J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr
Copyright ⓒ 2019 KSII

악성코드 침입탐지시스템 탐지규칙 자동생성 및 검증시스템☆

Automatic Malware Detection Rule Generation and Verification System

김 성 호[1] 이 수 철[2*]
Sungho Kim Suchul Lee

요 약

인터넷을 통한 서비스 및 사용자가 급격하게 증가하고 있다. 이에 따라 사이버 공격도 증가하고 있으며, 정보 유출, 금전적 피해 등이 발생하고 있다. 정부, 공공기관, 회사 등은 이렇게 급격한 사이버 공격 중 알려진 악성코드에 대응하기 위하여 시그니처 기반의 탐지규칙을 이용한 보안 시스템을 사용하고 있지만, 시그니처 기반의 탐지규칙을 생성하고 검증하는 데 오랜 시간이 걸린다. 이런 문제를 해결하기 위하여 본 논문에서는 잠재 디리클레 할당 알고리즘을 통한 시그니처 추출과 트래픽 분석 기술 등을 이용하여 시그니처 기반의 탐지규칙 생성 및 검증 시스템을 제안하고 개발하였다. 개발한 시스템을 실험한 결과, 기존보다 훨씬 신속하고, 정확하게 탐지규칙을 생성하고 검증하였다.

☞ 주제어 : 악성코드, 탐지규칙, 스노트, LDA, 네트워크 위협

ABSTRACT

Service and users over the Internet are increasing rapidly. Cyber attacks are also increasing. As a result, information leakage and financial damage are occurring. Government, public agencies, and companies are using security systems that use signature-based detection rules to respond to known malicious codes. However, it takes a long time to generate and validate signature-based detection rules. In this paper, we propose and develop signature based detection rule generation and verification systems using the signature extraction scheme developed based on the LDA(latent Dirichlet allocation) algorithm and the traffic analysis technique. Experimental results show that detection rules are generated and verified much more quickly than before.

☞ keyword : Malware, Detection rule, Snort, LDA, network threat

1. 서 론

인터넷을 통한 초 연결사회가 도래하였다. 현대 사회에서 인터넷은 일상생활에서 가장 쉽게 접할 수 있는 거대한 정보 공간이 되었다. 기업들과 개인은 인터넷을 통한 전자상거래, 동영상 스트리밍 서비스, 소셜 네트워크 서비스(SNS) 등 다양한 서비스를 제공하거나 받고 있다. 그리고 스마트폰 태블릿 PC 등의 스마트 기기의 대중화로 사용자는 언제, 어디서나 인터넷을 접속하여 다양한 서비스를 사용할 수 있게 되었다. 이런 인터넷은 편리한 환경에서 다양한 정보를 제공해 주고 있다. 그러나 인터넷을 통해 제공되는 서비스와 인터넷 사용자들이 증가함에 따라 서비스를 제공하는 업체와 서비스를 받는 사용자를 대상으로 악의적인 공격 또한 급격하게 증가하였다.

세계적인 보안 기업 카스퍼스키 연구소(Kaspersky Lab)에서 발표한 보고서에 따르면 2017년에 매일 360,000개의 신규 악성코드들이 발견되었다[1]. 이는 2016년 대비 11.5%나 증가한 수치이다. 그리고 맥아피 연구소(McAfee Lab)에서 발표한 보고서에 따르면 2017년 4분기에만 신규 악성코드가 6천 3백만 개 증가하였으며, 현재까지 발견된 신규 악성코드는 7억 개에 달하고 있다[2]. 그리고 이런 악성코드들은 전산망 마비, 개인정보 및 주요 정보 유출, 금전적 피해를 발생시키고 있다.

인터넷을 통하여 다양한 서비스를 제공하고 있는 기업과 공공기관 등은 이런 악성코드를 통한 사이버 공격에 대응하기 위하여 방화벽, 침입탐지시스템(IDS), 침입방지

1 Security Technology Research Division, National Security Research Institute, Daejeon, 34044, Korea
2 Dept. of Computer Science and Information Engineering, Korea National University of Transportation, Uiwang, Kyunggi, 16106, Korea
* Corresponding author (sclee@ut.ac.kr)
[Received 21 September 2018, Reviewed 15 October 2018(R2 5 December 2018), Accepted 14 January 2019]
☆ 이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2017R1C1B5017028). 이 연구는 2019년 한국교통대학교 지원을 받아 수행하였음.

http://dx.doi.org/10.7472/jksii.2019.20.2.9

시스템(IDS), 웹 방화벽(WAF) 등의 다양한 네트워크 보안 시스템을 도입하여 운영하고 있다. 다양한 AI 기반, 행동 기반 악성코드 탐지기술이 활발히 연구되고 있음에도 불구하고 대부분의 보안 시스템들은 알려진 악성코드 탐지에 있어 높은 탐지 성능을 자랑하는 시그니처 기반의 탐지기법을 사용한다.

시그니처 기반의 탐지기법은 탐지규칙이라 불리는 악성코드에 대한 추상화된 정의를 필연적으로 내포한다. 탐지규칙을 효과적으로 정의하기 위해서는 네트워크, 보안, 운영체제 등에 대한 전문적인 지식이 필요하다. 부적절한 탐지규칙의 운용은 수많은 오탐(false positive)을 발생시켜 보안 시스템 성능 저하 등을 유발하고 심지어는 네트워크 보안 시스템이 설치된 네트워크 전체 망을 마비시킬 수도 있다.

이러한 보안 분야 특성을 고려하여 CERT(Computer Emergency Response Team)팀 등 보안 전문가가 시그니처 기반의 탐지규칙을 생성한다. 전문가에 의해서 생성된 탐지규칙은 정확성이 높고, 잘못되거나 비효율적으로 생성될 가능성이 적다는 장점이 있지만, 탐지규칙을 생성하고 검증하는 데 많은 시간이 소요된다는 단점이 있다.

현재 전문가에 의한 탐지규칙 생성 검증 절차는 일반적으로 시그니처 분석, 탐지규칙 생성, 탐지/차단 테스트, 보안장비 엔진파트 검토, 실망 테스트의 과정으로 대략 2~3일 걸리는 것으로 알려져 있다[21].

본 논문에서는 급격히 증가하고 있는 악성코드에 효율적으로 대응하기 위하여 보안 전문가뿐만 아니라 준전문가들도 신속하고 정확하게 탐지규칙을 생성하고 검증할 수 있는 시스템을 제안한다. 제안 기법은 기존에 수행하였던 시그니처를 추출한 연구[21]와 트래픽 분석 결과를 이용하여 스노트 탐지규칙을 자동으로 생성하고, 가상 환경의 IDS 서버를 구축하여 생성한 탐지규칙을 검증하는 것이다.

