https://doi.org/10.7840/kics.2023.48.8.959

SaaS 응용의 사용자 행위 탐지를 위한 플로우 기반의 고도화된 규칙 생성

박 지 태•, 백 의 준*, 신 창 의**, 최 정 우*, 홍 윤 환***, 호 태 규***, 김 명 섭°

A Flow-Based Advanced Rule Generation for Detecting User Action in SaaS Application

Jee-Tae Park•, Ui-Jun Baek*, Chang-Yui Shin**, Jung-Woo Choi*,
Yoon Hwan Hong***, Tae-Gyu Ho***, Myung-Sup Kim°

요 약

최근 네트워크 기술의 발전으로 클라우드 서비스를 기반으로 하는 응용에 대한 관심이 높아지고 있다. 여러 가지 클라우드 서비스 중 SaaS(Software as a Service)가 가장 활발하게 사용되고 있으며, Google, Microsoft와 같은 기업에서도 널리 사용하고 있다. SaaS는 구독 형태로 제공되며, 라이선스, 사용 기간, 사용 가능 인원 등에 따라서 금액이 달라지기 때문에 사용자 행위 탐지에 대한 연구가 수행되고 있다. 본 연구의 선행 연구에서는 사용자 행위 탐지를 위해 규칙 기반 행위 탐지 방안에 대해 제안하였다. 하지만 선행 연구에서는 규칙을 생성 할 때, SNI 정보에 대한 의존도가 너무 높기 때문에 특정 응용을 대상으로는 낮은 성능을 보인다는 문제점이 있다. 따라서 본 논문에서는 기존 방법의 문제점을 해결할 수 있는 고도화된 규칙 생성 방법을 제안한다. 제안하는 방법의 타당성을 검증하기 위해 본 연구의 선행 연구와 탐지 성능에 대한 비교 실험을 수행한다.

Key Words : Network Traffic Analysis, Rule based User Acion Detection, Signature Generation

ABSTRACT

Recently, with the development of network technology, interest in applications based on cloud services is increasing. Among various cloud services, SaaS (Software as a Service) is most actively used, and is widely used by large corporations such as Google and Microsoft. SaaS is provided in the form of a subscription, and since the amount varies depending on the license, usage period, number of users,etc., research on user action detection is being conducted. In our previous study, a rule-based action detection method was proposed. However, in previous research, when generating rules, there is a problem in that they show low performance for specific applications because the dependence on SNI information is high. Therefore, in this paper, we proposed an advanced rule generation method that can solve the problems of the existing method detection performance with previous study in this study.

※ 본 논문은 2020년도 산업통상자원부 및 한국산업기술평가관리원(KEIT) 연구비 지원에 의한 연구이며(No. 20008902, IT비용 최소화를 위한 5채널 탐지기술 기반 SaaS SW Management Platform(SMP) 개발), 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과임. (2021RIS-004)

• First Author : Department of Computer and Information Science, Korea University, pjj5846@korea.ac.kr, 학생회원
° Corresponding Author : Department of Computer and Information Science, Korea University, tmskim@korea.ac.kr, 종신회원
* Department of Computer and Information Science, Korea University, {pb1069, choigoya97}@korea.ac.kr, 학생회원
** Defense Agency for Technology and Quality, superego99@dtaq.re.kr
*** Doctorsoft co. LTD, {hong, tghui92}@doctorsoft.co.kr

논문번호 : 202304-078-B-RN, Received April 17, 2023; Revised June 5, 2023; Accepted June 15, 2023

Ⅰ. 서 론

최근 네트워크 기술의 발전으로 다양한 응용이 발생하고 있으며, 특히 클라우드 서비스를 기반으로 하는 응용이 활발하게 개발되고 있다. 클라우드 서비스는 네트워크가 연결되어 있는 환경에서 가상화된 컴퓨터 리소스를 서비스 형태로 제공한다. 제공되는 가상화 리소스의 형태에 따라 PaaS, IaaS, SaaS로 구분되며, SaaS가 가장 널리 사용되고 있다. SaaS는 리소스를 서비스 형태로 제공하는 클라우드 서비스로 Google, Microsoft와 같은 대기업을 포함한 여러 기업에서 채택하고 있으며, 점차적으로 사용이 증가하고 있다. SaaS는 사용자가 원하는 기능을 응용 및 서비스 형태로 제공 할 수 있기 때문에, 일반 사용자가 사용하기에 가장 용이하다. 또한 기존의 일반 응용처럼 개별적으로 1회성 구매가 아니라 정해진 기간 동안 구독하는 형태로 제공되기 때문에 필요한 기능에 따라 적절하게 구매 할 경우 비용을 절감 할 수 있다. 하지만 기업과 같은 여러 사용자가 사용하는 환경에서 불필요한 기능을 포함하거나 실제 사용자 수보다 더 많은 라이선스를 구매 할 경우 과도한 지출이 발생 할 수 있다. 따라서 네트워크 관리자는 사용자의 행위에 대한 정확한 탐지 및 지속적인 모니터링을 통해 불필요한 지출을 막아야 한다.

사용자 행위 탐지는 네트워크 보안 및 관리 측면에서 중요한 역할을 하기 때문에 오래 전부터 연구가 되어왔다[1-3]. 악성 행위를 수행하는 해커 혹은 공격자들은 네트워크 내 악성 행위가 방화벽 및 보안 시스템에 탐지되지 않기 위해 정상 행위인 것처럼 한다. 이를 위해 대상 응용의 정상적인 동작 과정 및 행위를 사전에 분석하고, 분석된 정상 행위 정보를 활용하여 실제 공격을 수행한다. 네트워크 관리자는 네트워크 내 사용자의 행위에 대해 지속적으로 모니터링을 통해 악성 행위에 대해 대비해야 한다.

사용자 행위 탐지 연구는 각 연구 별로 목적에 따라 대상 응용과 사용자 행위 정의가 다르다[2-5]. 본 연구의 선행 연구에서 규칙 기반 사용자 행위 탐지 방법에 대한 연구를 수행하였으며, 높은 정확도로 대상 응용의 행위를 정확하게 탐지 할 수 있었다[5]. 하지만 선행 연구에서는 SNI(Server Name Indication)와 헤더 정보를 사용하여 규칙을 생성하지만, SNI 정보에 대한 의존도가 너무 높기 때문에, 행위에 대한 고유 SNI 정보가 없거나 SNI 시그니처가 자주 바뀌는 경우에 제안하는 방법을 적용하기 어렵다는 문제점이 있다.

