Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.5, Oct. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.5.1009

자연어 처리 기반 멀티 소스 이벤트 로그의 보안 심각도 다중 클래스 분류

서 양 진†‡
이포즌 (연구소장)

A Multiclass Classification of the Security Severity Level of Multi-Source Event Log Based on Natural Language Processing

Yangjin Seo†‡
EPOZEN Co., Ltd. (Head of R&D)

요 약

로그 데이터는 정보 시스템의 주요 동작과 상태를 이해하고 판단하는 근거로 사용되어 왔으며, 여러 보안 분야 응용에서도 중요한 입력 데이터로 사용된다. 로그 데이터로부터 필요한 정보를 얻어 이를 근거로 의사 결정을 하고, 적절한 대응 방안을 취하는 것은 시스템을 보호하고 안정적으로 운영하는 데 있어 필수적인 요소이지만, 로그의 종류와 양이 폭발적으로 증가함에 따라 기존 도구들로는 효과적이고 효율적인 대응이 쉽지 않은 상황이다. 이에 본 연구에서는 자연어 처리 기반의 머신 러닝을 이용해 멀티 소스 이벤트 로그의 보안 심각도를 여러 단계로 분류하는 방법을 제안하였으며, 472,972건의 훈련 및 테스트 샘플을 이용하여 실험을 수행한 결과 99.59%의 정확도를 달성하였다.

ABSTRACT

Log data has been used as a basis in understanding and deciding the main functions and state of information systems. It has also been used as an important input for the various applications in cybersecurity. It is an essential part to get necessary information from log data, to make a decision with the information, and to take a suitable countermeasure according to the information for protecting and operating systems in stability and reliability, but due to the explosive increase of various types and amounts of log, it is quite challenging to effectively and efficiently deal with the problem using existing tools. Therefore, this study has suggested a multiclass classification of the security severity level of multi-source event log using machine learning based on natural language processing. The experimental results with the training and test samples of 472,972 show that our approach has archived the accuracy of 99.59%.

Keywords: Security Event Log, Natural Language Processing, Multi-source, Multiclass Classification, Cybersecurity

I. 서 론

정보 시스템을 보호하는 솔루션은 여러 데이터를 기반으로 위협을 탐지하는데, 그러한 데이터 중에서도 로그는 정보 시스템의 동작을 이해하고 그 상태를 판단하는 일에 있어 기본적이고 중요한 입력 데이터로 사용된다[1]. 예를 들어 특정 서버의 동작에 문제가 생겼을 때 이의 해결을 위해 운영자가 우선으로 확인하는 데이터 중에는 해당 서버가 남기는 로그가 포함되며, 해당 로그만으로 문제가 해결되지 않았을

Received(08. 22. 2022), Modified(08. 29. 2022), Accepted(08. 31. 2022)
† 주저자, research.yj.seo@gmail.com
‡ 교신저자, research.yj.seo@gmail.com(Corresponding author)

때 운영자는 관련 로그 예를 들면 운영체제가 남기는 로그를 확인한다. 로그 데이터는 이미 발생한 문제의 원인을 찾고 분석하는 용도로 사용될 뿐만 아니라 특정 문제의 발생 여부를 판단하거나 해당 문제가 발생할 가능성을 예측하는 근거 데이터로도 활용될 수 있다.

로그 데이터를 생산하는 주체에는 단일 서버에서도 운영체제와 여러 시스템 프로그램과 응용 프로그램이 있으며, 조직의 규모에 따라서는 수백 대의 서버와 네트워크 장비를 포함하는 많은 장비가 하루에 적게는 수만에서, 많게는 수천만 건의 로그 데이터를 생산해 내기에 로그를 분석하는 일은 숙련된 전문가에게도 많은 시간과 수고를 들여야 하는 일이 된다. 또한, 정보 시스템의 규모와 복잡도가 날로 증가함에 따라 로그 데이터의 양이 폭발적으로 증가하고 있어 로그 데이터로부터 필요한 정보를 찾아내는 작업은 기존 방법이나 도구들로는 감당하기 어려운 수준에 이르렀다[2].

로그 분석에는 전통적으로 통계, 확률, 패턴 매칭 기반의 도구들이 사용되어 왔으며, 분석 목적 또는 적용 분야에 맞는 도구가 선택되어 사용된다. 예를 들어 시스템 성능 모니터링을 위해 로그 분석을 할 때는 로그 데이터로부터 성능 지표가 될 수 있는 값을 추출할 것이며 이를 통계 기반의 도구를 이용해 분석할 것이다. 이러한 분석에는 운영자 또는 관리자가 정의한 규칙들이 함께 적용되는 경우가 일반적이다. 예를 들면 일정 시간 동안의 CPU 평균 사용률이 사용자가 정의한 임곗값 이상일 때 경보를 발생하는 규칙을 정의하고 적용할 수 있다.