본 논문의 공헌은 다음과 같다. 첫째, [21] 알고리즘을 활용하여 실제로 정보보안 체계에서 사용 가능한 시스템을 제안하고, 구현하였다. 둘째, 기존 탐지규칙 생성과 검증 방식과 비교하여 탐지규칙 생성 작업 시간을 획기적으로 줄일 수 있는 자동화 시스템을 제안하고, 구현하였다. 비 보안 전문가는 제안하는 시스템을 통해 적절한 탐지규칙을 자동으로 생성하고, 생성한 탐지규칙의 성능을 검증할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 시그니처 기반의 대표적인 탐지규칙인 스노트[3] 탐지규칙 문법을 분석하여 탐지규칙을 생성하고 검증하는 데 필요한 문제를 확인하고 관련 연구를 요약하였다. 3장에 본 논문에서 사용하는 기계학습 알고리즘 관련 배경 지식을 간략히 설명하고 해당 알고리즘을 활용하는 방안을 설명한다. 4장에서는 제안하는 시스템의 전체적인 구조 및 방법을 설명하고, 5장에서 제안한 시스템의 성능을 결과를 보여주며, 6장에서 결론을 기술하였다.

2. 관련 연구

2.1 시그니처 기반 탐지규칙

시그니처 기반 탐지규칙 중 세계적으로 가장 많이 사용되고 있으며, TTA 표준[4]으로 채택되어 국내 네트워크 보안 장비에서 대부분 지원하는 스노트 탐지규칙을 분석하였다. 본 논문에서는 채택된 TTA 표준을 기반으로 스노트 탐지규칙을 자동으로 생성하고 검증하는 시스템을 설계하는데 필요한 요구사항을 정의하였다.

스노트 탐지규칙은 논리적으로 탐지규칙 헤더와 탐지규칙 옵션으로 구분되어 있다. 탐지규칙 헤더는 표 1과 같이 rule action, protocol, source/destination IP(netmask), source/destination port number, direction operator 정보로 구성된다. 탐지규칙 옵션은 크게 general, payload detection, non-payload detection, post-detection으로 구성된다.

General 탐지규칙 옵션은 탐지규칙에 관련된 정보를

(표 1) 스노트 탐지규칙 구성 및 예
(Table 1) Snort rule syntax and examples

탐지규칙	Rule Header							Rule Options
구성	Rule Action	Protocol	Source IP/Netmask	Source Port Numbers	Direction Operator	Destination IP/Netmask	Destination Port Numbers	General, Payload Detection, Non-Payload Detection, Post-Detection
의미	처리 방법	프로토콜	송신자 IP주소	송신자 포트번호	패킷 방향	수신자 IP주소	수신자 포트번호	옵션
예제	alert	tcp	any	any	->	192.168.1.0/24	111	Content:"|00 01 86 a5|"; msg:"mountd access";

입력할 때 사용되며, 패킷을 탐지하는 것과는 연관이 없다. Payload detection 탐지규칙 옵션은 패킷 페이로드의 정보를 확인하여 탐지하기 위한 설정값들을 지정한다. Non-payload detection 탐지규칙 옵션은 패킷에서 페이로드 이외의 정보를 확인하여 탐지하기 위한 설정값들을 지정한다. 마지막으로 post-detection 탐지규칙 옵션은 탐지규칙과 일치하는 패킷을 탐지한 이후에 동작하는 설정값들을 지정한다.

탐지규칙 헤더에서 각각의 구성은 공백으로 구분하고, 탐지규칙 헤더와 옵션은 소괄호로, 탐지규칙의 옵션과 옵션은 세미콜론으로 구분한다. 옵션은 옵션 키와 옵션값으로 구성되는데 콜론으로 구분한다. 표 1의 예제 탐지규칙을 문법에 맞게 작성하면 다음과 같다. “alert tcp any any -> 192.168.1.0/24 111 (content:|00 01 86 a5|; mag:”mounted access“;)”

스노트 탐지규칙을 생성하는 데 가장 중요한 것은 정상 패킷 페이로드에는 포함되어 있지 않고 악성 패킷 페이로드에만 포함된 악성 시그니처를 추출하는 것과 악성 시그니처를 포함하고 있는 패킷의 IP, Port, 프로토콜 등 패킷 세부정보 분석이다. 본 논문에서는 [21]에서 제안한 악성 시그니처 정보 추출기법을 활용하여 payload detection 탐지규칙 옵션을 자동으로 생성할 수 있는 시스템을 제안한다. 나아가 제안시스템은 분석한 패킷 상세정보를 이용하여 탐지규칙 헤더와 non-payload detection 탐지규칙 옵션을 자동으로 생성한다.

그리고 탐지규칙을 분석한 결과 많은 옵션의 사용은 탐지하는 트래픽에 따라 탐지 성능을 높일 수도 있지만, 반대로 오탐률을 증가시킬 수도 있다. 그래서 자동으로 생성되는 옵션은 최소화하고, 사용자가 임의로 추가할 수 있도록 구현하였다.

2.2 악성 시그니처 추출 연구

악성코드 또는 악성 트래픽에서 자동으로 악성 시그니처를 추출하는 연구가 많이 진행됐다. [5]는 메타스플로잇(Metasploit)를 이용하여 생성된 웜에서 발생하는 트래픽에서 악성 시그니처를 추출하기 위하여 두 개의 허니팟을 구축한 후, 송수신하는 트래픽에서 LCS(Longest Common Substring) 알고리즘을 이용하였다. [6][7]은 악성코드에서 사용되는 함수를 분석하여 시그니처를 추출하였는데 [6]은 함수의 엔트로피 분석을 통한 시그니처 추출, [7]은 유클리드 거리를 이용한 확률기법 및 엔트로피를 이용하여 시그니처를 추출하였다. [8]은 악성코드를 48바이트씩 나눈 후 이미 알려진 악성코드의 시그니처와 인접한 바이트 시퀀스를 비교하는 방식으로 변형된 시그니처를 추출하였다. [9]는 다형성 웜에서 발생하는 트래픽을 기존 웜에서 발생한 트래픽들과 CSS(Colored Set Size) 알고리즘을 이용하여 시그니처를 추출하였다.

최근 연구로는 [26]은 악성트래픽에서 비지도학습기반의 Segmentation 알고리즘을 이용하여 분할 한 시그니처의 엔트로피 정보를 이용한 Voting Experts 알고리즘과 Ranking 알고리즘을 이용하여 시그니처를 추출, [27]은 심층 신뢰 신경망(DBN)에서 Denising Auto-Encoder 알고리즘을 이용하여 쿠쿠샌드박스의 악성코드 분설 결과를 이용하여 시그니처를 추출, [28]은 악성코드를 동적으로 분석하면서 추출된 복호화 키, URL 등을 추출하여 시그니처로 사용하였다.

