Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.3, Jun. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.3.365

정상 사용자로 위장한 웹 공격 탐지 목적의 사용자 행위 분석 기법*

신 민 식,[1†] 권 태 경[2‡]
[1,2]연세대학교 (대학원생, 교수)

User Behavior Based Web Attack Detection in the Face of Camouflage*

MinSik Shin,[1†] Taekyoung Kwon[2‡]
[1,2]Yonsei University (Graduate student, Professor)

요 약

인터넷 사용자의 급증으로 웹 어플리케이션은 해커의 주요 공격대상이 되고 있다. 웹 공격을 막기 위한 기존의 WAF(Web Application Firewall)는 공격자의 전반적인 행위보다는 HTTP 요청 패킷 하나하나를 탐지 대상으로 하고 있으며, 새로운 유형의 공격에 대해서는 탐지하기 어려운 것으로 알려져 있다. 본 연구에서는 알려지지 않은 패턴의 공격을 탐지하기 위해 기계학습을 활용한 사용자 행위 기반의 웹 공격 탐지 기법을 제안한다. 공격자가 정상적인 사용자인 것처럼 위장할 수 있는 부분을 제외한 영역에 집중하여 사용자 행위 정보를 정의였으며, 벤치마크 데이터셋인 CSIC 2010을 활용하여 웹 공격 탐지 실험을 수행하였다. 실험결과 Decision Forest 알고리즘에서 약 99%의 정확도를 얻었고, 동일한 데이터셋을 활용한 기존 연구와 비교하여 본 논문의 효율성을 증명하였다.

ABSTRACT

With the rapid growth in Internet users, web applications are becoming the main target of hackers. Most previous WAFs (Web Application Firewalls) target every single HTTP request packet rather than the overall behavior of the attacker, and are known to be difficult to detect new types of attacks. In this paper, we propose a web attack detection system based on user behavior using machine learning to detect attacks of unknown patterns. In order to define user behavior, we focus on features excluding areas where an attacker can camouflage as a normal user. The experimental results shows that by using the path and query information to define users' behaviors, best results for an accuracy of 99% with Decision forest.

Keywords: Anomaly Detection, User Behavior, Web Attack, Machine Learning

I. 서 론

오늘날 인터넷의 보편화와 스마트 기기의 보급으로 웹 어플리케이션을 통해 다양한 서비스를 이용하는 것이 일상이 되고 있다. WordPress, Drupal과 같은 플랫폼의 발전으로 웹 어플리케이션 신규 개발 및 변경은 점점 빨라지고 있다. 자연스럽게 웹 어플리케이션은 사용자들의 중요정보를 수집하고 처리하는 영역이 되었고, 이에 따라 해커들의 주요 공격 대상이 되고 있다. 웹 공격이 점점 증가하고 있는 상황에 웹 어플리케이션에는 여전히 많은 취약점이 존재하는데 2019년 Acunetix에서 발표한 웹 어플리케이션 취약점 보고서에 따르면 1년간 1만 개 이상의 웹 어플리케이션 대상으로 스캔한 결과 웹 어플리

Received(01. 18. 2021), Modified(04. 14. 2021), Accepted(04. 16. 2021)
* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2021-2020-0-01602)
† 주저자, msshinaktl@gmail.com
‡ 교신저자, taekyoung@yonsei.ac.kr(Corresponding author)

케이션의 46%는 위험도 상 수준의 취약점이, 87%는 위험도 중 수준의 취약점이 존재한다고 밝혔다[1].

웹 공격 방지를 위한 웹 방화벽은 전통적인 규칙 기반 탐지 방식의 한계로 새로운 유형의 공격에 대해서는 탐지가 힘든 것으로 알려져 있다. 또한, HTTP 요청 패킷 하나하나를 대상으로 웹 공격 여부를 판단해야 하다 보니 제한적인 패킷 구조 내에서 다양한 특성(feature)을 추출하려고 노력한다. 이때 HTTP 요청 패킷에는 공격자가 정상적인 사용자인 것처럼 위장할 수 있는 영역이 존재하는데 이러한 영역을 특성으로 삼는 경우 정확도가 떨어질 수 있다. 예를 들어 기존 연구[2]에서 수행한 방식과 동일하게 HTTP 요청 패킷에서 6개 영역(url, host, Content-Length, Content-Type, Cookie, payload)의 값을 추출하고 LR(Logistic Regression), SVM(Support Vector Machine) 두 개 기계학습 알고리즘을 통해 웹 공격 탐지를 수행한 결과 Table 1.의 좌측 열과 같은 결과를 확인할 수 있다. 하지만 url, payload, host 정보 외에는 공격자가 위장할 수 있는 영역이므로 이를 정상적인 사용자의 값으로 변조할 경우 Table 1.의 우측과 같이 성능이 떨어지는 것을 확인할 수 있다.