따라서 본 논문에서는 기존의 규칙 생성 방법의 문제점을 해결하기 위해 플로우 통계 정보를 추가로 적용한 탐지 규칙 생성 시스템을 제안한다. 제안하는 방법은 규칙 생성 할 때 SNI 정보 이외 통계 정보를 추가적으로 사용하기 때문에 기존 규칙 생성 방법의 SNI에 대한 높은 의존도를 해결할 수 있다.

본 논문은 1장 서론에 이어 2장에서는 사용자 행위 탐지 분야의 관련 연구를 설명하고, 3장에서 제안하는 방법을 설명한다. 4장에서는 제안하는 방법을 검증하기 위한 실험에 대해 기술하며, 5장에서는 결론으로 제안하는 방법의 전반적인 내용과 향후 연구에 대해 설명한다.

Ⅱ. 관련 연구

2.1 사용자 행위 탐지 연구

사용자 행위 탐지 연구를 수행하기 위해 우선적으로 탐지 할 대상 응용의 사용자 행위를 정의해야 한다. 사용자 행위 정의에 대한 예시는 표 1에 나타나있으며, 응용의 특성, 사용 형태에 따라서 다양하게 정의된다.

응용의 특성은 대상 응용이 무엇을 기반으로 동작하는 지를 나타내며, 웹 기반, 설치 기반으로 구분된다. 웹 기반은 네트워크가 있는 환경에서 웹을 통해 동작하는 응용이며, 설치 기반은 특정 호스트에 설치를 통해 동작하는 응용을 뜻한다. 사용 형태는 대상 응용의 사용 목적에 따라 다양하게 정의되며, 대표적으로 메신저, 파일 작업 등이 있다. C. Hou et. al[2]와 K. Park et. al[3]는 각각 WeChat[2], KakaoTalk[3]의 메신저 응용을 대상으로 Chat, Receive a Message 등의 행위를 정의하고, M. Conti, L. et. al[4]는 Google+를 대상으로 open, refresh, send post 등의 정의한다. 본 연구의 선행 연구[5]에서는 파일 작업을 위한 Microsoft Office 365를 대상으로 시작, 로그인, 로그아웃, 종료의 4 가지 행위를 정의하였다.

네트워크 사용자 행위 탐지에 대한 연구는 크게 웹 기반 응용과 모바일 응용을 대상으로 한 연구로 분류된다. 먼저 웹 기반 응용에 대한 연구는 일반적인 웹에서

표 1. 사용자 행위 정의 예시
Table 1. An Example of User Action Definition

응용	특성	형태	사용자 행위 예시
WeChat[2]	웹	메신저	Chat, File Transfer 등
KakaoTalk[3]	웹	메신저	Join a chat room, Receive a message 등
Google+[4]	웹, 설치	파일 작업	open, refresh, delete post 등
Office 365[5]	웹, 설치	파일 작업	시작, 로그인, 로그아웃, 종료

발생하는 트래픽을 분석하여 사용자 행위를 추적한다. 주로 HTTP의 암호화 되지 않은 정보를 활용하여 수행되며, 사용자 행위 추론, 식별과 관심사 등을 추론한다. 모바일 기반 응용에 대한 연구는 모바일 기기에서 사용되는 응용에 대한 사용자 행위 및 개인 정보를 추적한다. 주로 Android 기기 환경에서 WeChat, KakaoTalk 등의 메신저 혹은 Twitter, Instagram 등의 SNS 응용을 대상으로 연구가 진행되었다[2,3].

2.2 자동 시그니처 생성 연구

시그니처 생성 방법론에 대한 연구는 응용 트래픽 분류 분야에서 오랫동안 수행되었다. 대상 응용의 고유 트래픽 패턴을 추출하여 시그니처로 정의하고, 정의 된 시그니처를 활용하여 대상 응용을 탐지하는 방법이다. 시그니처는 헤더, 통계, 페이로드 등의 대상 응용 트래픽의 고유 정보에 따라 여러 가지 형태로 정의 할 수 있으며, 그 중 페이로드 시그니처가 가장 정확하고 널리 사용된다[6-9]. 하지만 시그니처를 생성하기 위해서는 대상 응용에 대한 사전 분석이 필요하며, 이를 위해 많은 시간과 비용이 든다. 또한, 페이로드 시그니처의 경우, 암호화 기술 적용 및 보급화로 패킷 내 페이로드 내용이 암호화되기 때문에 고정된 문자열 값이 아닌 암호화 된 문자열 혹은 일반적인 문자열 사이에 가변적인 문자가 추가되어 나타난다. 따라서 암호화 된 페이로드를 대상으로 시그니처를 수동으로 생성하는 것은 더욱 어려워졌다.

이를 해결하기 위해서 최근에는 시그니처를 자동으로 생성하거나, 효율적으로 생성하는 방법론[8-11]에 대한 연구가 점차적으로 증가하고 있다. Lee. S. et al.는 LDA(Laten Dirichle Allocation) 알고리즘을 기반으로 malware 탐지를 위한 IDS 규칙을 자동으로 생성하는 LARGen을 제안한다[8]. Shim. K. S et. al.은 AprioriAll 알고리즘을 기반으로 자동 페이로드 시그니처 생성 방법을 제안한다[9]. Zhang, R. et. al은 DBSCAN 알고리즘 기반의 클러스터링과 모델 추론 기반의 LIME 알고리즘을 활용하여 악성 트래픽을 대상으로 자동 규칙 생성 방법을 제안한다[10].

본 논문에서는 선행 연구에서 수행된 규칙 기반의 행위 탐지 방법의 문제점을 개선하기 위해 플로우 통계 정보를 활용한 규칙 생성 시스템을 제안한다.

Ⅲ. Advanced Rule Generation System

본 장에서는 먼저 제안하는 시스템 구조와 세부 모듈에 대해 설명하고, 다음으로 규칙을 생성하기 위한 시그니처를 판별하는 방법에 대해 설명한다. 제안하는 시스템에 대한 구조는 그림 1에 나타나있으며, 크게 규칙 생성과 규칙 테스트 파트로 구분된다.