본 연구는 로그 분석의 여러 적용 분야 중에서 로그 데이터에 포함된 이벤트 내용이 보안 측면에서 어떤 심각도(severity level)를 가지는지를 분류하는 응용을 다룬다. 해당 로그 분석에 사용될 수 있는 가장 간단한 형태의 도구는 키워드 매칭이다. 특정 단어의 포함 여부나 특정 단어 조합의 존재 여부에 따라 심각도를 판단하는 방법으로, 예를 들면 로그에 "error"라는 단어가 들어가 있으면 심각도 2로, "fail"이 들어가 있으면 심각도 3으로 분류되도록 정의하는 방식이다. Fig.1.은 이벤트 로그의 예인데, 앞선 정의에 따르면 Log #2에는 심각도 3이, Log #3에는 심각도 2가 부여된다. Fig.1.에서 〔Log #1〕처럼 굵은 글씨로 표시한 부분은 로그의 식별이 쉽도록 저자가 추가한 것으로 실제 로그에는 해당 항목이 없다.

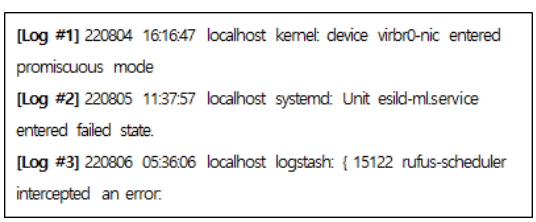


Fig. 1. Security event log example

이벤트 로그 보안 심각도 분류에 사용될 수 있는 또 다른 도구로는 패턴 매칭이 있다. 패턴 매칭을 이용하면 키워드 매칭보다 더 유연하면서도 복잡한 조건으로 로그를 분석하는 것이 가능해진다. 대표적인 패턴 매칭 도구로는 정규식이 있으며, 패턴 매칭을 기반으로 하여 로그 분석을 수행하는 도구에는 로그 파서[3]와 로그 클러스터링[4]이 있다. 엄밀하게 말해 로그 파서와 로그 클러스터링은 로그 분석 작업 흐름 중 앞쪽에 위치하는 요소로서, 로그 파서는 비정형(unstructured) 로그 데이터를 정형 데이터로 바꾸어 주는 기능을, 로그 클러스터링은 다양한 로그를 그룹화해 주는 기능을 수행하여 최종적인 로그 분석 작업에 도움을 준다.

Fig.2.는 Fig.1.의 로그에 대해 대표적인 로그 파서인 Drain[5]을 적용한 예이다. 로그 파싱에서 정규식은 로그 문자열 내의 특정 패턴을 찾아내는 기능으로 사용될 수 있으며, 로그 문자열 내의 고정 부분과 변동되는 부분을 구분하는 처리에 도움을 준다. Log #3에 파서를 적용한 결과를 보면 15122라는 숫자가 〈*〉으로 바뀌어 있는데, 이는 숫자 패턴에 대한 정규식 정의에 따른 것이다. Drain의 적용 시 로그 파싱 결과를 어떤 포맷으로 출력할지를 사용자가 정의할 수 있으며 Fig.2.의 예에서는 "〈Date〉 〈Time〉 〈Server〉 〈Component〉: 〈Content〉"로

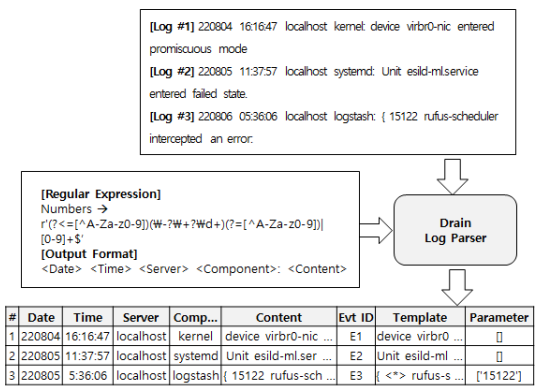


Fig. 2. Event log parsing example

정의하였다. 하단에 있는 파싱 결과 뒤쪽 세 개의 열을 보면 이벤트 식별자(Evt ID), 이벤트 템플릿(Template), 파라미터(Parameter) 항목을 확인할 수 있다. 파서는 입력으로 들어온 로그의 고정 부분과 변동 부분을 구분하여 고정 부분에 해당하는 이벤트 템플릿과 변동 부분에 대한 파라미터 목록을 추출한다. 로그 클러스터링은 로그 파싱이 이벤트 템플릿을 추출하는 것과 유사한 방식으로 로그 그룹을 분리하여 로그 분석을 돕는다.

