발생시킴으로서 다수의 이벤트를 발생시키지 않고, 실제 취약성이 있는 자산에 위협이 발생한 1EA의 Alert을 발생시킬 수 있는 것이다.

모니터링 시스템에서의 오탐(False-Positive)과 미탐(False-Negative)의 문제는 매우 민감한 문제이다. 오탐을 줄이면 미탐이 증가하기 때문에 오탐과 미탐 사이에서 적절한 조절이 필요하다. 소프트웨어 취약점 탐지 모델은 기존의 SIEM에서 탐지하지 못했던 미탐 문제를 개선하는 모델이다. 미탐이 감소

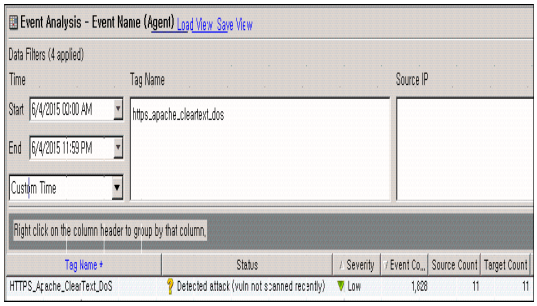

Fig. 11. Security solutions detection event

한다는 뜻은 오탐이 증가할 수 있다는 의미이다. 미탐이 감소된다 하더라도 오탐이 기존의 모델과 비슷한 수준으로 증가하거나 오히려 더 높은 수준으로 증가한다면 해당 모델은 실서비스에 적용될 수 없다.

Table 10. The proposed rules and Existing rules Compare

Rules	Attack Type	Items	Value
The proposed rules	***Software vulnerability Attack***	***Alert***	***124***
		True positive	***60***
		False positive	***64***
		True positive rate	***48.39 %***
Existing rules	**Total**	**Alert**	**42,697**
		True positive	**1,649**
		False positive	**41,048**
		True positive rate	**3.86 %**
	Web-based Attack	Alert	23,319
		True positive	843
		False positive	22,476
		True positive rate	3.62 %
	Scanning	Alert	11,440
		True positive	269
		False positive	11,171
		True positive rate	2.35 %
	BruteForce& Dictionary Attack	Alert	9,042
		True positive	41
		False positive	9,001
		True positive rate	0.45 %

	DDoS(Anomaly Traffic)	Alert	4,416
		True positive	193
		False positive	4,223
		True positive rate	4.37 %
	Application Vulnerability	Alert	2,547
		True positive	155
		False positive	2,392
		True positive rate	6.09 %
	Local System Attack	Alert	1,486
		True positive	29
		False positive	1,457
		True positive rate	1.95 %
	Internal Resources Attack	Alert	1,353
		True positive	7
		False positive	1,346
		True positive rate	0.52 %
	Malware	Alert	1,112
		True positive	35
		False positive	1,077
		True positive rate	3.15 %
	Networks Attack	Alert	411
		True positive	13
		False positive	398
		True positive rate	3.16 %
	Etc.	Alert	513
		True positive	64
		False positive	449
		True positive rate	12.48 %

실서비스에 적용 후 한달간 데이터 분석을 해본 결과 Table 10과 같이 기존의 시스템에서 탐지하지 못한 124건을 탐지한 것을 확인하였다. 124건중 48%인 60건이 실제 위협이벤트였으며, 이는 60건의 미탐을 추가로 발견했음을 의미한다. 소프트웨어 취약점의 탐지가 개선된 것이다. 기존의 10대 공격유형에 대한 이벤트의 정탐율은 0.52%~12.48%로서 평균 3.86% 이다. 소프트웨어 취약점 탐지 모델은 기존의 탐지방법 대비 10배가 넘는 정탐율을 나타내고 있다. 물론 취약점 진단결과를 기반으로 소프트웨어 취약점에 대해서만 추가로 탐지를 하기 때문에 기존의 Rules보다 정탐율이 높을 수 밖에 없다. 하지만 기존에 탐지하지 못했던 소프트웨어 취약점을 추가적으로 탐지하면서도 오탐율을 높이지 않는다는 것이 입증되었다.