그 외의 연구로는 공격에 사용될 수 있는 취약점 시그니처를 추출하는 연구[10]가 있다. 이 취약점 시그니처는 해당 악성 트래픽이 악용하는 취약점을 직접적으로 탐지규칙의 시그니처로 활용한다. 따라서 악성코드 탐지에 대한 근본적인 문제인 탐지율을 극대화할 수 있다. 그러나 취약점이 악용될 수 있는 경로는 무한하여 해당 경로를 모두 포괄할 수 있는 시그니처를 기술하고 이를 상용 악성코드 탐지규칙으로 활용하는 것은 현재의 컴퓨터과학에서 NP-Complete로 불리는 난제이다[21].

이처럼 기존 연구들은 특정 악성 트래픽에서만 시그니처 추출이 가능하거나, 또는 많은 악성코드 샘플들이 있어야만 시그니처 추출이 가능하다는 한계를 가지고 있다.

3. 백그라운드

3.1 토픽 모델링 기법

토픽 모델링(topic modeling)은 비자율지도(unsupervised) 기계학습 기법의 하나로 구조화되지 않은 문서의 집합(corpus)에서 맥락과 관련된 단서들을 이용하여 유사한 의미가 있는 단어들을 클러스터링하는 방식으로 문서(document)를 분류하고, 주제(topic)를 찾아내기 위한 텍스트마이닝 알고리즘이다[11].

토픽 모델링 알고리즘은 소셜 네트워크의 키워드를 추출하여 변화하는 이슈를 추적[12], 논문지에 게재된 논문을 분석하여 시기별 주목받는 주제를 파악[13], 신문기사의 주제를 추출하여 시기별 이슈 변화의 분석[14] 등에 활용되었다. 이처럼 토픽 모델링 기법은 특정 분야에 대한 동향 분석에 사용되어 그 성능이 입증된 알고리즘이다.

토픽모델링 알고리즘은 벡터 공간 모델(VSM: Vector Space Model)[15], 잠재 의미 분석(LSA: Latent Semantic Analysis)[16], 확률 잠재 의미 분석(pLSA: Probabilistic Latent Semantic Analysis)[17], 잠재 디리클레 할당(LDA : Latent Dirichlet Allocation)[18]이 있다. 본 논문에서는 [21]에서 잠재 디리클레 할당 알고리즘을 기반으로 개발된 LARGen 기법을 활용하여 탐지규칙 생성에 필요한 시그니처 추출을 수행하고 이를 이용해 스노트 탐지규칙을 자동생성하고 생성한 스노트 탐지규칙을 검증하는 시스템을 제안한다.

3.2 잠재 디리클레 할당 기법

잠재 디리클레 할당(LDA) 알고리즘은 텍스트 문서(document)에 내재한 주제(topic)를 추론하기 위해 고안된 토픽 모델링 알고리즘이다. 최근에는 매우 큰 자료(Big Data)에서 토픽을 발견하기 위한 통계적 토픽 모델로 많이 사용되고 있다[19]. 이절에서는 LDA를 간략히 요약한다.

3.2.1 가정

잠재 디리클레 할당 알고리즘은 두 개의 가정을 전제하고 있다.

(가정 1) 모든 문서(θ_i)는 한 개 이상의 주제를 내포하고 있으며, 어떤 문서가 특정 주제를 내포할 확률은 디리클레 분포를 따른다.

(가정 2) 모든 주제(ϕ_k)는 한 개 이상의 관련된 단어를 포함하고 있으며, 특정 주제가 특정단어를 내포할 확률은 디리클레 분포를 따른다.

디리클레 분포는 연속 확률분포의 하나로 K 차원의 실수 벡터 중 벡터의 요소가 양수이며 모든 요소를 더한 값이 1인 경우에 대한 확률 값이 정의되는 분포이다.

3.2.2 확률분포 추론

잠재 디리클레 할당 알고리즘의 목적은 수집된 문서들에서 자동으로 숨어있는 주제를 추론하는 것이다. 이것은 관측 불가능한 사후 확률 분포(ϕ_k, θ_i)를 계산하는 것을 의미하며 수식으로 표현하면 다음과 같다.

$$p(\theta, \phi | D, \alpha, \beta) = \frac{p(\theta, \phi, D | \alpha, \beta)}{p(D | \alpha, \beta)}$$

여기서 D는 분석대상이 되는 문서 corpus를 의미하며, 이 수식은 베이즈의 확률 분포 이론에 기초하고 있다. 잠재 디리클레 할당 확률 생성 수식의 해를 구하는 것은 NP-완비 문제이므로, 해를 구함에 있어 기댓값 최대화(EM : Expectation-Maximization) 알고리즘, 기브스 표집(Gibbs sampling) 알고리즘 등을 통한 근사접근 방식을 적용하고 있다.

3.2.3 잠재 디리클레 할당을 통한 시그니처 추출

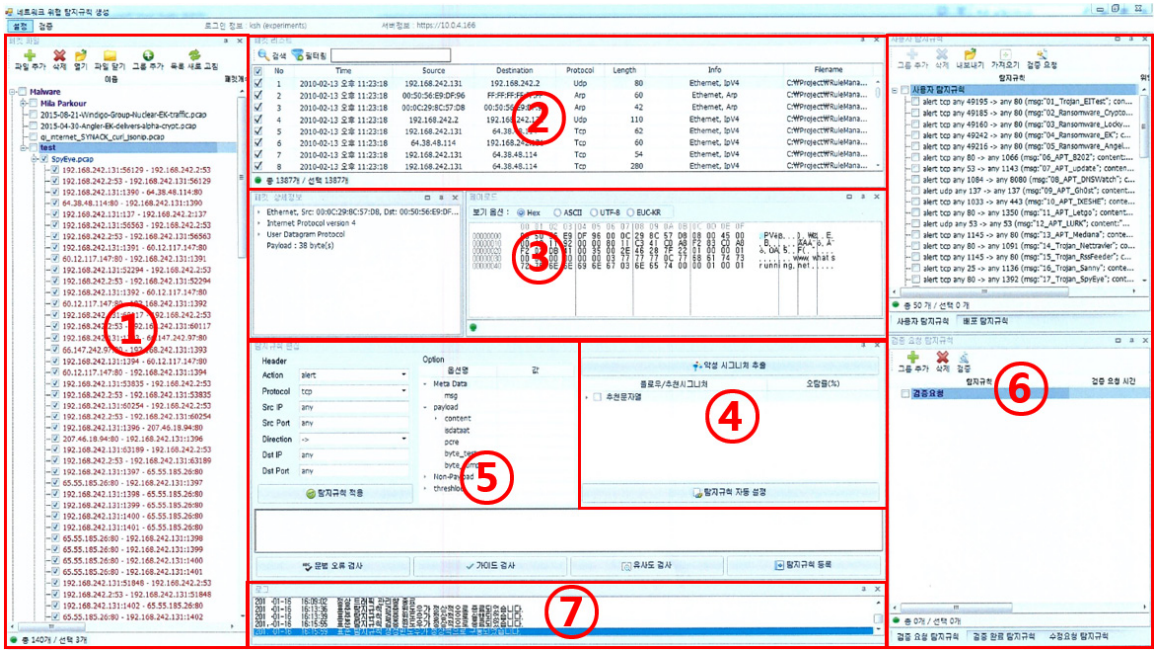
잠재 디리클레 할당 알고리즘은 문서 분류와 같은 작업에서 탁월한 결과를 보여주었다. 네트워크 트래픽과 일반 문서는 여러 가지 측면에서 유사하다. 표 2와 같이 네트워크 트래픽에서 플로우 집합은 문서의 집합과 대응시킬 수 있으며, 각각 플로우에 있는 페이로드를 문서와 대응시킬 수 있다. 그리고 페이로드의 이진 코드는 각 문서의 단어로 대응시킬 수 있다.

