pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2013.2.6.265

A Text Mining-based Intrusion Log Recommendation in Digital Forensics

Sujeong Ko†

ABSTRACT

In digital forensics log files have been stored as a form of large data for the purpose of tracing users' past behaviors. It is difficult for investigators to manually analysis the large log data without clues. In this paper, we propose a text mining technique for extracting intrusion logs from a large log set to recommend reliable evidences to investigators. In the training stage, the proposed method extracts intrusion association words from a training log set by using Apriori algorithm after preprocessing and the probability of intrusion for association words are computed by combining support and confidence. Robinson's method of computing confidences for filtering spam mails is applied to extracting intrusion logs in the proposed method. As the results, the association word knowledge base is constructed by including the weights of the probability of intrusion for association words to improve the accuracy. In the test stage, the probability of intrusion logs and the probability of normal logs in a test log set are computed by Fisher's inverse chi-square classification algorithm based on the association word knowledge base respectively and intrusion logs are extracted from combining the results. Then, the intrusion logs are recommended to investigators. The proposed method uses a training method of clearly analyzing the meaning of data from an unstructured large log data. As the results, it complements the problem of reduction in accuracy caused by data ambiguity. In addition, the proposed method recommends intrusion logs by using Fisher's inverse chi-square classification algorithm. So, it reduces the rate of false positive(FP) and decreases in laborious effort to extract evidences manually.

Keywords : Digital Forensics, Text Mining, Intrusion Log Recommendation, Association Word Knowledge Base, Fisher's Inverse Chi-square Classification Algorithim

디지털 포렌식에서 텍스트 마이닝 기반 침입 흔적 로그 추천

고 수 정†

요 약

디지털 포렌식에서의 로그 데이터는 사용자의 과거 행적에 대한 추적을 목적으로 대용량의 형태로 저장된다는 특성을 가지고 있다. 이러한 대용량의 로그 데이터를 단서가 없이 수동으로 분석하는 절차는 조사관들에게는 어려운 일이다. 본 논문에서는 포렌식 분석을 하는 조사관들에게 믿을 만한 증거를 추천하기 위하여 대용량의 로그 집합으로부터 해킹 흔적을 추출하는 텍스트 마이닝 기술을 제안한다. 학습 단계에서는 훈련 로그 집합을 대상으로 전처리를 한 후, Apriori 알고리즘을 이용하여 침입 흔적 연관 단어를 추출하고, 신뢰도와 지지도를 병합하여 각 연관 단어의 침입 흔적 확률을 계산한다. 또한, 침입 흔적 확률의 정확도를 높이기 위하여 스팸 메일의 여과에 사용된 Robinson의 신뢰도 계산 방법을 이용하여 확률에 가중치를 추가하며, 최종적으로 침입 흔적 연관 단어 지식 베이스를 구축한다. 테스트 단계에서는 연관 단어 지식 베이스를 기반으로 테스트 로그 집합에 대해 피셔(Fisher)의 역 카이제곱 분류 알고리즘을 적용하여 침입 흔적 로그일 확률과 정상 로그일 확률을 계산하고, 이를 병합하여 침입 흔적 로그를 추출한다. 추출된 로그를 조사관에게 침입 흔적이 있는 로그로서 추천한다. 제안한 방법은 비구조화된 대용량의 로그 데이터를 대상으로 데이터의 의미를 명확하게 분석할 수 있는 학습 방법을 사용함으로써 데이터의 모호성으로 인해 발생하는 정확도 저하 문제를 보완할 수 있으며, 피셔의 역 카이제곱 분류 알고리즘을 이용하여 추천함으로써 오분류율(false positive)을 감소시키고 수동으로 증거를 추출하는 번거로움을 줄일 수 있다는 장점을 갖는다.