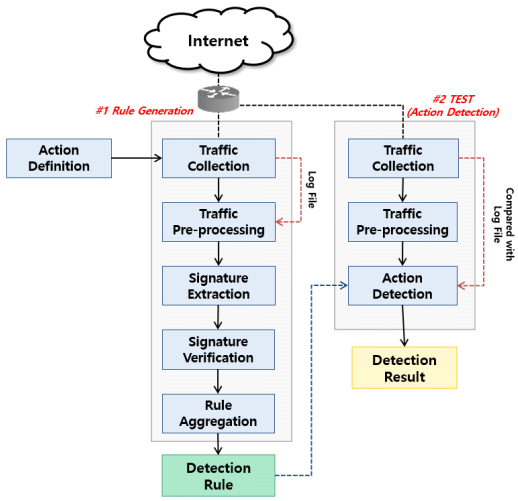


그림 1. 제안하는 시스템 전체 구조
Fig. 1. Entire Structure of Proposed System

먼저 규칙 생성 파트는 5 가지의 모듈로 이루어져 있으며, 트래픽 수집, 전처리, 시그니처 추출, 시그니처 검증, 규칙 취합 과정으로 구성되어 있다. 규칙 테스트 파트는 3 가지의 모듈로 이루어져 있으며, 트래픽 수집, 전처리, 행위 탐지 과정으로 구성된다.

3.1 행위 정의(Action Definition)

행위 정의 과정은 대상 응용에 대한 분석을 통해 어떠한 행위를 탐지 할 것인지 정의한다. 사용자 행위는 응용의 특성에 따라 다양하게 정의 할 수 있기 때문에, 대상 응용의 특성과 행위 탐지의 목적에 따라 정의한다. 본 논문에서는 SaaS 응용을 대상으로 효율적인 지출 관리 및 네트워크 보안을 목적으로 연구를 수행하기 때문에 사용자의 대상 응용 사용 내역과 관련된 행위를 탐지한다. 따라서 사용자의 실제 사용 내역과 관련된 4 가지 행위(응용 시작, 로그인, 로그아웃, 응용 종료)를 정의한다.

3.2 트래픽 수집 (Traffic Collection)

트래픽 수집은 수집 대상 응용에 대한 시그니처를 정의하기 위해 대상 응용의 트래픽을 수집하는 과정이다.

본 연구에서는 Wireshark를 활용하여 패킷을 수집하며, 수집 된 트래픽에 대하여 각 행위가 수행된 시간, 호스트 IP, 대상 응용, 수행된 행위에 대한 정보를 기록하고, 이를 로그 파일로 저장한다. 기록된 정보는 규칙

생성과 행위 탐지의 전처리 과정에 사용된다.

3.3 트래픽 전처리(Traffic Pre-processing)

트래픽 전처리 모듈은 수집 된 패킷 단위의 트래픽 파일을 플로우 단위로 전처리 과정을 수행하며, 플로우는 5-tuples 정보(Source IP, Source Port, Protocol, Destination IP, Destination Port)가 같은 패킷들의 집합으로 정의한다.

규칙 생성 과정은 대상 응용의 행위에 대한 공통 특징을 추출하며, 정확한 특징을 추출하기 위해서는 대상 응용의 행위에 대한 GT(Ground Truth) 트래픽이 필요하다. 따라서 규칙 생성 과정에는 수집된 트래픽에 대하여 Protocol, Destination IP와 같은 헤더 정보로 대상 응용에 대한 트래픽으로 필터링을 수행하고, 로그 파일로부터 행위 시점을 로드하고, 행위 시점을 기준으로 +3, -3초에 해당하는 트래픽을 자르는 과정이 수행된다. 규칙 생성에 대한 전처리 과정은 그림 2에 나타나있다.

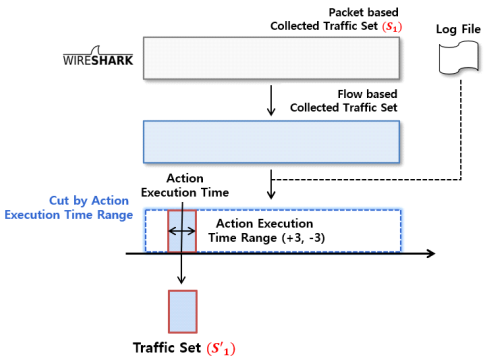


그림 2. 규칙 생성에 대한 트래픽 전처리 과정
Fig. 2. Traffic Pre-processing for Rule Generation

3.4 시그니처 추출 (Signature Extraction)

시그니처 추출 과정은 대상 응용 트래픽을 입력으로 공통 특징을 추출하고, 이를 활용하여 규칙을 생성하는 과정이다. 대상 응용에 대해 추출된 트래픽 특징 예시는 그림 3에 나타나있으며, 대상 응용의 개별 행위에 대한 트래픽을 입력으로 헤더, SNI, 통계 정보로 구성된 세 가지 시그니처를 추출한다.

먼저 헤더 시그니처는 플로우의 5-tuples 정보를 기준으로 공통 정보를 찾으며, 주로 대상 응용에서 고정된 Destination IP 혹은 Port, Protocol을 사용 할 경우 해당된다.

다음으로 SNI 시그니처는 암호화된 플로우에서 도출되는 공통 SNI 정보를 찾는다. SNI 정보는 HTTPS 통신 과정에서 TLS-handshake 과정에서 확인 할 수 있는 정보로 호스트가 통신하는 대상에 대한 정보를 가지고 있다. 응용을 실행 할 때, 로그인, 로그아웃 과정과 같이 특정 사이트를 방문하게 되거나 우회 할 경우, 특정 SNI 정보를 가지게 된다.

하지만 입력으로 받은 트래픽에는 대상 응용 뿐만 아니라 백그라운드로 실행되는 여러 가지 프로세스로부터 생성된 여러 가지 플로우가 포함되기 때문에 많은 양의 SNI 정보를 가지게 되기 때문에, GT 트래픽을 대상으로 공통 SNI 시그니처를 추출한다.

마지막으로 통계 시그니처는 플로우에 대한 통계 정보를 나타내며, PSD(Packet Size Distribution)를 활용한다. PSD는 플로우 내 패킷들의 길이 분포로, 1~N번째 패킷 길이를 벡터로 저장하며, 패킷 방향에 따라 +, - 값을 가진다. 예를 들어 그림 3에서 응용 X의 로그인 행위를 수행 할 때, 발생한 플로우에서 7번째까지 (N=7인 경우)의 패킷의 길이를 PSD로 나타내면 [+10, +14, -20, +32, +25, -18, -258]의 벡터로 나타낼 수 있다.