따라서 본 연구에서는 위와 같은 문제 해결을 위해 웹 어플리케이션 사용자의 행위에 기반한 웹 공격 탐지를 수행한다. 공격자가 정상 사용자로 위장할 수 있는 영역을 제외한 Path, Query 정보만으로 사용자 행위 특성을 표현할 수 있도록 가공하고, 기계학습을 통해 웹 공격 탐지를 수행한다.

Table 1. Comparison between original and camouflaged dataset

Score	Original		Camouflaged	
	LR	SVM	LR	SVM
Accuracy	0.963	0.963	0.701	0.747
Precision	0.971	0.973	0.805	0.700
Recall	0.940	0.940	0.393	0.708
F1-Score	0.955	0.956	0.528	0.704
AUC	0.995	0.996	0.842	0.818

II. 사용자 행위 기반의 웹 공격 탐지 기법 설계

2.1 배경 및 개요

사용자 행위를 분석하여 웹 공격 여부를 판단할 수 있다고 가능성을 확인한 것은 웹 공격 행위만의 고유한 특성이 존재하기 때문이다. 일반적으로 웹 공격은 다른 행위가 전혀 없던 상태에서 HTTP 요청 하나를 통해 갑자기 공격에 성공할 수 없다. 공격에 성공하기까지 수많은 HTTP 요청이 이루어지는데 공격자의 행위를 다음과 같은 순서로 정리할 수 있다.

1) 웹 어플리케이션 분석 : 취약점 포인트를 확인하기 위해 여러 웹 페이지에 접근하면서 웹 어플리케이션의 동작을 분석한다. 메뉴를 이동하거나 기능을 수행했을 때 URL과 파라미터는 어떤 방식으로 요청하는지 확인한다.
2) 취약점 확인 : 취약점 가능성을 확인하기 위해 파라미터 값에 특수문자나 공격 구문을 삽입해본다. 사용자 입력값이 응답 소스코드에 필터링 없이 삽입되는지, 특수문자를 입력했을 때 SQL 오류 구문이 출력되지는 않는지 등을 확인한다.
3) 필터링 우회 : 웹 방화벽과 같은 필터링이 있을 때 이를 우회하기 위해 노력한다. exec, document, and 등과 같은 특정 단어만 막고 있는지 확인하기도 하고, 정규표현식을 통해 막고 있는 경우 필터링의 강도가 어떤지 확인하기 위해 여러 입력값을 시도해보고 우회 패턴을 정립한다.
4) Payload 구성 : 취약점이 존재하는 것을 확인한 경우 최종적으로 원하는 결과를 얻기 위한 payload를 구성한다. 예를 들어 SQL 인젝션 취약점을 찾았고 DB 내 데이터 탈취를 원하는 경우 전체 데이터 조회를 위해 select 쿼리문을 제작한다.

이러한 공격 행위를 반복하는 동안 Fig.1.과 같이 사용자가 요청한 URL 내 Path, Query 정보에 공격자의 고유한 행위 특성이 반영될 수 있다. 취약점을 확인한 뒤 2) ~ 4) 행위를 반복하는 동안 Path 정보는 동일한 값을 계속 요청할 것이고, Query 정보는 공격 구문에서 탐지 우회 패턴이 가미되어 조금씩 수정된 값들을 지속해서 요청할 것이다. 만약 공격에 실패할 경우 성공할 때까지 1) ~ 4)의 행위를

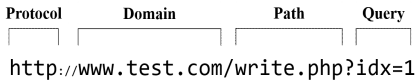

Fig. 1. URL format

계속 반복한다. 이러한 행위를 해커가 직접 수행하느냐 혹은 별도 자동화 툴을 통해 수행하느냐에 따라 약간의 차이만 있을 뿐 전반적인 웹 공격 행위의 특성은 크게 다를 바가 없다.