(표 2) 일반 문서와 네트워크 트래픽
(Table 2) Natural language documents and network traffic

일반 문서	네트워크 트래픽
문서 집합(corpus)	플로우 집합(flow set)
문서(document)	페이로드(payload)
단어(word)	이진 코드(binary)

네트워크 플로우는 트래픽을 생성한 응용 프로그램 정보 또는 플로우 정보를 전달하는 기본적인 정보들의 혼합체이다. 그리고 현재 네트워크를 통한 공격은 많은 경우에 HTTP, E-Mail 등의 인터넷 응용 플로우에 삽입된 형태로 전송되고 있다[20]. 예를 들어 블라스터 웜은 HTTP 플로우에 삽입된 형태로 네트워크에 전파된다. 이렇게 블러스터 웜을 포함한 HTTP 플로우를 잠재 디리클레 할당 알고리즘에서 사용하고 있는 용어로 정의하면 다음과 같다. “HTTP 주제와 블라스터 웜 주제를 모두 포함하고 있다.” 그리고 플로우에는 각 주제를 구성하는 이진 코드가 존재한다. 예를 들면 웹 서비스를 제공하기 위한 플로우에는 “HTTP”, “./GET”, “WWW“ 등과 같은 이진 코드가 포함되어 있다.

이렇게 일반 문서와 네트워크 트래픽을 대응시키면 잠재 디리클레 할당 알고리즘을 이용하여 정상 일반 응용 프로그램에서 생성한 트래픽과 악성코드에서 생성한 트



(그림 1) 탐지규칙 생성 시스템 GUI
(Figure 1) Rule Generation System GUI

래픽을 입력하면, 문서를 분류하는 것처럼 트래픽을 주제별로 분류할 수 있다. 그리고 악성 트래픽이 포함된 주제 그룹에 포함된 단어(이진 코드)를 분석하면 해당 악성 트래픽 시그니처의 추출이 가능하다. 제안 시스템은 시그니처 기반으로 동작하는 침입탐지시스템에 기반을 두고 있어, 악성코드의 다형성, 난독화 등에 원천적으로 취약하다. 최근 10여 년간 시그니처 기반 탐지연구에서의 이 문제에 대한 연구동향을 [21]에서 정리하고 있다.

4. 스노트 탐지규칙 자동 생성/검증 시스템

본 논문에서는 스노트 탐지규칙 자동생성 시스템(RGS: Rule Generation System)과 검증 시스템(RVS: Rule Verification System)을 제안한다. 악성 트래픽에서 잠재 디리클레 할당 알고리즘을 이용하여 악성 시그니처를 추출하고[21], 추출한 악성 시그니처 및 트래픽 분석 정보를 이용하여 스노트 탐지규칙을 준 전문가도 신속하게 생성하고 검증할 수 있는 시스템이다.

4.1 탐지규칙 생성 시스템(RGS)

탐지규칙 생성 시스템은 탐지규칙을 생성해야 하는 악성 트래픽을 쉽게 분석할 수 있는 환경을 제공하고, 자동으로 후보 시그니처를 추천해주며, 스노트 문법에 맞게 탐지규칙을 생성해 주는 기능을 하고 있다.

4.1.1 주요 모듈

탐지규칙 생성 시스템의 주요 모듈로는 1)트래픽 분석, 2)시그니처 추출, 3)탐지규칙 자동생성, 4)문법 오류검사, 5)가이드 검사, 6)유사도 검사 모듈이 있다.

1) 트래픽 분석 모듈은 .pcap 파일을 플로우별로 조립하고, 플로우별 패킷의 개수, 연결시간, 데이터 처리를 위한 데이터전송 대역폭, 포함된 세그먼트 수, 긴급데이터 수, 평균 지연시간, 비정상 패킷 포함 여부를 분석한다. 그리고 패킷별로 L2/L3/L4 및 페이로드 정보를 분석한다.

2) 시그니처 추출 모듈은 사용자가 선택한 파일 또는 플로우에서 [21]을 활용하여 시그니처를 추출한다.

3) 탐지규칙 자동생성 모듈은 사용자가 분석된 패킷의 페이로드 문자열을 선택하거나 [21]을 이용하여 추출한 시

그니처를 선택하면 자동으로 스노트 문법에 맞게 탐지규칙을 생성한다. 선택한 시그니처가 한 개일 경우에는 'content', 'offset', 'depth' 옵션을 자동으로 계산하여 생성하며, 다수개의 시그니처를 선택할 경우 정규표현식으로 표현이 가능할 경우 'pcre' 옵션을 이용하며, 정규표현식으로 표현할 수 없으면 여러개의 'content' 옵션을 사용한다.

4) 문법 오류검사 모듈은 작성한 스노트 탐지규칙의 문법적 오류 여부를 검사하며, 문법적으로 오류가 존재할 경우 하이라이팅 하여 사용자에게 알려준다.

5) 가이드 검사 모듈은 불필요한 연산이나 과부하가 발생 가능한 연산의 수행 가능성이 있는지 분석한다. 예를 들면, 불필요한 연산 가능성은 스노트 탐지규칙 옵션에서 payload detection 탐지규칙 옵션과 non-payload detection 탐지규칙 옵션의 적용 순서를 분석한다. 네트워크 트래픽에서 페이로드 부분의 검사는 많은 데이터를 확인해야 하여서 연산량도 많고, 시간도 많이 필요하다. 반면, 페이로드 이외의 부분 검사는 연산량도 적고, 시간도 오래 걸리지 않는다. 만약 페이로드에서 "ABCDEF" 문자가 포함된 패킷에서 TTL 값이 10 이하인 패킷을 탐지하는 것보단 TTL 값이 10 이하인 패킷에서 페이로드에 "ABCDEF" 문자가 포함된 패킷을 찾는 게 더 연산량도 적고, 시간도 적게 소모된다. 과부하가 발생 가능한 연산은 탐지규칙 옵션에서 페이로드 문자열을 비교하는 'content' 옵션을 너무 많이 사용하거나, 비교하는 문자열의 길이가 너무 짧으면 발생할 수 있다. 정규표현식을 이용하여 탐지하는 'pcre' 옵션에서도 광범위한 문자열 비교를 사용할 경우 과부하가 발생할 수 있다. 이처럼 불필요한 연산 또는 과부하가 발생 가능한 탐지규칙일 경우 사용자에게 알려준다.