Application	Action	Common Feature		
		Information		Value
Application "X"	Application Start	Header Information	Destination IP	13.x.x.x
			Destination Port	443
			Protocol	TCP
		SNI Information	SNI	"www.xxx.com"
		Statistical Information	PSD	*(Changing Value)*
	Login	Header Information	Destination IP	*(Changing Value)*
			Destination Port	TCP
			Protocol	443
		SNI Information	SNI	"xxx.xxxonline.com"
		Statistical Information	PSD (N=7)	[+10, +14, -20, +32, +25. -18, -258]

그림 3. 트래픽 특징 추출 예시
Fig. 3. An Example of Traffic Feature Extraction

규칙을 생성하기 위해서는 먼저 추출된 시그니처들 중 공통적으로 발생하는 시그니처를 선별하고, 이를 시그니처로 정의해야 한다. 트래픽 특징 별로 형태와 값이 다르기 때문에 선별 방법이 달라진다.

일반적으로 헤더, SNI 정보는 문자열로 되어있으며, 주로 고정된 값을 가진다. 따라서 헤더, SNI 시그니처는 고정된 문자열을 가지기 때문에 수집된 여러 GT 트래픽 셋에서 모든 플로우를 대상으로 헤더, SNI 정보를 추출하고, 트래픽 셋 별로 공통적으로 나오는 정보를 시그니처로 정의한다. 예를 들어, 응용 A의 행위 B에 대한 여러 가지의 수집 된 트래픽 셋에서 목적지 주소가 "*133.x.x.x*", 프로토콜 및 포트 번호가 TCP, 443의 헤더 정보와 SNI 정보가 "*xxx_teststring.com*"가 공통으로 발생 할 때, 해당 값들을 취합하여 응용 A의 행위 B에 대한 시그니처로 정의 할 수 있다

통계 정보는 헤더, SNI 정보와 다르게 정수형 벡터 값으로 구성된다. 하지만 플로우의 통계 정보는 트래픽 셋 별로 값이 달라질 수 있기 때문에 고정된 값을 시그니처로 정의 할 수 없다. 따라서 통계 정보는 고정된 벡터 값을 사용하지 않고 클러스터링을 활용하여 중심 벡터를 시그니처로 정의한다.

본 연구에서는 개별 행위에 대한 여러 개의 GT 트래픽 셋을 대상으로 PSD에 대한 분석을 수행한다. 통계시그니처 추출 과정은 대상 행위의 여러 가지 GT 트래픽 통계적 특징을 모두 포괄하는 PSD를 찾는 것으로, k-means 클러스터링을 통해 모든 Trace의 플로우를 포함하는 클러스터를 찾는 것을 목표로 한다.

먼저 시그니처 추출 과정에서 도출된 PSD 벡터 값을 대상으로 k-means 클러스터링 알고리즘을 적용하여 Centroid Vector가 도출한다. 다음으로 Euclidean Distance를 활용하여 Centroid Vector로부터 각각의 클러스터 내 플로우 벡터간의 거리를 계산하고, 그 중에서 최솟값을 Threshold로 설정한다.

이 때, 클러스터 수에 해당하는 k는 다양하게 설정할 수 있으며, 1부터 점차적으로 증가시키면서, 수집된 모든 GT 트래픽 셋을 포함하는 k를 최적의 클러스터 수로 설정한다. 예를 들어 로그인 행위에 대해 10개의 GT Traffic Trace가 있을 경우, k 값을 1부터 증가시키면서 10 개의 Trace를 입력으로 클러스터링을 수행한다. k 값에 따라 도출되는 Centroid Vector와 Threshold가 모든 GT Traffic Trace의 플로우를 최소 1개 이상 포함하는 경우의 k 값을 최적의 클러스터 수로 설정한다.

마지막으로 최적의 k 값에 따라 도출된 클러스터를 대상으로 Centroid Vector와 Threshold를 통계 시그니처로 정의한다. 클러스터링을 활용한 통계 시그니처를 도출하는 과정은 그림 4에 나타나있다.

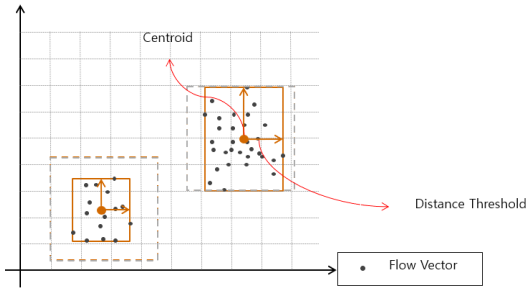


그림 4. 클러스터링을 활용한 공통 통계 시그니처 도출 과정
Fig. 4. A Process of Common Statistical Signature Extraction using Clustering

대상 행위에 대한 탐지 시그니처를 정의 할 때, 하나의 행위에서 여러 가지 공통 시그니처가 나올 수 있다. 예를 들어, 응용 A의 시작 행위를 탐지 할 때, "*1.x.x.x*", "*TCP*", "*443*"과 같이 특정 IP, Port, Protocol의 헤더 시그니처와 "*abcdefg*"의 SNI 시그니처가 함께 나올 수 있다. 해당 경우에는 검증 과정을 통해 정의된 시그니처가 대상 응용의 행위를 정확하게 탐지 하는지 평가한다. 또한, 탐지 규칙 생성 과정에서 잘못된 공통 시그니처가 추출될 경우도 있으며, 이 경우에도 마찬가지로 규칙 검증 과정에서 생성된 규칙에 대한 검증 및 평가를 통해 잘못 생성된 규칙을 판별하고 제거한다.

3.5 시그니처 검증 (Signature Verification)

시그니처 검증 과정은 생성된 타입 별 시그니처를 대상으로 검증하는 과정이다. 검증 과정은 대상 행위랑 동일한 트래픽과 다른 행위의 트래픽 셋이 입력으로 들어간다. 대상 행위랑 동일한 트래픽의 경우, 시그니처가 트래픽을 포함하여 제대로 탐지가 되는지를 확인하며, 다른 행위의 트래픽의 경우 탐지가 안되는지 확인한다. 예를 들어, 응용 A의 로그인 시그니처의 경우, 같은 로그인 행위 트래픽을 넣었을 때, 시그니처가 해당 트래픽을 커버할 경우, 시그니처가 제대로 추출되었다고 판단한다. 또한, 로그인 시그니처를 대상으로 로그아웃 행위 트래픽을 넣었을 때, 시그니처가 해당 트래픽을 커버하지 못할 경우, 마찬가지로 시그니처가 제대로 추출되었다고 판단한다. 두 가지 과정 중에 한 가지라도 만족을 못할 경우에는 대상 시그니처를 폐기한다.