Fig.2.는 위와 같은 웹 공격 사용자의 행위 정보에 집중하고자 벤치마크 데이터셋에서 특성 추출 및 가공 단계부터 기계학습까지의 전체 흐름을 나타낸다. 각각의 단계별로 모델 성능을 향상시킬 수 있는 방안을 고려하면서 웹 공격 탐지 기법을 설계한다.

2.2 데이터 수집 단계

본 연구에서는 기존 연구에서 활발히 활용되고 있는 CSIC 2010 HTTP 데이터셋[9]을 사용한다. CSIC 2010 데이터셋은 스페인에서 가장 큰 연구 전담 공공기관인 CSIC에서 발표한 데이터셋으로 기존의 데이터셋은 실제 웹 어플리케이션을 대상으로 하고 있지 않아 웹 공격 탐지용으로 적합하지 않다는 취지로 발표되었다. 현재 시점으로 10년 전 데이터인 만큼 최신 웹 어플리케이션 환경과 상이하고 신규 취약점이 부족할 수 있으나 다수의 기존 연구[10-12]와 비교를 위해 해당 데이터셋을 활용할 필요가 있다.

CSIC 2010 데이터셋은 Fig.3.과 같은 HTTP 트래픽 데이터의 모음으로 총 61,065건의 패킷 중 36,000개의 정상 패킷과 25,065개의 비정상 패킷으로 구성되어 있다. 비정상 패킷에는 SQL 인젝션, 버퍼 오버플로우, 정보 수집, 파일 다운로드, CRLF 인젝션, XSS, SSI, 파라미터 변조 등과 같은 공격이 포함되어 있다.

```
GET http://localhost:8080/tienda1/index.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5;
Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml
+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=1F767F17239C9B670A39E9B10C3825F4
Connection: close
```

Fig. 3. CSIC 2010 dataset sample

CSIC 2010 데이터셋의 내용을 분석한 결과 다음과 같은 특징을 가진다. 1) 웹 서버 단에서 저장하는 로그 데이터가 아닌 HTTP 요청 패킷의 모음이다. 웹 로그에서는 확인 가능한 시간 정보가 없으므로 패킷 데이터가 시간 순서대로 쌓인 건지 알 수 없으며, 쿠키에 세션값은 할당되어 있으나 중복되는 세션값이 없고 IP 정보 또한 없으므로 사용자를 구분할 수 있는 기준이 없다. 따라서 본 연구에서는 특정 개수만큼의 패킷을 한 사용자가 요청하였다고 임의로 가정하여 샘플링 한다. 2) 비정상 패킷 데이터 내에 정상 데이터가 일부 섞여 있어 기계학습 결과에 악영향을 끼칠 수 있다. 본 연구에서는 패킷 하나하나가 아닌 사용자의 전반적인 행위 정보를 통해 웹 공격 여부를 판단하므로 문제가 되지 않는다. 하지만 real world 환경에서의 웹 공격 행위를 대표하기에는 지나치게 적은 수의 정상 패킷이 섞인 것으로 판

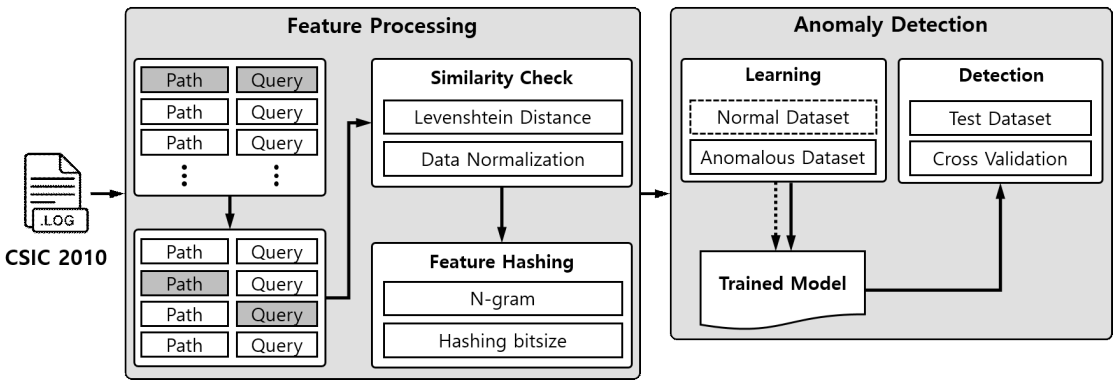

Fig. 2. Web attack detection model architecture

단되어 비정상 데이터에 정상 데이터를 추가로 섞어 재구성한다. 정상 패킷을 추가할 때에는 두 가지 방식을 사용한다. 만약 10개의 패킷을 샘플링하고 50%의 비율로 정상 패킷을 추가한다고 가정할 때 (1) 비정상 패킷 5개 뒤에 정상 패킷 5개를 순서를 유지한 채 이어붙인 방식과, (2) 비정상 패킷 5개에 정상 패킷 5개를 추가한 뒤 총 10개 패킷의 순서를 랜덤하게 섞는 방식 두 가지로 구성한다.