6) 유사도 검사 모듈은 생성한 탐지규칙을 기존에 등록된 탐지규칙들과 유사도 분석을 수행한다.

4.1.2 RGS GUI

탐지규칙 생성 시스템의 GUI는 그림 1과 같이 7개의 창으로 구성되어 있다. 모든 창의 크기와 위치는 사용자 마음대로 변경할 수 있다.

첫 번째 악성 패킷 관리 창은 .pcap 포맷으로 저장된 악성 패킷을 관리하고 분석한 정보를 보여준다.

두 번째 패킷 목록 창은 악성 패킷 관리 창에서 선택한 파일 또는 플로우에 포함된 패킷들의 목록을 보여준다. 패킷 발생 시간, 출발지/목적의 IP/Port 정보, 프로토콜, 페이로드 길이 정보를 제공한다.

세 번째 패킷 상세 정보창은 패킷 목록 창에서 선택한 패킷의 L2/L3/L4의 헤더 정보를 보여주며, 페이로드 정보를 Hex, ASCII, UTF-8, EUC-KR 형태로 보여준다.

네 번째 추천 탐지 문자열 창에서는 시그니처를 추출하고 싶은 플로우를 악성 패킷 파일 관리창에서 선택을 한 후 '악성 시그니처 추출' 버튼을 클릭하면 프로그램에서 사전에 데이터베이스에 입력해 놓은 정상 트래픽과 [21]을 이용하여 추출한 악성 시그니처 목록을 보여준다.

다섯 번째 탐지규칙 편집 창은 추천 탐지 문자열 창에서 추출된 시그니처를 선택하면 자동으로 해당 플로우 정보를 분석하여 탐지규칙 형식에 맞게 생성한 스노트 탐지규칙의 결괏값을 보여준다. 사용자가 임의로 수정할 수 있다.

여섯 번째 탐지규칙 관리 창은 탐지규칙 편집 창에서 탐지규칙 편집을 완료한 후 저장한 탐지규칙의 목록과 실제 정보보안 시스템에 적용된 탐지규칙 목록을 가지고 있다. 여기서 생성한 초기의 탐지규칙들은 검증을 수행한 후에 보안 장비에 적용할 수 있다.

마지막 일곱 번째 창은 생성 시스템을 시작한 후 실행된 작업 내용을 보여준다.

4.2 탐지규칙 검증 시스템 (RVS)

탐지규칙 검증 시스템은 가상 IDS 서버와 가상 네트워크 환경을 이용하여 간단하고 신속하게 생성한 탐지규칙을 검증하는 기능을 제공한다.

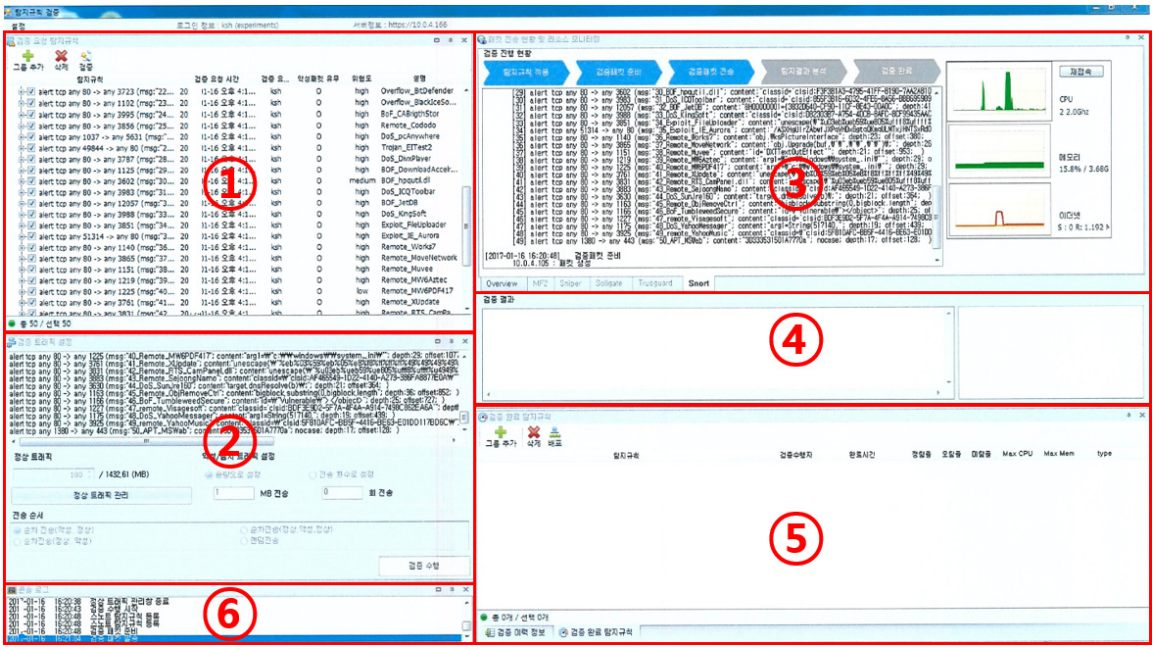
4.2.1 주요 모듈

탐지규칙 검증 시스템의 주요 모듈로는 1)탐지규칙 관리, 2)검증 트래픽 생성, 3)가상 IDS 서버 관리, 4)탐지 결과 분석 모듈이 있다.

1) 탐지규칙 관리 모듈은 생성 시스템에서 자동으로 생성된 탐지규칙 또는 사용자가 직접 생성한 탐지규칙을 탐지규칙 생성시 사용한 악성 트래픽과 대응시켜서 관리한다. 만약에 탐지규칙과 대응되는 악성 트래픽이 존재하지 않을 때는 임의로 탐지가 되는 트래픽을 생성한다.

2) 검증 트래픽 생성 모듈은 검증시 사용할 정상 트래픽과 탐지규칙 생성시 사용한 악성 트래픽을 사용자의 설정에 맞게 트래픽 용량 및 전송순서대로 트래픽을 생성하여 가상의 IDS 서버로 전송한다.