3.6 규칙 취합 (Rule Aggregation)

규칙 취합 과정은 각 행위 별로 생성된 규칙에서 중복되거나 불필요한 규칙을 제거하고, 하나의 통합된 규칙으로 취합하는 과정이다. 대상 응용의 4 가지 행위(시작, 로그인, 로그아웃, 종료) 별로 생성된 규칙들을 대상

응용에 대한 하나의 규칙으로 취합한다.

3.7 행위 탐지 (Action Detection)

생성된 규칙을 평가하기 위하여 행위 탐지 과정을 수행하며, 수집된 트래픽과 행위 정보가 기록된 로그 파일, 생성된 규칙을 대상으로 행위를 정확하게 탐지하는지 평가하여 잘못 생성된 규칙을 판별한다.

행위 탐지는 탐지 규칙 내 헤더, SNI, 통계 시그니처 순으로 매칭을 통해 수행된다. 먼저 헤더 시그니처 매칭을 수행하며, 헤더 시그니처가 있을 경우, 헤더 시그니처에 대한 매칭을 수행하고, 헤더 시그니처가 없거나 매칭이 되지 않을 경우, SNI 시그니처 매칭이 수행된다. 같은 방법으로 SNI 시그니처가 있을 경우, SNI 시그니처에 대한 매칭을 수행하고, SNI 시그니처가 없거나 매칭이 되지 않을 경우, 통계 시그니처 매칭이 수행된다. 시그니처 매칭을 활용한 행위 탐지 과정은 그림 5에 나타나있다.

행위 탐지 과정은 트래픽 수집 과정에서 수집된 트래픽 셋과 기록된 로그 파일을 활용하며, 테스트용 트래픽 셋을 입력으로 호스트 IP, 대상 응용의 행위, 탐지 시간을 실제 기록과 비교하여 정탐지(True Detection), 오탐지(False Detection)로 구분한다. 구분하는 과정에 대한 예시는 그림 6에 나타나있으며, 실제 기록과 탐지 결과가 모두 정확하게 일치하면 정탐지로 정의하고, 탐지가 되지 않거나 한 가지 항목이라도 잘못 탐지 될 경우를 오탐지로 정의한다.

검증 과정에는 Recall, Precision, F-measure의 3 가지 평가 지표를 사용하며, 각 지표에 대한 계산 과정은 수식 (1), (2), (3)에 나타나있다.

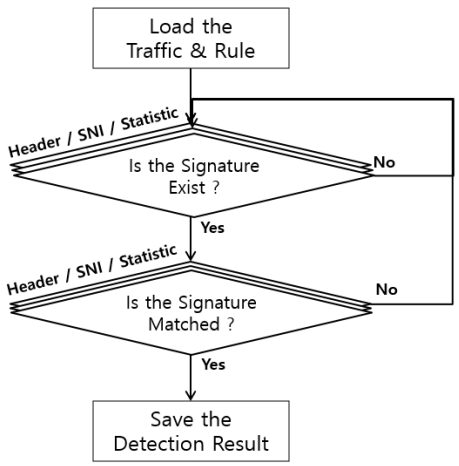


그림 5. 시그니처 매칭을 활용한 행위 탐지
Fig. 5. Action Detection using Signature Matching

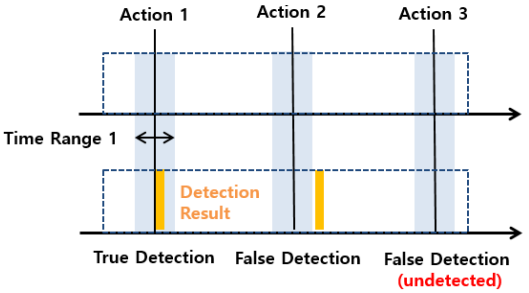


그림 6. 규칙 검증 과정 내 탐지 결과 예시
Fig. 6. An Example of Detection Results in Rule Verification Process

검증 과정은 수집 된 테스트용 트래픽 셋, 행위 기록과 생성된 규칙을 대상으로 정탐지, 오탐지, 미탐지의 탐지 결과를 도출한다. 마지막으로 도출된 탐지 결과를 활용하여 평가 지표를 계산한다. 정탐지는 실제 기록과 동일하게 탐지를 한 것으로 True Positive (TP)와 True Negative (TN)으로 나타내며, 오탐지는 실제 기록과 다르게 탐지한 것으로 False Positive (FP)와 실제 기록이 있어도 탐지를 못한 False Negative (FN)로 구성된다.

$$Recall = \frac{TP}{TP + FP} \quad (1)$$

$$\mathrm{Precision} = \frac{TP}{TP + FN} \quad (2)$$

$$F-measure = \frac{2 \times Precision \times Recall}{\mathrm{Precisio} + Recall} \quad (3)$$

Ⅳ. 실 험

본 논문에서 제안하는 시스템의 타당성을 검증하기 위해 실험을 진행한다. 실험은 SNI 정보에 대한 높은 의존도 문제와 이를 해결하기 위해 통계 정보를 추가하였을 때, 탐지 성능에 미치는 영향을 살펴보는 것을 목적으로 수행하며, 크게 두 가지 형태로 진행한다.

첫 번째 실험은 통계 시그니처만 활용하여 통계 시그니처의 전반적인 탐지 성능 확인하며, 두 번째 실험은 그림 5의 시그니처 적용 방법을 사용하여 선행 연구[5]와 제안하는 방법의 탐지 성능을 비교한다. 선행 연구는 제안하는 방법과 마찬가지로 규칙 기반의 행위 탐지 방법이지만, 헤더, SNI 정보만 사용하기 때문에 SNI 정보에 대한 의존도가 높은 방법이다.

선행 연구와 제안하는 방법의 탐지 성능을 비교하기 위해 동일한 데이터 셋을 적용하여 탐지 규칙을 생성하

며, 생성된 규칙을 활용하여 행위 탐지를 수행한다. 최종적으로 두 가지 방법에서 도출되는 탐지 결과를 비교하여 탐지 성능을 평가한다.