로그 파싱이나 로그 클러스터링에서 사용되는 템플릿은 유사하거나 같은 종류의 로그를 구분하는 작업에서는 효과적인 도구가 되지만, 각 로그 메시지의 보안 심각도를 판단하는 작업에서는 명확한 한계를 보이는데, 이는 보안 템플릿 만으로는 보안 심각도를 판단할 수 없는 경우가 존재하기 때문이다. Fig.3.의 두 로그 메시지는 다른 보안 심각도를 가지는 데이터이지만 이벤트 템플릿에 따르면 같은 종류의 로그로 분류될 가능성이 크다. 다시 말해 템플릿에 기반한 보안 심각도 분류로는 파라미터의 내용까지 확인해야 하는 경우나 고정 부분은 유사하지만 다른 보안 심각도를 가지는 로그들이 존재하는 경우를 효과적으로 지원하지 못한다. 또한, 로그 파싱이나 클러스터링은 보통 한 종류의 로그 파일을 처리하도록 고안되어 있기에 여러 파일에서 온 멀티 소스 로그 데이터에 대해서는 기능이 제대로 동작하지 못한다[6][7]. 하나의 조직안에서 발생하는 다양한 로그 파일 각각에 대해 로그 파싱이나 클러스터링을 수행한 후 보안 심각도 분류를 수행하는 것도 가능한 일이겠으나 다양한 로그 파일에 대해 통합된 분류를 수행하는 것이 더 효율적인 작업이 될 것임은 자명하다. 이에 본 연구는 멀티 소스 이벤트 로그의 보안 심각도를 분류하는 새로운 도구로서 자연어 처리 기반의 방법을 제안하였다.

자연어 처리 기반의 방법을 사용하면 비정형 로그 원본 데이터를 정형 데이터로 바꿀 필요 없이 로그 분석을 수행하는 것이 가능하다. 자연어 처리 기반의 방법을 사용해도 로그 데이터를 컴퓨터 알고리즘이 처리할 수 있는 데이터 표현(representation)으로 바꾸어 주는 과정은 여전히 필요하지만, 여러 종류의 로그에 대해, 각 로그 종류별로 추가적인 작업 없이 해당 과정을 수행할 수 있기에 멀티 소스 이벤트 로그를 분석하는 과업에 있어서는 자연어 처리 기반의 방법이 전통적인 파싱이나 클러스터링 보다 효율적이다. 자연어 처리 기반의 방법에서 사용하는 데이터 표현은 머신 러닝이나 딥러닝 모델과 함께 적용돼 사람이 인지하거나 정의하기 어려운 로그 데이터 내의 복잡한 관계를 모델링할 수 있어 더욱 효과적인 로그 분석이 가능해진다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 본 연구가 다루고자 하는 문제와 연구에 사용된 데이터셋 및 보안과 관련된 이벤트 로그 특성에 관해 자세하게 설명한다. 3장은 대표적인 머신 러닝 분류 모델인 Random Forest를 이용해 멀티 소스 이벤트 로그의 보안 심각도 다중 클래스 분류를 수행하고 그 결과를 분석한다. 마지막으로 4장에서는 본 연구의 결론과 향후 연구에 대해 정리한다.

[Log #4] level : 10, log : fru-name: error-message
[Log #5] level : 3, log : fru-name: state:[state] error-message

Fig. 3. Log messages of same template having different security levels each

II. 멀티 소스 이벤트 로그의 보안 심각도 분류

2.1 다중 클래스 분류

본 연구는 자연어 처리 기반의 머신 러닝을 이용해 여러 포맷의 이벤트 로그 데이터의 각 이벤트에 대해 보안 심각도를 다중 클래스로 분류하는 작업에 대한 것이다. 보안 심각도를 판단하는 단위는 로그 레코드 하나이다. 로그 데이터로부터 보안과 관련된 이벤트를 추출할 때 하나의 이벤트가 여러 로그 레코드에 걸쳐서 기록되는 때도 있겠지만 본 연구는 개별 로그 레코드에 대해 보안 심각도를 판단하는 작업을 다룬다. 하나의 로그 이벤트가 Fig.4.의 Log #8처럼 두 줄 이상으로 기록되는 때도 있는데 이러한 경우도 하나의 레코드이다. 하나의 조직에서 생산되는 여러 로그 중 어떤 것이 보안과 관련된 로그인가도 분류가 필요한 문제이지만 본 연구에서는 분류기 모델의 학습이나 테스트의 입력으로 들어오는 모든 로그가 보안과 관련된 것이라 가정하고 해당 로그의 보안 심각도를 분류한다. 보안 분야 데이터에 머신 러닝 또는 딥러닝을 적용하는 응용에 있어 최근 주목받고 있는 기법의 하나가 비지도 학습 기반 이상 탐지인데[8][9], 본 연구는 지도학습 기반으로 보안 심각도를 3개 이상의 다중 클래스로 분류하는 문제를 다룬다. 보안 분야에서 이상 탐지가 충분히 또는 더