2.3 특성 추출 및 가공 단계

CSIC 2010 데이터셋에서 Path, Query 정보를 추출한 후 이를 각각 순차적으로 열거한다. 이후 각각 앞뒤 데이터 간의 유사도를 측정하는데 여러 텍스트 기반의 유사도 측정 방식 중 편집거리(levenstein distance) 알고리즘을 활용한다. 편집거리 알고리즘은 두 개의 문자열이 같아지려면 몇 번의 추가, 편집, 삭제가 필요한지 최소 개수를 구하여 문자열 간의 유사도를 수치화할 수 있는 알고리즘이다. 유사도 측정 결과를 별도 가공 없이 사용하기에는 고유한 데이터가 많아 모델의 정확도를 낮출 수 있으므로 각 유사도 측정 결과값을 5, 10 단위로 평준화시킨다.

최종적으로 추출된 유사도 측정 결과값 시퀀스는 하나의 문자열로 취급하여 피처 해싱(feature hashing) 과정을 거친다. 피처 해싱 기법은 특성을 벡터화하는 방식으로 수행속도가 빠르고 연산 과정에서 메모리 공간 절약이 가능한 장점이 있다. Path, Query 두 개의 특성 각각에 대해 피처 해싱 과정을 거쳐 각 특성의 차원(dimension)을 통일시킨다. 다수의 패킷이 쌓인 사용자 1명당 특성 풀이 생성되고 나면 기계학습을 위한 데이터 준비가 마무리된다.

2.4 기계학습 단계

특성 가공이 끝난 데이터는 레이블 값을 기준으로 학습 및 테스트로 사용하여 학습 모델의 성능을 측정한다. 지도 학습에 NL-SVM(Non-linear SVM), DF(Decision Forest), LR 알고리즘을 사용하였고, 기계학습 알고리즘의 평가를 위해 5-fold Cross Validation을 적용한다. 데이터 가공 시 세부 설정값에 따른 웹 공격 탐지 모델의 성능 변화를 분석하고, 최종 실험결과를 토대로 기존 연구와 성능을 비교한다.

III. 실험 방법 및 결과

실험은 Intel (R) Core (TM) i5-6600 CPU @ 3.30 GHz, 32 GB RAM 환경에 Ubuntu 18.04.2 LTS OS 상에서 진행하였다. 특성 추출 및 가공과 기계학습 기법은 Python 2.7 언어로 개발한 스크립트와 MAMLS(Microsoft Azure Machine Learning Studio) 플랫폼, Scikit-learn 모듈을 활용하였다.

3.1 주요 설정값의 최적화

Fig.2.의 feature processing 단계에서 설정값을 조절하며 반복 실험한 결과 가장 이상적인 모델 성능을 확인할 수 있는 최적화된 값을 파악할 수 있었다.

Data normalization은 데이터의 앞뒤 유사도를 측정할 때 측정값의 평준화 단위를 말한다. Table 2.에서는 데이터 평준화 적용 여부 및 평준화 단위에 따른 결과를 확인할 수 있다. AUC 지표를 기준으로 데이터 평준화를 적용하지 않거나 10단위로 적용한 것보다 5단위로 적용했을 때 0.945로 가장 높은 분류 성능을 확인할 수 있었다.

데이터 샘플링은 몇 개의 패킷을 한 명의 사용자가 요청한 것인지 가정하는 값이다. AUC 지표를 기준으로 Table 3.와 같이 샘플링 개수가 커질수록 모델의 분류 성능은 높게 나왔으나 가능한 적은 수의 패킷이 쌓였을 때 공격 여부를 판단할 수 있어야 웹 공격 탐지 모델로서 의미가 있으므로 최소 샘플링 개수를 파악할 필요가 있다. 반복실험 결과 오탐 비율은 낮으면서 공격 여부를 판단할 수 있는 최소 개수는 3인 것을 확인할 수 있었다. 본 논문에서 제안하는 모델에서는 최소 4개의 패킷이 쌓여야 하고 더 많이 쌓일수록 공격 여부를 더 정확하게 판단할 수 있다는 것을 확인하였다.