4.1 데이터 셋

실험에는 Autodesk 응용을 대상으로 선정하였으며, AutoCAD LT 2023 라이선스를 사용하여 수집하였다. 데이터 셋은 규칙 생성(Rule Generation)을 위해 10 Trace의 트래픽 셋과 규칙 검증(Rule Verification)을 위해 10 Trace의 트래픽 셋을 수집하였다. 수집된 트래픽 셋에 대한 정보는 표 2에 나타나있으며, 두 번째 실험에서 수행하는 두 가지 방법 모두 동일한 트래픽 셋을 입력으로 진행하였다.

규칙 생성을 위한 10 가지 Trace에는 각 Trace 별로 4 가지 행위가 모두 포함되어 있으며, 전처리 과정에서 각 행위 별로 트래픽을 분할하여 사용한다. 규칙 검증을 위한 10 가지 Trace에서는 각 Trace 별로 4 가지 행위를 반복하였으며, 각 Trace 별 탐지 결과는 전체 40회 수행된 행위를 대상으로 도출된 정탐지(TP, TN), 오탐지(FP), 미탐지(FN)를 활용하여 Recall, Precision, F-measure, Accuracy를 계산한다. 선행 연구에서는 다른 행위가 섞여있지 않고 대상 행위만 수행한 트래픽을 사용하기 때문에, 정탐지에 True Negative (TN)이 없었지만, 제안하는 방법의 검증에는 하나의 Trace에 다른 행위가 섞여 있기 때문에, 탐지 결과로 TN과 평가 지표로 Accuracy가 추가되었다.

표 2. 트래픽 데이터 셋에 대한 정보
Table 2. An Information of the Traffic Data Set

수행 과정	App.	Tr.	트래픽 정보		
			Flow	Pkt.	Byte
규칙 생성	Autodesk	#1	1,471	6,718	1,293,630
		#2	548	3,394	1,108,537
		#3	1,034	4,632	1,363,959
		#4	1,377	9,069	5,272,284
		#5	1,449	1,0521	5,896,061
		#6	1,251	5,626	1,618,614
		#7	1,377	9,063	5,272,284
		#8	1,510	10,441	6,898,861
		#9	1,051	5,001	1,222,614
		#10	1,277	8,851	3,211,084
테스트	Autodesk	#1	611	35,516	29,320,256
		#2	538	4,956	1,793,377
		#3	1,014	8,565	3,029,158
		#4	685	14,658	10,215,861
		#5	626	5,122	1,840,231
		#6	842	9,061	2,997,251
		#7	763	6,561	2,274,159
		#8	851	56,190	55,916,543
		#9	628	5,708	2,059,036
		#10	743	6,346	2,247,152

규칙 생성 과정에서 수집된 Autodesk 트래픽 데이터 셋을 활용하여 규칙을 생성하며, 생성된 규칙에 대한 예시는 표 3에 나타나있다. 먼저 대상 행위는 로그인으로, 헤더 시그니처에는 공통적인 특징이 나타나지 않았으며, SNI 시그니처에는 각 행위 별로 여러 개의 SNI 정보가 도출 되었다. 또한, 3 가지 SNI 정보 다음으로 [and, 3, 3]로 나타나있으며, 이는 3 가지 SNI 정보가 모두 도출되는 경우에 행위를 탐지 하는 것을 나타낸다. 마지막으로 통계 시그니처는 k=14, N=5로 설정한 PSD의 Centroid Vector와 Threshold가 시그니처로 나타나 있다.

하지만 Autodesk를 분석 하였을 때, 몇 가지 고려할 사항이 있다. 첫 번째로 Autodesk는 종료를 하더라도 백그라운드에서 동작하기 때문에 완전히 응용이 종료되었다고 보기 어렵다. 따라서 Autodesk의 종료 행위 탐지 시점은 로그아웃 행위와 동일하게 정의하였다. 두 번째로 로그아웃과 종료 행위에서 도출된 트래픽 분석을 하였을 때, 종료 행위에서 추가적으로 플로우가 발생하지 않았으며, 플로우 내 패킷 길이가 불규칙적이고, 값의 편차가 크기 때문에 통계 시그니처를 적용하지 않고, 선행 연구[5] 행위 탐지 방법을 적용한다. 따라서 탐지 실험 (1)에서는 네 가지 행위 중 시작과 로그인 행위에 대해서는 헤더, SNI, 통계 시그니처를 모두 사용하였으며, 로그아웃과 종료 행위에 대해서는 헤더, SNI 시그니처만 사용하였다.

표 3. Autodesk 행위 규칙 예시
Table 3. An Example of Autodesk Action Rule

행위	시그니처 종류	내용
Login	Header	-
	SNI	[accounts.autodesk.com, auth.autodesk.com, app.autocad360.com],
		[and, 3, 3]
	Statistical	PSD: [575.9321, -64, -1501.43, -1500.3, 58]
		Threshold : 4.236682

4.2 탐지 성능 평가 (1)

첫 번째 실험의 탐지 성능 평가 결과는 표 4와 그림 7에 나타나있으며, 10 가지의 규칙 검증 트래픽 셋을 입력으로 행위 탐지를 수행한다. 탐지 결과는 행위 별로 전체 Trace에 대한 정탐지(TP, TN), 오탐지(FP), 미탐지(FN)를 계산하고, 이를 활용하여 행위 별 평균 Recall, Precision, F-measure, Accuracy로 나타낸다.

응용 시작은 모두 제대로 탐지하였지만, 추가적으로 2건의 오탐지가 발생하고, 약 90.91의 F-measure와 95%의 Accuracy가 도출되었으며, 로그인은 2건의 미탐지가 발생하고, 약 88.89의 F-measure와 95%의 Accuracy가 도출되었다. 종료 및 로그아웃의 경우 선행 연구 방법을 적용하였기 때문에 2건의 오탐지와 3건의 미탐지가 발생하였으며, 약 73.69의 F-measure와 87.5%의 Accuracy가 도출되었다. 탐지 성능 결과를 보면 통계 시그니처를 적용하는 경우가 전반적으로 탐지 성능이 높게 나타났다.