키워드 : 디지털 포렌식, 텍스트 마이닝, 침입 흔적 로그 추천, 연관 단어 지식 베이스, 피셔(Fisher)의 역 카이제곱 분류 알고리즘

1. 서 론

컴퓨터의 보급이 일반화 되면서 활용되는 데이터의 많은 부분들이 디지털 자료로 작성되어 사용되고 있다. 이러한 디지털 환경은 디지털 장치를 활용한 사이버 범죄의 다양화로 발전함에 따라 사이버 범죄를 다루는 피해 시스템의 증

※ 이 논문은 인덕대학교 교내학술연구비에 의하여 연구되었음.
† 종신회원 : 인덕대학교 컴퓨터소프트웨어과 교수
논문접수 : 2013년 2월 4일
수 정 일 : 1차 2013년 4월 26일
심사완료 : 2013년 4월 26일
* Corresponding Author : Sujeong Ko(sjko@induk.ac.kr)

거수집, 복구 및 분석을 하는 컴퓨터 포렌식 기술이 다양한 분야에서 연구되고 있다[1,2]. 특히, 데이터 마이닝 기술을 이용하여 대량의 데이터로부터 유용한 유형을 발견하고 추출하여 포렌식 분석에 이용하려는 연구가 활발하게 진행되고 있다[3,4]. 포렌식 분석 과정에서는 삭제된 파일의 복원작업, 숨겨진 파일을 찾아내는 작업, 대용량의 정보 속에서 특정 정보를 추출하는 작업, 암호화된 파일에서 범죄와 관련된 정보를 추출하는 작업, 기타 장애가 발생한 디렉토리를 복구하여 증거를 추출하는 작업 등이 이루어진다. 포렌식 분석 대상의 하나인 텍스트 정보는 중요한 정보를 추출할 수 있는 핵심 자료이다. 유용한 텍스트 자료의 대부분은 보통 상당히 대량의 자료이므로 디지털 포렌식 분석가가 이를 대상으로 단서가 없이 수동으로 각 데이터의 의미를 분석하고 유용한 정보를 추출하는 절차는 상당한 시간과 노력을 필요로 하는 번거로운 작업이다.

이러한 문제점을 해결하고 더욱 효과적으로 분석 작업을 수행하기 위한 방법을 연구한 기존의 연구로는 수상한 사용자를 찾아내기 위하여 로그들을 결합하고 그 결과를 기반으로 의사 결정 트리를 이용하는 기술[5], 그리고 로그 데이터로부터 프로파일을 생성하고 그 규칙 집합을 사용하는 방법[6], 포렌식 분석을 위하여 텍스트 군집을 이용하는 방법 등이 있다[4]. 프로파일을 이용하는 방법과 텍스트 군집을 이용한 방법은 학습된 규칙 집합을 제공하여 초반에 단서가 없는 상태에서 증거를 추출하는 데 도움이 되나 증거 추출을 위한 별도의 시간이 필요하다는 단점을 갖는다. 의사 결정 트리를 이용한 방법은 사용자 간의 관계를 이용하여 수상한 사용자를 찾아낼 수 있다는 장점이 있으나 로그 파일 중 사용자의 IP 필드 외에 다른 필드에 대한 분석이 생략되어 증거를 찾는 데 있어서 정확도의 한계를 나타낸다. 따라서, 조사관들이 조사를 시작할 때 단서가 없는 경우 유용하게 사용될 수 있는 학습된 자료가 필요하며, 또한 증거 추출하는 데 시간과 노력을 절약할 수 있는 효율적인 텍스트 마이닝 기술이 필요하다.

본 논문에서는 포렌식 분석을 하는 조사관들에게 믿을 만한 증거를 제공하기 위하여 대용량의 로그 집합으로부터 침입 흔적 증거를 추출하고, 침입 흔적 로그를 추천하는 텍스트 마이닝 기술을 제안한다. 침입 흔적이 있는 훈련 로그 집합을 대상으로 전처리를 시행한 후, Apriori 알고리즘[7]을 이용하여 침입 흔적 연관 단어를 추출한다. Apriori 알고리즘의 신뢰도와 지지도를 병합하여 연관 단어의 침입 흔적 확률을 계산[8,9]하며, Robinson이 사용한 신뢰도 계산 방법[10]을 사용하여 가중치를 추가함으로써 침입 흔적 연관 단어 지식 베이스를 구축한다. 다음으로, 침입 흔적 연관 단어 지식 베이스를 기반으로 피셔(Fisher)의 역 카이제곱 분류 알고리즘[11]을 이용하여 테스트 로그 집합의 로그에 대한 침입 흔적 확률을 계산한다. 최종적으로, 계산된 확률을 기반으로 순위를 부여하고 조사관에게 침입 흔적이 있는 로그를 추천한다.