Fig. 4. Event log example with a long line or multiple lines and complex structure

효과적인 도구가 되는 문제들이 있지만, 이벤트 로그의 보안 심각도를 여러 단계로 자동 분류할 수 있다면 관리자 또는 운영자가 보안 위협에 대한 체계적인 대응 전략을 세우는 것이 가능해진다.

2.2 데이터셋

본 연구에는 데이콘이 주관한 "로그 분석을 통한 보안 위험도 예측 AI 경진대회"에 사용된 데이터셋(이하 데이콘 로그 데이터셋)을 이용하였다[10]. 데이콘 로그 데이터셋은 Kibana나 Logstash 같은 주요 정보 시스템의 동작 로그, Linux와 Windows 같은 운영체제 감사(audit) 로그, Syslog 같은 시스템 이벤트 로그, 보안 솔루션 관련 로그 등 보안과 관련된 다양한 로그들을 포함하고 있으며, 472,972건의 학습 샘플과 1,418,916건의 테스트 샘플로 구성되어 있는데, 테스트 샘플의 경우 경진대회의 특성상 정답(레이블)이 공개되어 있지 않다. 테스트 샘플에 대한 추론 결과를 데이콘 사이트에 업로드하여 성능을 일정 부분 확인하는 것은 가능하나 이에 대한 분석이 불가능하여 본 연구에는 학습 샘플을 나누어

Table 1. Samples' level distribution

level	Instances #
0	334065
1	132517
2	12
3	4141
4	10
5	2219
6	8

학습과 테스트에 이용하였다. 샘플의 레이블(클래스)은 0부터 6까지 총 7단계로 부여되어 있으며 큰 숫자일수록 위협이 큰 것을 의미한다. 각 로그에 어떤 기준과 방법을 적용해 레이블을 부여했는지는 공개되어 있지 않다. Table 1.은 472,972건 전체 샘플의 레이블별 분포를 보여주는데, 0과 1에 해당하는 샘플이 전체건수의 98.6%에 이른다. 어떤 조직에서 발생하는 전체 이벤트 중 비정상 이벤트가 드물다[11][12]는 사실을 고려할 때 데이콘 로그 데이터셋은 현장 상황에 부합하는 특성을 가지는 것으로 판단된다.

2.3 보안 관련 이벤트 로그의 특성 및 전처리

로그 데이터는 작성자의 필요와 목적에 따라 생성하여 기록하는 문자열로서 일정한 포맷을 가지는, 토큰들의 나열이다. 그러한 토큰에는 1) "error"나 "fail"처럼 사람이 일상에서 사용하는 단어, 2) 사람이 일상에서 사용하는 단어와 유사한 것(예를 들면 변수 이름), 3) 일상에서 사용되는 것은 아니지만 사람이 정의한 일정 형태를 가지는 것(예를 들면 URL 주소), 4) 특정 함수 및 입력값에 따라 생성되는 임의의 문자열 5) ppid=86518와 같은 키와 값 쌍 등이 있다. 보안 관련 이벤트 로그는 긴 임의의 문자열과 복잡한 내용을 포함하는 경우가 있어 파싱의 난이도가 높은 편이다. 앞에서 본 Fig.4.는 그러한 로그의 예이며, Drain 파서를 이용해 로그 파싱을 수행한 결과 파싱에 실패하거나 의미 있는 결과를 출력하지 못했다. 복잡한 규칙을 여러 개 추가함으로써 파싱 품질을 일정 수준 개선하는 것은 가능한 일이지만 여러 종류의 로그 각각에 대해 그러한 작업을 수행하는 수고가 있어야 하며 그러한 노력에도 불구하고 의미 있는 결과를 얻기가 쉽지 않다.

본 연구는 이러한 문제에 사용될 수 있는 또 다른 도구로서 자연어 처리 기반의 방법을 사용한다. 자연어 처리 기반으로 로그 분석을 수행할 때 문자열 데이터를 컴퓨터가 처리할 수 있는 데이터 형태로 변환하는 대표적이고 기본적인 방법은 토큰화 및 벡터화이다[13]. 토큰화(tokenization)는 문자열을 공백이나 쉼표 같은 구분자를 기준으로 토큰으로 나누어주는 것이고 벡터화(vectorization)는 그렇게 나누어진 토큰들을 정의된 방법에 따라 숫자로 표현해주는 것이다.