표 4. 행위 탐지 실험 결과 (1)
Table 4. Detection Result of the Experiment (1)

행위	Tr.	Detection Result (1)			
		TP	TN	FP	FN
		Recall	Precision	F-measure	Accuracy
시작	all	10	28	2	0
		100%	83.33%	90.91	95%
로그인	all	8	30	0	2
		80%	100%	88.89	95%
로그아웃	all	7	28	2	3
		70%	77.78%	73.69	87.5%
종료	all	7	28	2	3
		70%	77.78%	73.69	87.5%

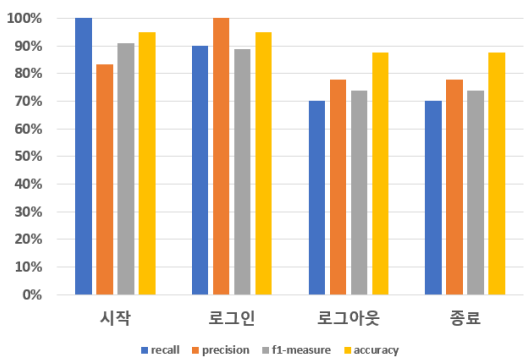


그림 7. 행위 탐지 실험 결과 (1)
Fig. 7. Detection Result of the Experiment (1)

4.3 탐지 성능 평가 (2)

두 번째 실험의 탐지 결과는 표 5와 그림 8에 나타나 있으며, 10 가지의 규칙 검증 트래픽 셋을 입력으로 행위 탐지를 수행한다. 탐지 결과는 첫 번째 실험과 마찬가지로 각 방법에 대한 전체 Trace의 평균 탐지 성능을 계산하여 나타낸다.

선행 연구 방법의 탐지 성능은 평균 79~83%의 Recall, Precision이 도출되며, 제안하는 방법의 탐지 성능은 평균 89~94%의 Recall, Precision이 도출된다. 특히 선행 연구에서는 Adobe Creative Cloud를 대상으로 평균 96~100%의 Recall, Precision이 도출되었지만, 동일한 방법을 Autodesk를 대상으로 적용 할 때, 성능이 감소한 것을 알 수 있다. 이는 대상 응용에 따라 SNI 시그니처가 도출되지 않거나 잘못 도출 될 수 있으며, SNI 의존도가 높은 행위 탐지 방법을 적용 할 경우 탐지 성능이 크게 떨어질 수 있음을 보여준다.

탐지 결과를 분석해보면 선행 연구에서 제안하는 방법보다 오탐지와 미탐지가 상대적으로 많이 발생하는 것을 알 수 있다. 먼저 전체적으로 정의된 헤더 및 SNI 시그니처가 잘못된 경우에 오탐지가 가장 많이 발생하였다. 또한 특정 Trace에서 SNI 시그니처 일부가 달라질 경우 미탐지가 발생한다. 예를 들어, "xxx.123.com"의 SNI 시그니처에서 실제 트래픽에서 "xxx.124.com",

표 5. 행위 탐지 실험 결과 (2)
Table 5. Detection Result of the Experiment (2)

방법	Tr.	Detection Result (2)			
		Recall	Precision	F-measure	Accuracy
선행 연구[5]	all	79.14%	88.32%	80.82	81.4%
제안하는 방법	all	94.55%	91.57%	93.04	95.28%

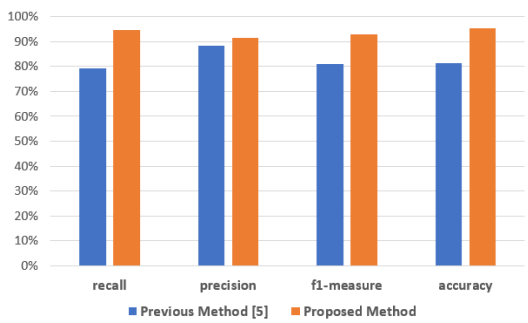


그림 8. 행위 탐지 실험 결과 (2)
Fig. 8. Detection Result of the Experiment (2)

혹은 “xxx.123t.com”의 SNI가 발생 할 경우에 시그니처가 일치하지 않기 때문에 탐지 할 수 없다. 이러한 경우에 선행 연구에서는 미탐지로 판단을 하지만, 제안하는 방법에서는 통계 시그니처를 한 번 더 적용하여 미탐지를 줄인다.

제안하는 방법은 선행 연구에 비해 더 높은 탐지 성능을 보이지만, 몇 가지 개선 사항 존재한다. 먼저 제안한 방법의 탐지 성능은 평균 89~94%의 Recall, Precision로 나타났으며, 이는 생성된 규칙에 부분적으로 잘못된 시그니처가 포함된 것으로 보인다. 따라서 행위 탐지 규칙 생성을 위한 공통 특징 추출 알고리즘과 생성된 규칙에 대한 검증 과정에 대한 개선이 필요하다고 판단된다. 특히 제안하는 방법에서 발생한 오탐지에는 로그인 행위의 비율이 높았으며, 본 연구에서는 탐지 규칙을 생성 할 때, 자동 로그인 상황은 고려하지 않아서 나타나는 현상으로 보인다. 자동 로그인 현상은 Autodesk 응용에서 로그인 행위 이후에 로그아웃을 하지 않고 응용 종류 및 시작 행위를 수행하면 자동으로 로그인이 되어있는 상태를 나타내며, 이에 대한 트래픽 분석을 수행 한 후에 해당 상황에 대한 시그니처를 정의한다면 해결 할 수 있을 것으로 판단된다.

V. 결 론

본 논문에서는 행위 탐지 연구에 대한 필요성을 언급하고,. 본 연구의 선행 연구를 포함한 여러 가지 관련 연구를 소개한다. 또한 선행 연구의 SNI 정보에 대한 의존도가 너무 높다는 문제점을 제시하고, 제시한 문제점을 해결하기 위해 트래픽 통계 정보를 활용한 규칙 생성 방법을 제안한다. 제안하는 방법은 헤더, SNI, 통계 정보를 사용하기 때문에 기존 방법보다 SNI 정보에 대한 의존도를 줄일 수 있다.

제안하는 방법을 검증하기 위해 Autodesk 응용을 대상으로 선행 연구와 탐지 성능에 대한 비교 실험을 진행하였으며, 선행 연구에 비해 높은 탐지 성능을 보인다. 특히, 선행 연구에서는 Adobe Creative Cloud를 대상으로 약 96% 이상의 F-measure로 높은 탐지 성능을 보였지만, Autodesk는 약 80%의 F-measure로 제안하는 방법의 91%의 탐지 성능에 비해 낮은 탐지 성능을 보였다, 이는 선행 연구의 방법은 SNI 정보에 대한 높은 의존도로 인해 특정 응용을 대상으로는 성능이 저하되는 것을 보여준다.