따라서 본 논문에서 제안한 소프트웨어 취약점 탐지 모델은 소프트웨어 취약점을 효과적으로 탐지하는 것으로 확인 하였다.

IV. 결 론

본 논문에서 SIEM을 이용한 소프트웨어 취약점을 효과적으로 탐지하고 대응할 수 있는 모델을 제안해 보았다. 자산이 가지고 있는 취약점에 대한 위협을 탐지하는 **Risk = Vulnerability + Threat**의 개념에 충실한 모델이다. 다만 보안시스템에서 탐지하는 이벤트 자체가 오탐이 있기 때문에, 취약점 진단을 통해 소프트웨어 취약점을 알고 있고, 해당 소프트웨어 취약점을 탐지하기 위한 이벤트가 발생 하였다 하더라도 그 이벤트 자체가 오탐이면 무의미한 Alert이 발생하게 된다. 오탐과 미탐의 문제는 보안시스템이 가지고 있는 기본적인 문제로서 기존의 오탐과 미탐의 문제는 개선되지 않는다.

제안한 모델의 효과를 극대화하기 위해서는 취약점 진단 단계의 정보와 보안시스템의 정보가 동일한 표준으로 상호 공유가 가능해야 한다. 하지만 현재 CVE를 제외하고는 동일한 표준으로 전송되지 않기 때문에 많은 수작업을 동반하고 있다. 또한 동일한 표준을 쓰지 않음으로서 실시간으로 동기화 되지 않기 때문에 지속적으로 취약점 DB를 관리해야 한다.

향후 연구과제로서 NVD가 활성화 될 수 있도록 각 요소에 대한 활용방안에 대한 연구가 필요하다. 추가적으로 논문의 모델은 취약점 진단에서 수집 된 정보를 소프트웨어 취약점을 탐지하는데 활용한 것이고, 보안시스템의 설정을 Customizing 하고 관리하는 방안에 대한 연구도 필요하다.

References

[1] HP, http://www8.hp.com/us/en/software-solutions/siem-security-information-event-management/index.html