자연어 처리 기반의 로그 분석이라고 해서 이벤트 로그 원본 문자열 그대로를 사용할 필요는 없다. 로그를 분석할 필요와 목적에 부합하지 않는, 의미 있는 정보를 제공하지 않는 부분을 학습 데이터에 포함하는 것은 저장 공간과 처리 속도 측면에서 해가 될 것임이 분명하기에 해당 항목들을 제거해 주는 것이 좋다. 예를 들어 임의로 생성된 문자열은 이벤트 로그의 보안 심각도를 판단하는 데 있어 의미 있는 데이터가 아니므로 제거하는 것이 좋은 선택이다. 3장에서 수행할 실험에서 로그 데이터의 날짜, 시간, 인터넷 주소(IP address), 키와 값 쌍에서 값, 임의 생성 문자열이나 인코딩된 문자열, 숫자 항목은 제거해 주었다. 임의 생성 문자열이나 인코딩된 문자열이 이벤트 로그의 보안 심각도를 판단함에 있어 유용한 정보를 제공하는 경우는 없다고 판단되며, 해당 문자열 안에는 구두점(punctuation mark)이 포함되는 경우가 있어 이는 의미 없는 토큰들이 다수 생성되는 결과로 이어질 수 있어 제거해 주었다. 로그에서 시간 값은 중요한 정보이지만 본 연구에서는 데이터의 발생 순서에 따른 정보는 고려하지 않고 하나의 레코드 각각에 대해 보안 심각도 분류를 수행하므로 날짜와 시간을 제거하는 것이 문제가 되지 않는다. 데이터의 발생 순서를 고려하는 분석이라면 해당 값을 제거하지 않거나 해당 값을 제거하더라도 로그가 발생 순서에 맞게 입력으로 들어가도록 해야 한다. 나머지 항목들도 분석 목적에 따라 데이터에서 제거되거나 포함될 수 있다. Fig.5.는 이벤트 로그 데이터에서 임의의 문자열이 제거된 예이다. 가운데 상자 안에 있는 "--MARK--: 〈val〉" 항목은 이해를 돕기 위해 임의의 긴 문자열이 하나의 "값"으로 인식된 것임을 나타낸 것으로 실제로는 제일 아래에 있는 형태로 바로 제거 처리된다.

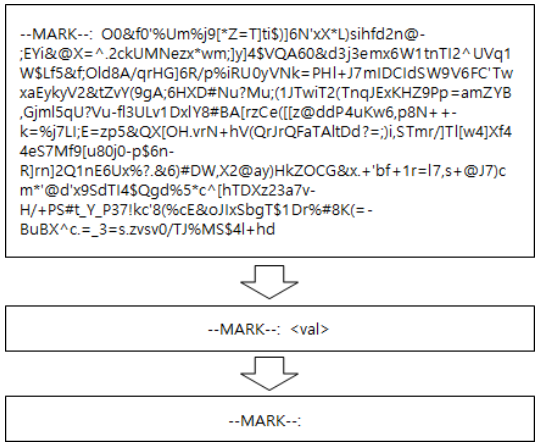
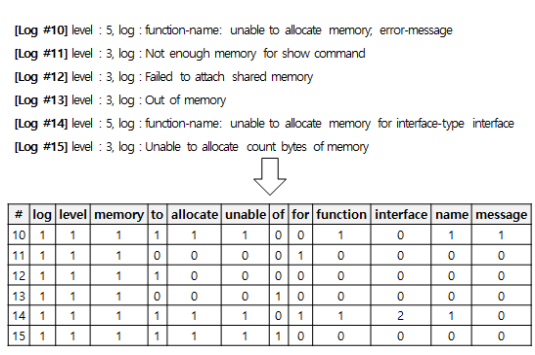


Fig. 5. Removal of the uninformative part in event log data

III. Random Forest 기반 멀티 소스 이벤트 로그의 심각도 다중 클래스 분류

3.1 Random Forest 모델

멀티 소스 이벤트 로그의 보안 심각도를 자연어 처리 기반으로 다중 클래스 분류하는 방법의 성능을 평가하기 위해 대표적인 머신 러닝 분류 및 회귀분석 모델인 Random Forest(이하 RF)를 사용해 실험을 수행하였다. RF는 앙상블 모델로서 훈련을 통해 여러 개의 의사 결정 트리를 생성하고 가장 많은 표를 얻은 레이블을 출력으로 내보낸다[14]. 본 실험에는 Python scikit-learn 라이브러리에서 제공하는 RF 분류기를 사용하였으며, 이벤트 로그의 토큰화와 벡터화에도 scikit-learn 라이브러리에서 제공하는 CountVectorizer(이하 CV)를 이용해