3) 가상 IDS 서버 관리 모듈은 탐지규칙 검증시 사용할 가상 IDS 서버를 추가/삭제/변경하는 기능을 하고 있다. 스노트 서버를 기본적으로 탑재하고 있으며 윈스 IDS, 안랩 IDS, 시큐아이 IDS, 인프니스 네트웍스 IDS와 연동을 할 수 있도록 개발되어 있다.



(그림 2) 탐지규칙 검증 시스템 GUI
(Figure 2) Rule Verification System GUI

4) 탐지 결과 분석 모듈은 생성한 검증용 트래픽을 가상 네트워크 환경을 통하여 가상 IDS 서버로 전송이 완료되면, 가상 IDS 서버들의 탐지 결과를 분석하여 탐지규칙별 정탐율, 오탐율, 미탐율을 계산하여 그래프로 보여준다.

4.2.2 RVS GUI

탐지규칙 검증시스템은 그림 2와 같이 6개의 부분으로 구성되어 있다.

첫 번째 검증 탐지규칙 목록 창은 탐지규칙 생성 시스템에서 만들거나 사용자가 직접 생성한 탐지규칙의 검증이 필요한 탐지규칙을 관리하는 창으로 탐지규칙 생성자 정보를 가지고 있으며, 여기에서 검증할 탐지규칙을 선택한다.

두 번째 탐지규칙 검증 환경 설정 창은 탐지규칙 검증시 사용할 가상 IDS 서버, 정상 트래픽, 정상/악성 트래픽 전송 순서를 설정할 수 있다.

세 번째 실시간 검증 진행 상황 창은 검증을 수행하고 있는 각 가상 IDS 서버별로 탐지규칙 적용, 검증패킷 준비, 검증패킷 전송, 탐지결과 분석, 검증 완료의 다섯 단계로 진행된다. 검증 수행 버튼을 클릭하면 선택한 가상 IDS 서버에 검증을 탐지규칙을 등록한 후, 검증 설정값에 맞게 정상과 악성 트래픽을 생성한 후 가상 네트워크 환경을 통하여 전송한다. 검증을 수행하는 동안 실시간으로 가상 IDS 서버의 CPU, 메모리, 네트워크 사용 현황을 보여준다.

네 번째 검증 결과 창은 각 가상 IDS 서버로 검증 트래픽 전송이 완료되면 탐지 결과를 분석하여 각 IDS 시스템별 정탐, 오탐, 미탐 결과를 보여준다.

다섯 번째 검증 완료 탐지규칙 목록 창은 기존에 검증을 수행했던 탐지규칙들의 목록 관리 창으로 탐지규칙을 선택하면 기존의 검증 결과를 확인할 수 있다.

마지막 여섯 번째 창은 검증 시스템을 시작한 후 실행된 작업 내용을 보여준다.

5. 실험 내용 및 결과

4장에서 설계한 잠재 디리클레 할당 알고리즘을 이용한 스노트 탐지규칙 생성/검증 시스템을 개발하고, 실험을 진행하였다.

5.1 실험 환경

5.1.1 시스템 구축

실험에는 RX2540 M1 서버를 사용하였고, 서버의 세부

사양은 다음과 같다.

- CPU : Intel Xeon E5-2600 18 core
- Memory : DDR4 160GB
- HDD : 8TB

소프트웨어는 가상화 솔루션 VMware ESXi 4.0을 이용하여 탐지규칙 생성/검증 시스템, 패킷 발송 서버, 스노트 서버, 윈스 IDS, 안랩 IDS, 시큐아이 IDS, 인프니스 네트웍스 IDS를 구성하고 가상 네트워크를 사용하는 환경으로 구축하였다.

5.1.2 데이터셋

기계학습인 잠재 디리클레 할당 알고리즘을 이용하기 위하여 실험 데이터를 수집하였다. 정상 트래픽은 미국 국립 CyberWatch 중부 대서양 대학 사이버 방어 대회(MACCDC : National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition)[22]에서 공개하고 있는 익명화된 트래픽 30GB를 이용하였다. 이 트래픽은 네트워크 관리와 방어 훈련을 위해 공개된 데이터이다. 악성 트래픽은 화이트 해커인 Mila Parkour가 공개한 82개의 악성 트래픽[23]을 이용하였다. 악성 트래픽의 구성은 표 3과 같이 서비스 거부 공격, 트로이 목마, SQL 삽입 공격, 익스플로잇 공격 등이다.

(표 3) 실험에 사용한 공격 트래픽
(Table 3) Attack traffic used in experiment

공격 타입	개수	주요 악성 트래픽 명칭
Trojan	20	RssFeeder, Gh0st, LURK, Medianna, Nettravler...
Overflow	20	Heap_Overflow_PoC, ActiveX_Remote_BoF...
DoS	21	ActiveX_Remote_Dos, WkimgSrc...
Injection	3	URI_Handlerargument_inject, SQLInjection, ...
Insecure Method	7	EDraw_Office_Viewer_Component -5.1, ...
Arbitrary code execution	4	VMware_IntraProcessLogging.dll_5 .5.3.42958, ...
other	7	metasploit_ie_aurora_exploitWin7, ...

그리고 생성된 탐지규칙을 검증하기 위한 정상 데이터로 인터넷 데이터 분석을 위한 협동조합 연합 CAIDA[24]에서 공개하고 있는 2012년부터 2017년까지의 정상 트래픽 2TB를 이용하였다.

5.1.3 시그니처 자동 선택 설정

시그니처 추출을 위한 잠재 디리클레 할당 알고리즘은 디리클레 확률분포를 계산할 때 사용하는 매개변수 α, β의 설정이 필요하다. [22]에서 잠재 디리클레 할당 알고리즘 최적의 매개변수 관련 연구를 수행한 결과, 문서별 주제에 대한 디리클레 확률분포 매개변수 $\alpha = 50/K$, 주제별 단어에 대한 디리클레 확률분포 매개변수 $\beta = 0.01$ 값을 도출하였다. 본 논문에서는 이 최적의 매개변수를 이용하여 실험을 진행하였다.

토픽의 개수(K)는 모든 공격들과 응용 프로그램별로 분류가 될 수 있도록 10, 30, 50, 100으로 K의 값을 변경하면서 실험을 진행하였다.

잠재 디리클레 할당 알고리즘을 통하여 탐지규칙을 생성하려고 하는 악성 트래픽만 가지고 있으면 악성 트래픽에서 추출된 시그니처 중에서 오탐률이 낮은 상위 10%의 시그니처 중에서 문자열이 가장 긴 시그니처 또는 한 개의 패킷에서 두 개의 시그니처가 추출된 시그니처를 이용하였다. 탐지규칙 헤더는 선택된 시그니처의 패킷 분석을 통하여 자동으로 생성이 되게 하였다.