피처 해싱 기법에는 두 개의 설정값을 지정할 수 있는데 해싱 비트 크기는 해시 테이블 생성 시 비트

Table 2. Experimental results with data normalization

Data Norm	Score	NL-SVM	DF	LR
N/A	AUC	0.861	0.867	0.860
5		0.936	0.945	0.906
10		0.875	0.879	0.879

Table 3. Experimental results with data sampling

# Data Sampling	Score	NL-SVM	DF	LR
1	AUC	0.862	0.867	0.860
2		0.986	0.993	0.967
3		0.997	0.999	0.988
4		0.999	1	0.995
5		0.999	0.999	0.998

수를, n 값은 n-gram 방식의 텍스트 추출 시 몇 개의 연속적인 단어로 나열할 것인지 기준값을 말한다. 피처 해싱 설정값의 경우 값의 변화에 따라 실험 결과에 큰 영향을 주지는 않았으나 해시 테이블의 크기가 5 이하로 지나치게 작을 경우 해시 충돌(hash collision)이 일어날 수 있으므로 충분히 큰 값으로 설정할 필요가 있다. 또한 유사도 결과 값 자체보다 사용자 행위를 표현하는 유사도 결과 값들의 순서 정보가 중요하므로 n 값은 최소 2 이상으로 설정한다.

3.2 실험결과

Table 4.는 비정상 데이터에 정상 데이터를 50% 추가하여 재구성한 상태에서 accuracy, precision, recall, f1-score, AUC 5개의 지표를 통해 웹 공격 탐지 모델의 성능을 측정한 결과이다. 이 결과에서 데이터 추출 및 가공 단계의 주요 설정값 최적화를 적용한 Table 5. 결과와 비교한 결과 성능 향상을 확인할 수 있다. 실험에 사용한 3개 알고리즘 중 DF 알고리즘이 모든 지표에서 가장 높은 점수를 얻었다.

Table 6.에서는 동일한 데이터셋을 활용한 기존 연구의 결과와 비교한 결과이다. 기존의 모델 성능과 유사한 결과가 도출된 것을 확인할 수 있는데 이는 CSIC 2010 데이터셋의 특성상 비정상 데이터 내에

Table 4. Experimental results using recontructed dataset

Score	NL-SVM	DF	LR
Accuracy	0.772	0.734	0.778
Precision	0.694	0.867	0.709
Recall	0.795	0.525	0.782
F1-Score	0.741	0.635	0.743
AUC	0.813	0.827	0.796

Table 5. Performance evaluation for each algorithm

Score	NL-SVM	DF	LR
Accuracy	0.997	0.998	0.986
Precision	1	0.999	1
Recall	0.994	0.996	0.972
F1-Score	0.997	0.998	0.985
AUC	0.998	0.999	0.998

Table 6. Performance comparison on the CSIC 2010 dataset

Score	Model in [11]	Model in [12]	Model in [8]	Ours
Accuracy	0.980	0.998	0.999	0.998
Precision	0.985	0.997	1	0.999
Recall	0.960	0.997	0.997	0.996

정상 데이터가 일부 섞여 있는 것을 원인으로 분석하였다. 비정상 데이터를 재구성하여 정상 데이터를 추가한 상태인데 일부 샘플링 데이터에는 원래 데이터셋에 포함된 정상 데이터까지 포함되어 공격 여부를 판단하기에는 공격 행위 정보가 제한적이었던 것으로 파악되었다. 기존 연구와 달리 비정상 데이터 내에 정상 데이터가 다수 섞여 있더라도 사용자의 행위 정보에 집중하여 공격 여부를 판단할 수 있다는 것을 확인할 수 있었다.

IV. 관련 연구

4.1 인공지능을 활용한 비정상 탐지

전통적인 웹 공격 탐지 방법은 사전에 정의되지 않은 공격을 탐지할 수 없는 어려움이 있어 이를 해결하기 위해 인공지능을 활용한 탐지 방식이 다양하게 연구되고 있다.