#	log	level	memory	to	allocate	unable	of	for	function	interface	name	message
10	1	1	1	1	1	1	0	0	1	0	1	1
11	1	1	1	0	0	0	0	1	0	0	0	0
12	1	1	1	1	0	0	0	0	0	0	0	0
13	1	1	1	0	0	0	1	0	0	0	0	0
14	1	1	1	1	1	1	0	1	1	2	1	0
15	1	1	1	1	1	1	1	0	0	0	0	0

Fig. 6. An example of tokenization and vectorization by scikit-learn CountVectorizer

Bag-of-Words[15]를 구성하였다. Fig.6.은 6건의 이벤트 로그에 대해 CV를 이용해 토큰화와 벡터화를 수행한 예로 전체 토큰 중 출현 빈도 상위 12개만 표시한 것이다. CV는 입력된 각 로그 레코드별로 각 토큰이 포함된 수를 행렬로 구성한다. 토큰화 시 2글자 이상의 알파벳 문자와 숫자로 구성된 것만을 대상으로 하며 모든 구두점은 토큰을 나누는 구분자로 사용된 후 제거된다.

3.2 실험 및 결과

실험에는 472,972건의 샘플 중 같은 내용의 로그에 두 개 이상의 다른 클래스가 부여된 경우를 찾아 제거한 후 남은 471,888건의 샘플을 절반씩 학습 데이터와 테스트 데이터로 사용하였다. 심각도 2, 4, 6의 이벤트 로그는 건수가 많지 않아 전체 샘플을 학습과 테스트 데이터로 절반씩 나눌 때 모든 클래스가 클래스별로 절반씩 학습과 테스트 데이터에 포함되도록 하였다. CV 적용 시 최대 토큰 수를 지정하게 되는데, 실험 1에서는 최대 토큰 수를 5,000으로 적용하여 실험을 수행하였다. 최대 토큰 수를 5,000으로 지정하면 전체 학습 샘플에 대해 토큰화 수행 후 출현 빈도 상위 토큰 5,000개를 추려내며, 이 목록을 이용하여 각 이벤트 로그의 벡터화를 수행한다. 따라서 최대 토큰 수를 5,000으로 지정하면 RF 분류기의 입력으로 235943 X 5000의 행렬이 주어진다. RF 모델에서 사용할 트리 수를 100으로 지정하고 학습을 수행한 후 테스트 데이터에 대해 추론을 수행한 결과 70.67%의 정확도를 달성하였는데, 이에 대한 혼동 행렬(confusion matrix)을 확인해 보면(Table 2) 사실상 분류가 제대로 이루어지지 않았음을 알 수 있다. 데이터셋에서 보안 심각도 레벨 0을 가지는 로그의 비율이 약 70%라는 사실과 혼동 행렬에 나타난 결과를 바탕으로 결과를 분석하면, 벡터화에 사용하는 출현 빈도 상위 토큰의 개수를 큰 값으로 정하면 전체 샘플 중 높은 비율을 가지는 클래스에 해당하는 로그에 속한 토큰들이 대거 사용되어 다른 클래스의 이벤트 로그까지도 해당 로그로 분류된 것으로 판단된다. 이를 확인하기 위해 실험 2와 3에서는 최대 토큰 수를 500과 200으로 제한하여 보안 심각도 분류를 수행하였고, 각각 정확도 99.59%와 98.94%를 달성하였다.

Table 3과 4는 실험 2와 3의 결과에 대한 혼동 행렬인데 정확도 상으로는 둘 사이에 큰 차이가 없지만, 상세 내용에는 주의 깊게 살펴볼 부분이 몇 가지 있다. 첫째, 전체 샘플 중 98.6%를 차지하고 있는 보안 심각도 0과 1을 제외한 샘플들의 분류 정확도를 보면 최대 토큰 수를 500으로 한 경우는 85.70%지만, 최대 토큰 수를 200을 한 경우는 54.91%이다. 사실 보안 심각도 2, 4, 6은 해당 건수가 매우 적어서 전체 정확도 계산에 미치는 영향이 미미할 수밖에 없기에 실험에 사용된 불균형 데이터셋에 대한 다중 클래스 분류의 성능 평가를 정확하게 하기는 쉽지 않다. 조직 내에서 발생하는 이벤트 로그 중 높은 보안 심각도를 가지는 데이터의 수가 적을 수밖에 없는 현실에서 실험 1, 2, 3의 결과는 멀티 소스 이벤트 로그를 대상으로 자연어 처리 기반, 구체적으로는 토큰 출현 빈도 기반으로 다중 클래스