5.2 RGS/RVS 활용 사례

개발한 시스템을 이용하여 탐지규칙을 생성하고 검증을 하였다. 예제로 사용된 악성코드는 2009년에 발견된 이후 현재까지도 활동 중이며 수많은 변형된 악성코드가 발견되고 있는 스파이아이(SpyEye)이다[25]. 스파이아이는 윈도우, iOS의 환경에서 다양한 웹 브라우저(사파리, 크롬, 파이어폭스, 익스플로우)에서 동작하는 악성코드로 키 로깅, 트로이 목마 등 다양한 공격 기능이 있다.

악성 트래픽은 [23]의 "BIN_SpyEye_2010-02.pcap" 파일을 이용하였다. 탐지규칙 생성 시스템에서 트래픽을 분석한 결과 총 397개의 패킷으로 구성이 되어있으며 40개 이상의 플로우로 구성이 되어있지만, 통신한 IP는 총 7개(192.168.242.131, 192.168.242.2, 64.38.48.114, 60.12.117.147, 66.147.242.97, 207.46.18.94, 65.55.275.26)이며, 총 접속 시간은 86.345초이다. 패킷에 포함된 모든 플로우를 선택한 후 [22]의 정상 트래픽과 잠재 디리클레 할당 알고리즘을 이용하여 최소 단어의 길이를 16으로 설정하고 악성 시그니처를 추출하였다.

(그림 3) 시그니처 추출 결과
(Figure 3) Signature extraction result

추출된 결과는 그림 3과 같이 플로우당 여러 개의 시그니처가 추출되었다. 논문에서는 이 중에서 실행 파일명이 포함된 “LOAD
http://www.missboston.org/wp-includes/images/wlw/win.exe
1137” 시그니처를 선택하여 탐지규칙을 생성하였다. 생성된 탐지규칙은 다음과 같다.

*alert tcp any 80 -> any 1392 (msg:"Other_SpyEye" content:"LOAD
http://www.missboston.org/wp-includes/images/wlw/win.exe
1137"; offset:233; depth:72; nocase;)*

자동으로 생성된 탐지규칙을 탐지규칙 검증 시스템에서 스노트 서버, 가상 안랩 IDS 서버, 가상 윈스 IDS 서버, 가상 시큐아이 IDS 서버, 가상 인프니스 네트웍스 IDS 서버와 [24] 정상 트래픽 중 3GB를 이용하여서 탐지규칙을 검증하였다. 검증 결과 모든 가상 IDS 서버에서 정탐 100%, 오탐 0%, 미탐 0%가 나오는 것을 확인하였다.

이렇게 스파이아이 악성 트래픽을 이용하여서 탐지규칙을 생성하고 검증하는데 소요된 시간은 30분도 걸리지 않았다.

5.3 실험 결과

표 4는 탐지규칙 생성 시스템을 이용하여 자동으로 생성된 탐지규칙 결과의 일부이다. 이 탐지규칙을 탐지규칙 검증시스템을 이용하여 [24] 트래픽 중 샘플링 한 정상 트래픽 3GB와 악성 트래픽을 이용하여 검증한 결과 모두 정상적으로 악성 트래픽만 탐지하였으며, 부하도 발생하지 않았다.

6. 결 론

본 논문에서는 급증하는 악성코드에 대응하는 데 필요

(표 4) 자동 생성된 탐지규칙 샘플
(Table 4) Sample of automatically generated detection rule

유형	악성코드	탐지규칙
Trojan	RssFeeder	alert tcp any 1146 -> any 80 (msg:"Trojan_RssFeeder" content:"Professional3&macaddr=00:0C:29:71:24:89&owner=two13&version=1.2.0&t=4841"; offset:152; depth:71; nocase;)
Overflow	muvee_autoProducer_6.1(TextOut.dll)	alert tcp any 1151 -> any 80 (msg:"Overflow_muvee" content:"/EF/6220-5395/muvee%20autoProducer%20%3C=%206.1%20(TextOut.dll)%20ActiveX%20Remote%20BOF%20Exploit.html"; offset:58 ; depth:102;)
DoS	MSWorks7_WkImgSrv.dll	alert tcp any 1140 -> any 80 (msg:"DoS_WkImgSrv.dll" content:"|49 53 4F 2D 38 38 35 39 2D 31 2C 75 74 66 2D 38 3B 71 3D 30 2E 37 2C 2A 3B 71 3D 30 2E 37|"; offset:399; depth:30;)
Injection	IBM_Expeditor_cai_URI_Handlerargument	alert tcp any 80 -> any 1152 (msg:"Injection_Expeditor" content:"%22=\\10.1.1.100\vrt-doc\VRS_Reports\CAN-2004-0480%20(IBM%20Lotus%20Notes%20URI%20Handler%20Argument%20Injection)\Research\notes.ini%22">Click"; offset:530; depth:142; nocase;)
Insecure Method	Edraw_Office_View_Component-5.1_HttpDownloadFile()	alert tcp any 80 -> any 1596 (msg:"InsecureMothod_Edraw5.1" content:"|3E 4C 40 30 65 2D 28 2F|"; offset:672; depth:8; content:"|77 6A 4B 3A 52 7C|"; offset:992; depth:6;)
Arbitrary code execution	Vmware_IntraProcessLogging.dll_5.5.3.429	alert tcp any 80 -> any 1691 (msg:"ArbitaryCodeExecuriton_Vmware" content:"|7D 28 42 5A 23 66 67|"; offset:363; depth:7; content:"|26 5A 40 62 57|"; offset:469; depth:5;)
Others	SpyEye	alert tcp any 80 -> any 1392 (msg:"Other_SpyEye" content:"LOAD http://www.missboston.org/wp-includes/images/wlw/win.exe 1137"; offset:233; depth:72; nocase;)

한 시그니처 기반의 탐지규칙 생성을 전문가뿐만 아니라 준전문가도 신속하고, 정확하게 탐지규칙을 생성하고 검증할 수 있는 시스템을 제안하고, 실제 개발을 통하여 실험까지 진행하였다.

제안하고 개발한 시스템은 탐지규칙을 생성하고자 하는 악성 트래픽에 페이로드만 존재한다면 악성코드의 종류와는 상관이 없이 탐지규칙 생성이 가능하다.

시스템 실험 결과 정상적으로 탐지규칙을 생성하고 검증할 수 있었으며, 이 시스템을 이용할 경우 시스템 사용자와 탐지규칙에 따라 차이는 존재하지만, 1시간 이내에 탐지규칙 생성 및 검증을 할 수 있었다.

개발한 탐지규칙 생성/검증 시스템을 이용하면 전문가들은 기존보다 훨씬 빠른 속도로 탐지규칙을 생성하고 검증할 수 있을 것이며, 아직 숙련되지 않은 준전문가들도 기존보다 더 쉽게 탐지규칙을 생성하고 검증할 수 있을 것으로 예상한다.