Husák 등[3]은 대규모 네트워크 환경에서 HTTP 트래픽을 분석하는 방식을 제안하였다. OSI 3계층과 4계층의 데이터에 더해 HTTP 요청 패킷에서 host, path, user-agent, HTTP method, 응답 코드, referer, content-type 정보를 추출하였다. 이를 통해 Brute-force password attack, 프록시 연결, HTTP 스캐너, 웹 크롤러가 포함된 HTTP 트래픽의 패턴을 분류하였다.

Zolotukhin 등[4]은 정상 HTTP 요청 데이터 대상으로 웹 로그의 3개 영역(web resource,

query 속성, user-agent)에서 n-gram 방식으로 특성을 추출하고 클러스터링 및 비정상 탐지 알고리즘을 사용하여 웹 공격 탐지 방법을 제안했다. 기존의 데이터 마이닝 기법과 비교하여 더 높은 정확도를 보였다.

HTTP 요청 패킷 내 여러 정보 중 웹 공격 탐지를 위해 기존 연구에서 활용하고 있는 영역에는 공격자가 정상적인 사용자인 것처럼 위장할 수 있는 영역이 포함되어 있다. 반면 본 연구에서는 해당 영역을 제외하여 Path, Query 두 개 정보만으로 웹 공격 탐지를 수행한다.

4.2 사용자 행위 기반 비정상 탐지

사용자 행위 분석을 통해 웹 어플리케이션 사용자의 비정상 행위를 탐지하는 것은 DDoS (Distributed Denial of Service) 공격 탐지 분야에서 연구가 활발하다[5-7]. DDoS 공격은 일정 시간 안에 대량의 요청이 필요하므로 HTTP 요청 개수, 요청 간의 시간 차, 요청한 리소스의 크기 및 인기도 등과 같은 사용자의 브라우징 행위에 집중한다. 하지만 SQL 인젝션, XSS과 같이 사용자 입력 값 검증이 미흡하여 발생할 수 있는 웹 공격은 DDoS 공격만큼 대량의 요청이 불필요하다. 따라서 DDoS 공격 탐지 연구에서 활용한 기법을 웹 공격 탐지에 적용하는 것은 적합하지 않으며 기존과는 다른 방식의 사용자 행위 분석이 필요하다.

Zhang 등[8]은 웹 공격 탐지를 위해 사용자의 행위 속에 숨겨진 악성 의도를 분석하는 접근 방식을 제안하였다. CSIC 2010 HTTP 데이터셋에서 HTTP method, Path, Query 정보를 추출하고 워드 임베딩 기법을 통해 벡터화한 후 사용자 행위 모델로 사용하였다. 한 개의 파라미터를 사용한 최대 횟수, 고유한 Path의 비율, Query 최대 길이, Query 평균 길이 등 9가지를 특성으로 추출하여 시나리오 모델을 추가로 생성하였다. 이후 두 개 모델을 통해 앙상블 학습으로 공격 탐지 모델을 구축하였다.

Zhang 등[8]의 연구는 CSIC 2010 데이터셋을 활용하면서 사용자 행위 정보를 추출하여 비정상 행위를 탐지한다는 점에서 본 연구와 가장 유사하다. 하지만 공격자가 정상적인 사용자인 것처럼 위장할 수 있는 영역을 특성으로 활용하고, 텍스트 데이터를 벡터화할 때 본 연구에서 제안한 방식과 차이가 있다. 또한, 데이터셋을 별도 가공 없이 사용한 반면 본 연구에서는 비정상 데이터를 재구성하여 실험하였다.

V. 결 론

본 연구에서는 웹 공격 탐지를 위해 사용자의 행위 정보를 추출하는 방법에 대해 연구하였다. 제시한 방법은 공격자가 정상 사용자인 것처럼 위장할 수 있는 것을 방지하기 위해 해당 영역을 제외한 Path, Query 두 개 데이터에만 집중하여 사용자 행위 특성을 표현하였다. 또한, 알려지지 않은 유형의 공격을 탐지할 수 있도록 공격 구문의 내용은 활용하지않고 패킷 데이터 간의 유사도를 측정하여 순차적인 패킷 흐름 속에서 웹 공격 행위 여부를 판단하였다. 특히 비정상 데이터에 정상 데이터를 추가하여 real world에 근접하도록 재구성한 상태에서도 0.999의 정확도를 확인하여 본 연구의 우수성을 증명하였다.