Table 2. Confusion matrix for max token 5000

Real \ Predicted	0	1	2	3	4	5	6
0	166,730	0	0	0	0	0	0
1	66,039	2	0	0	0	0	0
2	6	0	0	0	0	0	0
3	2068	1	0	0	0	0	0
4	5	0	0	0	0	0	0
5	1087	0	0	3	0	0	0
6	4	0	0	0	0	0	0

Table 3. Confusion matrix for max token 500

Real \ Predicted	0	1	2	3	4	5	6
0	166,357	33	0	6	0	334	0
1	136	65,893	0	1	0	11	0
2	5	1	0	0	0	0	0
3	34	101	0	1861	0	73	0
4	2	0	0	3	0	0	0
5	85	3	0	143	0	859	0
6	0	1	0	0	0	3	0

Table 4. Confusion matrix for max token 200

Real \ Predicted	0	1	2	3	4	5	6
0	166,147	323	0	254	1	5	0
1	123	65,563	23	327	0	5	0
2	0	1	5	0	0	0	0
3	1102	3	0	890	0	74	0
4	0	0	0	0	5	0	0
5	20	12	0	218	0	840	0
6	0	1	0	0	0	0	3

분류를 수행할 때 최대 토큰 수의 선택이 분류 성능에 중요한 요소가 됨을 보여준다. 둘째는 보안 심각도 2, 4, 6을 가지는 이벤트 로그의 분류 결과이다. 최대 토큰 수를 500으로 한 경우는 해당 이벤트 로그를 여전히 한 건도 정확하게 분류해 내지 못하였지만, 최대 토큰 수를 200으로 한 경우는 좋은 분류 결과를 보였다. 해당 결과는 발생이 많지 않은 높은 보안 심각도의 이벤트 로그를 정확하게 분류하기 위한 목적이라면 벡터화에 사용될 최대 토큰 수를 적게 가져갈 필요가 있음을 보여준다.

실험 1, 2, 3에서는 하나의 이벤트 로그 상에 포함된 토큰들의 순서가 해당 이벤트의 보안 심각도 결정에 입력 정보로 사용되지 않았다. 로그 문자열에서 토큰의 위치가 다른 응용 예를 들면 번역 시 원본 문장에서 특정 토큰의 위칫값만큼의 정보를 제공하지는 않지만, 로그 분석 시 해당 정보를 고려하여 분류를 수행하면 성능에 어떤 변화가 있는지 확인할 필요가 있다. 본 연구에서는 각 이벤트 로그 레코드에 포함된 토큰들의 순서 정보를 보안 심각도 분류에 반영하는 방안으로써 바이그램[16]을 적용하였다. 바이그램은 두 개의 연속된 토큰을 묶은 것을 하나의 토큰처럼 취급하는 것으로 Fig.6의 Log #13에 대해서 "level log", "log out"과 같은 토큰을 추출하는 것이다. 바이그램이 하나의 이벤트 로그 문자열에 포함된 전체 토큰에서 각 토큰이 가지는 순서 정보를 나타내지 않지만 인접한 두 개의 토큰을 묶어 처리함으로써 순서 정보를 반영한다. 실험 1, 2, 3의 시나리오에 유니그램 대신 바이그램을 적용한 실험 4, 5, 6을 수행한 결과 각각 정확도 69.64, 99.15, 97.20%를 달성하여 오히려 저하된 성능을 보였으며, 혼동 행렬을 확인한 결과 상세 내용 측면에서도 저하된 결과를 보였다. 실험에 사용한 코드는 다른 연구자가 참고할 수 있도록 [17]에 공개하였다.

IV. 결 론

로그 분석은 그 목적과 적용 분야에 따라 사용할 도구를 선택하게 되는데, 보안 단일 분야에서도 상세 목적이나 응용 분야에 따라 적용할 도구나 방법의 신중한 선택이 필요하다. 본 연구에서는 멀티 소스 이벤트 로그 레코드 각각이 가지는 보안 심각도를 여러 단계로 분류하는 과업을 다루었는데, 기존 도구들이 다양한 종류를 가지는 대량의 로그에 대해 다중 클래스 분류를 효율적으로 수행하지 못한다는 한계를 가져 이를 해결할 수 있는 도구로서 자연어 처리 기반 머신 러닝을 이용하는 방안을 제시하였고, 분류 정확도 99.59%라는 높은 성능을 달성하였다.