참고문헌(Reference)

[1] Kaspersky Lab, “Kaspersky Lab Number of the Year: 360,000 Malicious Files Detected Daily in 2017”, 2017. https://www.kaspersky.com/about/press-releases/2017_kaspersky-lab-detects-360000-new-malicious-files-daily

[2] McAfee Lab, “McAfee Labs Threats Report”, 2018. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2018.pdf

[3] SNORT, http://www.snort.org/

[4] TTA, “Snort 기반 침입탐지시스템 참지 규칙 요구사항”, TTAK.KO-12.0283, 2015.

[5] Y. Tang S. Chen, ″Defending Against Internet Worms: A Signature-Based Approach″, in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol.2, pp. 1384-1394, 2005. http://doi.org/10.1109/INFCOM.2005.1498363

[6] Shabtai, A. Menahem, E. and Elovici, Y. ″F-Sign: Automatic, Function-Based Signature Generation for Malware″, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, Vol.41, no.4, pp. 494-508, 2011. http://doi.org/10.1109/TSMCC.2010.2068544

[7] G. Tahan, C. Glezer, Y. Elovici and L. Rokach, ″Auto-Sign: an automatic signature generator for high-speed malware filtering devices″, Journal in Computer Virology, Vol.6, no.2. pp. 91-103, 2010. https://doi.org/10.1007/s11416-009-0119-3

[8] K. Griffin, S. Schneider, X. Hu and T. Chiueh, ″Automatic Generation of String Signatures for Malware Detection″, 12th International Symposium, RAID 2009, pp. 101-120, 2009. https://doi.org/10.1007/978-3-642-04342-0_6

[9] Mohammed, M.M.Z.E., Chan, H.A. and Ventura, N., ″Honeycyber: Automated signature generation for zero-day polymorphic worms″, in Military Communications Conference, MILCOM 2008, pp. 1-6, Novemver, 2008. http://doi.org/10.1109/MILCOM.2008.4753178

[10] H. J. Wang, C. Guo, D. R. Simon, and A. Zugenmaier, “Shield: Vulnerability-driven network filters for preventing known vulnerability exploits.” ACM SIGCOMM, 2004. http://doi.org/10.1145/1015467.1015489

[11] D. M. Blei, ″Probabilistic Topic Models″, Communications of th ACM, Vol.55, pp. 77-84, 2012. https://doi.org/10.1145/2133806.2133826

[12] T. N. Rubin, A. Chambers, P. Smyth, M. Steyvers, ″Statistical topic models for multi-label document classification″, in Machine Learning, Vol.88, pp. 157-208, 2003. https://doi.org/10.100k7/s10994-011-5272-5

[13] S. M. Gerrish, and D. M. Blei, ″A language-based approach to measuring scholarly impact″, ICML’10 Proceedings, pp. 375-382, 2010. https://dl.acm.org/citation.cfm?id=3104371

[14] D. J. Newman, and S. Block, ″Probabilistic topic decomposition of an eighteenth-century American newspaper″, in the journal of American Society for Information Science and Technology, Vol.57, pp. 753-767, 2006. https://doi.org/10.1002/asi.v57:6

[15] G.Salton, A.Wong, and C. S. Yang, ″A Vector space model for automatic indexing″, Communications of the ACM, Vol.18(11), pp. 613-620, 1975. https://doi.org/10.1145/361219.361220

[16] S.Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, ″Indexing by latent semantic

analysis", Journal of the American Society for Information Science banner, 1990. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[17] T. Hofmann, "Probabilistic latent semantic analysis", UAI'pp Proceedings of the Fifteenth conference on Uncertainty in srtificial intelligence, pp. 289-296, 1999. https://dl.acm.org/citation.cfm?id=2073829

[18] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", in the journal of Machine Learning Research, Vol.3, pp. 993-1022, 2003. https://dl.acm.org/citation.cfm?id=944937

[19] T. N. Rubin, A.Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification", Machine Learning, Vol.88(1-2), pp. 157-208, 2012. https://doi.org/10.1007/s10994-011-5272-5

[20] C. C. Zou, D. Towsley and W. Gong, "Modeling and simulation study of the propagation and defense of internet e-mail worms", Dependable and Secure Computing, IEEE Transactions on Vol.4(2), pp. 105- 118, 2007. https://doi.org/10.1109/TDSC.2007.1001

[21] S. Lee, S. Kim, S. Lee, J, Choi, H. Yoon, D. Lee, and J. Lee "LARGen: Automatic Signature Generation for Malwares Using Latent Dirichlet Allocation", IEEE TDSC Vol.15(5), pp. 771- 783, 2018. https://doi.org/10.1109/TDSC.2016.2609907

[22] NETRESEC, "Capture files from Mid-Atlantic CCDC" http://www.netresec.com/?page=MACCDC

[23] M. Parkour, "contagio malware dump" http://contagiodump.blogspot.com

[24] CAIDA, "Data Collection, Curation and Sharing" http://www.caida.org/data/

[25] Wikipedia, "SpyEye" https://en.wikipedia.org/wiki/SpyEye

[26] Zhuo. Zhang, Zhibin Zhang, Patrick P.C.Lee, Yunjie Liu and Gaogang Xie "ProWord: An unsupervised approach to protocol feature word extraction", in INFOCOM, 2014 Proceedings IEEE, pp. 1393-1401, July, 2014. http://doi.org/10.1109/INFOCOM.2014.6848073

[27] Omid E. David and Nathan S. Netanahu, "DeepSign: Deep learning for automatic malware signature generation and classification", International Joint Conference on Neural Networks, July, 2015. http://doi.org/10.1109/IJCNN.2015.7280815

[28] Fabrizio Biondi, Francois Dechelle and Axel Legay "MASSE: Modular Automated Syntactic Signature Extraction", IEEE International Symposium on Software REliability Enginerring Workshops, Oct, 2017. http://doi.org/10.1109/ISSREW.2017.74

◐ 저 자 소 개 ◑

김 성 호(Sungho Kim)
2010년 전남대학교 대학원 정보보호협동과정(이학석사)
2011년~2014년 한국인터넷진흥원 선임연구원
2014년~현재 한국전자통신연구원 부설연구소 선임연구원
관심분야 : 네트워크 보안, 모바일 보안
E-mail : ksh98@nsr.re.kr

이 수 철(Suchul Lee)
2008년 서울대학교 전기·컴퓨터공학부(공학사)
2014년 서울대학교 대학원 컴퓨터공학부(공학박사)
2014년~2016년 한국전자통신연구원 부설연구소 연구원
2016년~현재 한국교통대학교 철도대학 철도경영·물류·컴퓨터학부(컴퓨터정보공학전공) 조교수
관심분야 : 정보통신 및 보안, 인공지능
E-mail : sclee@ut.ac.kr