하지만 4.2장 탐지 성능 평가에서 설명하였듯이 실험에는 Autodesk의 자동 로그인을 고려하지 않고 수동 로그인만 고려하여 규칙을 생성하였기 때문에 로그인 행위에 대한 오탐지가 높게 발생하였다. 또한 생성된 규칙에 잘못된 시그니처가 포함되어 있기 때문에, 오탐지와 미탐지가 일부 발생한 것으로 보인다.

따라서 향후 연구로 규칙 생성 방법과 알고리즘을 고도화 시켜 탐지 성능을 더욱 향상 시킬 예정이며, 특히 본 연구에서 고려하지 않은 자동 로그인 상황에 대해서도 규칙을 생성하여 검증 할 예정이다. 또한 다른 응용을 대상으로도 추가 실험을 수행할 예정이며, 특히 다른 연구에서 사용한 응용을 동일하게 사용하여 탐지 성능 비교를 할 예정이다.

References

[1] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, “Active learning for network traffic classification: A technical study,” in *IEEE Trans. Cognitive Commun. and Netw.*, vol. 8, no. 1, pp. 422-439, 2022. (https://doi.org/10.1109/TCCN.2021.3119062).

[2] C. Hou, J. Shi, C. Kang, Z. Cao, and X. Gang, “Classifying user activities in the encrypted WeChat traffic,” *2018 IEEE 37th IPCCC*, pp. 1-8, 2018. (https://doi.org/10.1109/PCCC.2018.8711267)

[3] K. Park and H. Kim, “Encryption is not enough: Inferring user activities on Kakaotalk with traffic analysis,” *WISA*, pp. 254-265, Springer, Cham, 2015. (https://doi.org/10.1007/978-3-319-31875-2_21)

[4] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, “Analyzing android encrypted network traffic to identify user actions,” in *IEEE Trans. Inf. Forensics and Secur.*, vol. 11, no. 1, pp. 114-125, 2016. (https://doi.org/10.1109/TIFS.2015.2478741)

[5] J. -T. Park, et al., “Rule-based user action detection system for saas application,” *2022 23rd APNOMS*, pp. 1-4, 2022. (https://doi.org/10.23919/APNOMS56106.2022.9919933)

[6] H.-M. An, et al., “Traffic identification based on applications using statistical signature free from abnormal TCP action,” *J. Inf. Sci. and Eng.*, vol. 31, no. 5, pp. 1669-1692, 2015. (https://doi.org/10.6688/JISE.2015.31.5.10)

[7] S.-H. Yoon, J.-S. Park, B. D. Sija, M.-J. Choi, and M.-S. Kim, "Header signature maintenance for Internet traffic identification," *Int. J. Network Manag.*, vol. 27, no. 1, pp. 1-15, 2017. (https://doi.org/10.1002/nem.1959)

[8] S. Lee, et al., "LARGen: Automatic signature generation for malwares using latent dirichlet allocation," in *IEEE Trans. Dependable and Secure Comput.*, vol. 15, no. 5, pp. 771-783, 2018. (https://doi.org/10.1109/TDSC.2016.2609907.)

[9] K.-S. Shim, Y.-H. Goo, D. Lee, and M.-S. Kim, "Automatic payload signature update system for the classification of dynamically changing internet applications," *KSII Trans. Internet and Inf. Syst. (TIIS)*, vol. 13, no. 3, pp. 1284-1297, 2019. (https://doi.org/10.3837/tiis.2019.03.009)

[10] R. Zhang, M. Tong, L. Chen, J. Xue, W. Liu, and F. Xie, "CMIRGen: Automatic signature generation algorithm for malicious network traffic," *2020 IEEE 19th Int. Conf. Trust, Secur. and Privacy in Comput. and Commun. (TrustCom)*, pp. 736-743, 2020. (https://doi.org/10.1109/TrustCom50675.2020.00101)

[11] K.-S. Shim, Y.-H. Goo, S.-H. Lee, B. D. Sija, and M.-S. Kim, "Automatic payload signature update system for classification of recent network applications," *J. KICS*, vol. 42, no. 1, pp. 98-107, 2017. (https://doi.org/10.7840/kics.2017.42.1.98)

[12] S. H. Kim and S. C. Lee, "Automatic malware detection rule generation and verification system," *J. Korean Soc. for Internet Inf.*, vol. 20, no. 2, pp. 9-19, 2019. (https://doi.org/10.7472/jksii.2019.20.2.9)

박 지 태 (Jee-Tae Park)

2017년 : 고려대학교 컴퓨터정보학과 학사

2017년~현재 : 고려대학교 컴퓨터정보학과 석박사통합과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

백 의 준 (Ui-Jun Baek)

2018년 : 고려대학교 컴퓨터정보학과 학사

2018년~현재 : 고려대학교 컴퓨터정보학과 석박사통합과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

신 창 의 (Chang-Yui Shin)

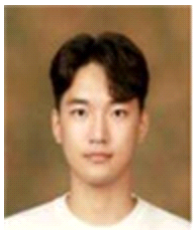
2003년 : 육군사관학교 운영분석학과 학사

2007년 : 고려대학교 전자컴퓨터 공학과 석사

2022년~현재 : 고려대학교 컴퓨터정보학과 박사과정

<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 분석

최 정 우 (Jung-Woo Choi)



2022년 : 고려대학교 컴퓨터정보학과 학사
2022년~현재 : 고려대학교 컴퓨터정보학과 석사과정
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석

호 태 규 (Tae-Gyu Ho)

2017년 2월 : 한성대학교 멀티미디어공학과 졸업
2020년 8월 : 성균관대학교 빅데이터학과 석사
<관심분야> 웹 공학, 데이터 분석, 빅데이터

홍 윤 환 (Yun Hwan Hong)

1991년 : 울산대학교 건축학과
2000년 5월~현재 : (주)닥터소프트 법인 설립-대표이사
<관심분야> 소프트웨어 관리, 웹 공학, 데이터 분석, 빅데이터

김 명 섭 (Myung-Sup Kim)

1998년 : 포항공과대학교 전자계산학과 학사
2000년 : 포항공과대학교 전자계산학과 석사
2004년 : 포항공과대학교 전자계산학과 박사
2006년 : Dept. of ECS, Univ of Toronto Canada
2006년~현재 : 고려대학교 컴퓨터정보학과 교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모니터링 및 분석, 멀티미디어 네트워크