제안한 도구를 이용해 전체 로그 건수에 대해 높은 분류 정확도를 달성할 수 있는 것은 사실이나 문제 영역의 특성상 높은 심각도를 가지는 로그의 건수가 많지 않고 이러한 특성을 효과적으로 다루기 위해서는 벡터화에 사용될 토큰 수를 적게 제한할 필요가 있음을 확인하였다. 그러나 너무 작은 숫자를 선택하면 전체 로그 데이터 중 높은 비중을 차지하는 낮은 심각도를 가지는 로그를 제대로 분류하지 못하는 결과로 이어질 수 있어 적절한 숫자를 선택하는 방법에 관한 추가적인 연구가 필요하다. 또한 모델이 내린 결정의 근거를 보다 정확하고 자세하게 제시하는 방안이 필요한데, 이와 관련해서 설명 가능한 인공지능(XAI) 기술을 함께 적용하는 방안도 이어지는 연구로 수행하고자 한다.

References

[1] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M.R. Lyu, "A survey on automated log analysis for reliability engineering," ACM Computing Surveys, vol. 54, no. 6, pp. 1-37, Jul. 2021.

[2] Z. Chen, J. Liu, W. Gu, Y. Su, and M.R., Lyu, "Experience report: Deep learning-based system log analysis for anomaly detection," arXiv preprint arXiv:2107.05908, Jul. 2021.

[3] J. Zhu, S. He, J. Liu, P. He, Q. Xie, Z. Zheng, and M.R. Lyu, "Tools and benchmarks for automated log parsing," Proceedings of IEEE/ACM 41st International Conference on Software Engineering, pp. 121-130, May 2019.

[4] M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, "System log clustering approaches for cyber security applications: A survey," Computers & Security, vol. 92, pp. 101739-101756, May 2020.

[5] P. He, J. Zhu, Z. Zheng, and M.R. Lyu, "Drain: An online log parsing approach with fixed depth tree," Proceedings of the 2017 IEEE International Conference on Web Services, pp. 33-40, Jun. 2017.

[6] R. Yang, D. Qu, Y. Qian, Y. Dai, and S. Zhu, "An online log template extraction method based on hierarchical clustering," EURASIP Journal on Wireless Communications and Networking, vol. 2019, no. 1, pp. 882-895, Dec. 2019.

[7] J. Raffety, B. Stone, J. Svacina, C. Woodahl, T. Cerny, and P. Tisnovsky, "Multi-source log clustering in distributed systems," Proceedings of the 11th International Conference on Information Science and Applications, pp. 31-41, Dec. 2020.

[8] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285-1298, Oct. 2017.

[9] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun, and R. Zhou, "LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs," Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 4739-4745, Aug. 2019.

[10] "AI competition for predicting security risk level through log analysis", dacon.io/competitions/official/235717/overview/description, Aug. 2022

[11] Z. Liu, T. Qin, X. Guan, H. Jiang, and C. Wang, "An integrated method for anomaly detection from massive system logs," IEEE Access, vol. 6, pp. 30602-30611, Jun. 2018.

[12] T. van Ede, H. Aghakhani, N. Spahn, R. Bortolameotti, M. Cova, A. Continella, M. van Steen, A. Peter, C. Kruegel, and G. Vigna, "DEEPCASE: Semi-supervised contextual analysis of security events," Proceedings of the 43rd IEEE Symposium on Security and Privacy, pp. 522-539, May 2022.

[13] K. Erk, "Representing words as regions in vector space", Proceedings of the 13th Conference on Computational Natural Language Learning, pp. 57-65, Jun. 2009.

[14] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," Proceedings of the 3rd International Conference on Information Computing and Applications, pp. 246-252, Sep. 2012.

[15] Y. Zhang, R. Jin, and Z.H. Zhou, "Understanding bag-of-words model: A statistical framework," International Journal of Machine Learning and Cybernetics, Vol. 1, no. 1, pp. 43-52, Dec. 2010.

[16] C. Wan, Y. Wang, Y. Liu, J. Ji, and G. Feng, "Composite feature extraction and selection for text classification," IEEE Access, vol. 7, pp. 35208-35219, May 2019.

[17] "NLP based log analysis test", allaboutxai.github.io/ml_dl/2022/08/25/ml_dl-ml_LogAnalysis/, Aug. 2022

〈저자소개〉

서 양 진 (Yangjin Seo) 정회원
1998년: 중앙대학교 컴퓨터공학과 (학사)
2000년: 중앙대학교 컴퓨터공학과 (석사)
2010년: 중앙대학교 컴퓨터공학과 (박사)
2004년~2010년: (주)소프트캠프 차장
2010년~2012년: (주)시큐아이 차장
2017년~2020년: (주)코튼캔디 CTO
2020년~현재: (주)이포즌 연구소장
〈관심분야〉 딥 러닝, 디지털 트윈, 디지털 전환, 보안