본 논문의 구성은 2장에서 포렌식 절차 기반 시스템의 구성도를 보이며, 3장에서는 텍스트 마이닝을 이용하여 침입 흔적 연관 단어를 추출하는 방법을 기술한다. 4장에서는 침입 흔적 연관 단어 지식 베이스를 구축하고, 이를 기반으로 테스트 로그 집합의 로그에 대해 침입 흔적 확률을 계산하며, 마지막으로 침입 흔적 확률 가중치가 높은 로그를 조사관에게 추천하는 방법을 기술한다. 5장에서 성능 평가를 하며, 6장에서는 결론을 기술한다.

2. 포렌식 절차 기반 시스템 구성도

포렌식 절차에 대해 정의한 여러 가지 연구가 있으나 본 논문에서는 경찰청에서 제정한 사이버 포렌식 표준 처리 절차[12]를 기준으로 포렌식 방법을 기술한다. 사이버 포렌식 표준 절차는 준비단계(Preparation), 수집단계(Collection), 검사단계(Examination), 요청접수/이송단계(Request Receipt/Transport), 분석단계(Analysis), 보고서작성단계(Reporting), 보전/관리단계(Presentation/Evidence Management), 법률적용/기소단계(Applying Law/Prosecution) 등의 단계로 정의한다. Table 1은 각 단계의 표준 포렌식 처리 절차에 대한 개요를 나타낸다.

Table 1. Standard forensic process

process	outline
Preparation	Preparation for collecting evidences
Collection	Collection of real digital evidences
Examination	Decision of data for analyzing
Request Receipt/Transport	Transport digital evidences for analyzing
Analysis	Analysis of digital evidences
Reporting	Reporting of job process that analyzes evidences
Presentation /Evidence Management	Presentation and management of extracted digital evidences
Applying Law/Prosecution	Decision of applying law and prosecution

Fig. 1은 Table 1의 포렌식 절차에 기반하여 작성한 디지털 포렌식에서 텍스트 마이닝 기반 침입 흔적 로그 추천 시스템의 구성도를 나타낸다. 준비/수집단계에서는 포렌식 자료를 준비하기 위하여 침입 흔적이 있는 시스템의 웹로그 파일을 수집한다. 검사단계에서는 수집한 웹로그 파일을 대상으로 데이터 정제 작업을 실시하여 포렌식 분석을 하기에 적합하지 않은 로그 파일을 제외시킨다. 분석단계에서는 비구조화된 데이터를 구조화된 데이터 형태로 변경하기 위하여 불용어처리, 숫자필터링, 어간 추출 등의 전처리 작업을 실시한 후, 훈련 로그 집합을 대상으로 Apriori 알고리즘을 이용하여 침입 흔적 연관 단어를 추출한다.

추출된 침입 흔적 연관 단어 집합을 대상으로 신뢰도와 지지도를 병합하여 침입 흔적 확률을 계산하고, Robinson이 사용한 신뢰도 계산 방법에 의하여 연관 단어에 가중치를