향후 별도의 웹 어플리케이션을 구축하여 웹 로그를 수집하거나 최신 웹 취약점 스캐닝 도구를 활용하는 등 다양한 패킷 데이터를 수집하여 실험에 활용할 필요가 있다. 본 연구에서 제안한 방식이 CSIC 2010 데이터셋 한정의 웹 공격 행위가 아닌 전반적인 웹 공격 행위를 아우를 수 있는지 검증이 필요하다.

References

[1] Acunetix, "Web Application Vulnerability Report" https://www.acunetix.com/blog/articles/acunetix-web-application-vulnerability-report-2019/, 2019.

[2] Smitha, Rajagopal, K. S. Hareesha, and Poornima Panduranga Kundapur, "A machine learning approach for web intrusion detection: MAMLS perspective," Soft Computing and Signal Processing, Springer, vol 1, no. 12, pp. 119-133, Jan. 2019.

[3] Husák, Martin, Petr Velan, and Jan Vykopal, "Security monitoring of HTTP traffic using extended flows," In Proc. the Availability, Reliability and Security (ARES), IEEE, pp. 258-265, Aug. 2015.

[4] Zolotukhin, M., Hämäläinen, T., Kokkonen, T., and Siltanen, J, "Analysis

of HTTP requests for anomaly detection of web attacks," In Proc. the Dependable, Autonomic and Secure Computing (DASC), IEEE, pp. 406-411, Aug. 2014.
〔5〕 Goseva-Popstojanova, Katerina, Goce Anastasovski, and Risto Pantev, "Classification of malicious web sessions," In Proc. the International Conference on Computer Communications and Networks (ICCCN), IEEE, pp. 1-9, Aug. 2012.
〔6〕 Ye, Chengxu, Kesong Zheng, and Chuyu She, "Application layer DDoS detection using clustering analysis," In Proc. the Computer Science and Network Technology (ICCSNT), IEEE, pp. 1038-1041, Dec. 2012.
〔7〕 Liao, Q., Li, H., Kang, S., and Liu, C, "Application layer DDoS attack detection using cluster with label based on sparse vector decomposition and rhythm matching," Security and Communication Networks, vol 8, no. 17, pp. 3111-3120, Mar. 2015.
〔8〕 Zhang, Yunyi, Jintian Lu, and Shuyuan Jin, "Web attack detection based on user behaviour semantics," In Proc. the Algorithms and Architectures for Parallel Processing, pp. 459-474, Sep. 2020.
〔9〕 Giménez, Carmen Torrano, Alejandro Pérez Villegas, and Gonzalo Álvarez Marañón, "HTTP data set CSIC 2010," Information Security Institute of CSIC (Spanish Research National Council), 2010.
〔10〕 Pham, Truong Son, Tuan Hao Hoang, and Vu Van Canh, "Machine learning techniques for web intrusion detection-a comparison," In Proc. the Eighth International Conference on Knowledge and Systems Engineering (KSE), IEEE, pp. 291-297, Oct. 2016.
〔11〕 Gharibeh, Samar, Shatha Melhem, and Hassan Najadat, "Classification on web application requests," In Proc. the 11th International Conference on Information and Communication Systems (ICICS), IEEE, pp. 1-5, Apr. 2020.
〔12〕 Xinyu Gong, Jialiang Lu, Yuchen Wang, Han Qiu, Ruan He, and Meikang Qiu, "CECoR-Net: A character-level neural network model for web attack detection," In Proc. the Smart Cloud (SmartCloud), IEEE, pp. 98-103, Dec. 2019.

〈저자소개〉

신 민 식 (MinSik Shin) 학생회원
2016년 2월: 연세대학교 컴퓨터공학과 학사
2021년 2월: 연세대학교 정보보호연구실 석사
〈관심분야〉 시스템 보안, 네트워크 보안, 스마트폰 보안 등

권 태 경 (Taekyoung Kwon) 종신회원
1992년 2월: 연세대학교 컴퓨터과학과 학사
1995년 2월: 연세대학교 컴퓨터과학과 석사
1999년 8월: 연세대학교 컴퓨터과학과 박사
1999년~2000년: U.C. Berkely Post-Doc
2001년~2013년 8월: 세종대학교 컴퓨터공학과 교수
2007년~2008년 Univ. Maryland at College Park 교환교수
2013년 9월~현재: 연세대학교 정보대학원 교수
〈관심분야〉 암호 프로토콜, 네트워크 프로토콜, 사물인터넷 보안, HCI 보안 등