〔2〕 Sangyong Choe, "Reconstruction of the hacking incident," Acorn
〔3〕 Myeonghun Gang, "Completion of IDS and security control seen as a big data analysis," Wowbooks
〔4〕 "Guidelines for Information Security Meas ures", (KoreaCommunicationsCommissio n 2013-3, 2013.01.17)
〔5〕 Dongjin Gim, Seongje Jo, "An Analysis of Domestic and Foreign Security Vulnerability Management Systems based on a National Vulnerability Database," 1(2), pp. 3-5, Nov 2010
〔6〕 Huijin Jang, "Comprehensive analysis system for intrusion detection and response," Agency for Defense Devlopment, pp. 16-19
〔7〕 IBM, https://exchange.xforce.ibmcloud.com/vulnerabilities/24008
〔8〕 ITU-T Q.4/17 Proposed initial draft text for Rec. ITU-T X.cybex, Cybersecurity information exchange framework (TD503)
〔9〕 "Requirements for Distribution and Sharing of Information in the Vulnerability DB", (Technical Report), TTAR-12.0016, Telecommunications Technology Association, pp 9-10, Dec 2012
〔10〕 Microsoft, https://technet.microsoft.com/ko-kr/library/security/ms15-001
〔11〕 Adobe, https://helpx.adobe.com/security/products/flash-player/apsb15-06.html
〔12〕 Oracle, http://www.oracle.com/technetwork/topics/security/javacpujun2013-1899847.html
〔13〕 PHP Group, http://php.net/
〔14〕 Apache Software Foundation, http://www.apache.org/dist/httpd/Announcement2.4.txt
〔15〕 Seongjin An, I Gyeongho, Bak Wonhyeong, "Security Monitoring&Control,EHANMEDIA," pp. 16-55, Apr 2014
〔16〕 Young-Jin Kim, Su-yeon Lee, Hun-Yeong Kwon, Jong-in Lim, "A Study on the Improvement of Effectiveness in National Cyber Security Monitoring and Control Services," korea institute of information security and cryptology, pp. 2-3, Feb 2009
〔17〕 Si-Jang Park, Jong-Hoon Park, "Current Status and Analysis of Domestic Security Monitoring Systems, korea institute of electronic communication science," pp. 2-3, Sep 2014
〔18〕 IBM, http://xforce.iss.net/ContentUpdates.do;jsessionid=2C5B979DC4827A7EAD8F254F587B9A44?xpu=75
〔19〕 Paloaltonetworks, https://downloads.paloaltonetworks.com/content/app-502-2736.html?__gda__=1433931570_54a6cb5825a3c7748542dcb09f1a616f
〔20〕 Wins, https://sniper2.wins21.com/pattern_update/SKRE2CWIS207528/help/h_1300_05894.html
〔21〕 HP, http://www8.hp.com/kr/ko/business-solutions/security-overview.html
〔22〕 Ji Hong Kim, Huy Kang Kim, "Automated Attack Path Enumeration Method based on System Vulnerabilities Analysis," korea institute of information security and cryptology, pp. 3-4, Oct 2012
〔23〕 "A Study on Construction of A vulnerability Management System for New Information Technologies," KISA-WP-2010-0018, pp. 36, Aug 2010
〔24〕 Gim Gyeonggi, "Research of improved CVSS for vulnerability management in financial ISAC," pp. 27, Jun2008
〔25〕 MITRE, https://cve.mitre.org/index.html
〔26〕 MITRE, https://cce.mitre.org/
〔27〕 MITRE, https://cpe.mitre.org/
〔28〕 Frst, https://www.first.org/cvss
〔29〕 NIST, http://scap.nist.gov/specifications/xccdf/
〔30〕 MITRE, https://oval.mitre.org/

〈저자소개〉

전 인 석 (In-seok Jeon) 종신회원
2009년 8월: 건국대학교 정보통신대학원 정보보호학과 석사
2014년 9월: 고려대학교 정보보호대학원 정보보호학과 박사 과정
2009년 9월~현재: Ahnlab CERT팀 주임 연구원
〈관심분야〉 네트워크보안, 정보보호관리체계, 정형기법, SIEM, SSCA, MMS, 스마트의료보안 등

한 근 희 (Keun-Hee Han) 종신회원
서울과학기술대학교 컴퓨터공학과 졸업
한양대학교 공학대학원 공학석사
고려대학교 대학원 이학박사
현재: 고려대학교 융합소프트웨어전문대학원 산학교수
〈관심분야〉 소프트웨어 보증, 시큐어 코딩, 정보보호관리 체계, 개인정보보호, 클라우드 컴퓨팅 보안, 스마트 의료 보안, 스마트 자동차 보안 등

김 동 원 (Dong-Won Kim) 종신회원
2009년 2월: 서울과학기술대학교 컴퓨터공학과 졸업
2012년 2월: 건국대학교 정보통신대학원 정보보호학과 석사
2014년 2월: 고려대학교 정보보호대학원 정보보호학과 박사 수료
2014년 2월: 현대오토에버 정보보안기술팀 과장
2014년 3월~현재: 서울호서전문학교 사이버해킹보안과 전임교수
〈관심분야〉 시큐어코딩, 정보보호, 모바일 보안, 지능형 차량 보안, SSCA, 정형기법 등

최 진 영 (Jin-Young Choi) 종신회원
1982년 서울대학교 컴퓨터공학과 (학사)
1986년 미국 Drexel University, Dept. of Mathematics and Computer Science (석사)
1993년 미국 Univ. of Pennsylvania, Dept. of Computer and Information Science (박사)
1996년~현재 고려대학교 컴퓨터-전파통신공학부 교수
〈관심분야〉정형기법, 임베디드 실시간시스템, 프로그래밍언어, 프로세스 대수, 소프트웨어 공학