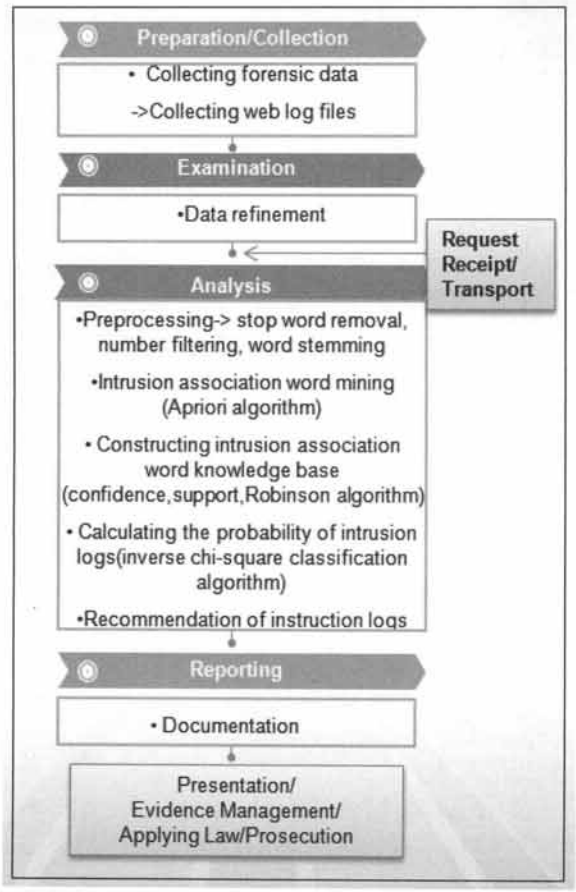

Fig. 1. System diagram of a text mining-based intrusion log recommendation in digital forensics

추가하여 침입 흔적 연관 단어 지식 베이스를 구축한다. 다음으로 테스트 로그 집합의 로그를 대상으로 전처리를 하고, 이를 대상으로 Apriori 알고리즘을 이용하여 연관 단어를 추출한다. 추출된 연관 단어를 기반으로 테스트 로그 집합의 로그가 침입 흔적이 있는 로그인가 정상 로그인가를 판별하기 위해, 피셔의 역 카이제곱 분류 알고리즘을 이용하여 각각의 로그에 대해 침입 흔적 로그일 확률과 정상 로그일 확률을 계산한다. 최종적으로, 이들을 병합하여 각 로그들의 침입 흔적 로그 확률을 계산하고 추천한다. 조사관에게 추천된 로그는 초기 단계의 조사에 유용하게 사용되어 침입 흔적을 효율적으로 분석하는 것을 도움으로써 시간을 절약하고 분류의 오류를 저하시킨다. 보고단계에서는 전체적인 과정에 대해 문서화하고 보고한다.

3. 텍스트 마이닝 기반 침입 흔적 연관 단어 추출

웹로그를 분석하기 위한 디지털 포렌식 테스트 마이닝 방법은 다양한 분야에서 연구되어왔다. 웹로그에서 침입을 탐지하기 위하여 전처리를 하는 방법[13], 빈발 유형을 찾기 위해서 전처리를 하는 기술[14], 그리고 텍스트 범주화의 사용에 의해 로그로부터 정상적인 사용자 행위와 악의를 가진 사용자 행위의 특징을 학습하는 방법[15] 등이 있다. 본 논문에서 제안한 방법은 데이터의 의미를 명확하게 분석함으로써 데이터의 모호성으로 인해 발생하는 정확도 저하 문제를 보완하는 디지털 포렌식 테스트 마이닝 방법을 제안한다. 본 장에서는 침입 흔적 로그를 추천하기 위한 전처리로, 단어간의 의미를 명확하게 할 수 있도록 훈련 로그 집합으로부터 연관 단어를 추출하는 학습 방법을 기술한다.

3.1 연관 단어 추출을 위한 전처리

웹로그 자료는 항상 적합하지 않은 정보, 불완전한 정보, 필요 없는 정보 등을 포함하기 때문에 Apriori 알고리즘에 적용되었을 때 그 정확도를 저하시킨다. 따라서, 자료에 대한 전처리 과정이 필요하다[14,16]. 전처리 과정의 첫 단계로, 최적의 자료를 추출하기 위한 데이터 정제 과정을 수행한다. 데이터 정제 과정의 다음 단계로, 각 로그를 대상으로 불용어 처리, 숫자 필터링, 그리고 어간추출 등의 과정을 차례로 실시한다. Table 2는 연관 단어 추출을 위한 전처리 과정을 나타낸다.

Table 2. Preprocessing for extracting association words

process	role
Data refinement	Cleaning images, failed requests, and unnecessary fields from a log file
Stop word removal	Special characters are classified into stop words because they do not contain important significance on log analysis
Number filtering	Filtering meaningless numbers excluding numbers in specified fields at a log file
Word stemming	Removing root

웹서버에 저장된 로그의 전처리를 하기 위하여 마이크로소프트 웹서버 IIS로부터 웹로그를 수집하였다. 웹서버는 웹사이트에서의 사용자 활동 이력을 텍스트 형태의 로그 파일로 기록하며, 이러한 로그 자료는 웹사이트의 디자인을 수정하거나 웹사이트의 성능을 전반적으로 향상시키는 데 사용되어왔다. NCSA, W3C, 그리고 IIS 등과 같은 웹로그의 많은 형태가 있으나 이들은 비슷한 형식을 갖는다. 웹서버에 저장된 로그는 date(날짜), time(시간), c-ip(클라이언트 IP주소), cs-username(사용자 이름), s-sitename(서비스 이름), s-computername(서버 이름), s-ip(서버 IP주소), s-port(서버 포트), cs-method(메서드), cs-uri-stem(URI 스템), cs-uri-query(URI 쿼리), sc-status(프로토콜 상태), sc-bytes(보낸 바이트 수), cs-bytes(받은 바이트 수), time-taken(걸린 시간), cs-host(호스트), cs-Useragent(사용자 에이전트), cs-Cookie(쿠키), cs-Referer(참조페이지) 등의 필드(field)로 구성된다. Fig. 2는 로그 파일의 예를 나타낸다.

수집한 로그로부터 침입 흔적을 분석하기 위하여 Table 2의 데이터 정제, 불용어 처리, 숫자 필터링, 어간추출 등의 전처리 과정을 끝낸 자료를 기초 자료로 활용한다. 데이터

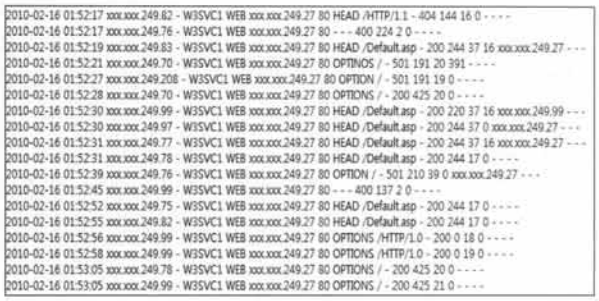

Fig. 2. Example of an IIS web server log file

정제 과정 중 로그 분석에 영향이 큰 필드만을 수집하기 위해 다음과 같이 정의된 필드 이외의 나머지 필드는 불용 필드로 간주한다. 전체 필드 중 date, c-ip, cs-method, cs-uri-stem, cs-uri-query, sc-status 등의 필드들을 추출한다. Fig. 3은 Fig. 2의 로그 파일에 대해 전처리를 완료한 결과를 나타낸다.

Fig. 3. Example of an IIS web server log file after preprocessiong

Fig. 3과 같이 전처리를 완료한 훈련 로그 집합의 로그들을 전문가들에 의해 해킹 흔적이 있는 로그와 정상 로그로 분류한다.

3.2 침입 흔적 연관 단어 마이닝

컴퓨터 포렌식 분야에서 효율적으로 사용되고 있는 데이터 마이닝 기술은 의사 결정 트리, 신경망, 그리고 SVM (support vector machine) 등이 있다[17,18]. 의사 결정 트리는 과거에 수집된 레코드들을 분석하여 이들 사이에 존재하는 패턴 특성을 나무의 형태로 만드는 것이고, 신경망은 인간 두뇌의 신경세포를 모방한 개념으로 마디(node)와 고리(link)로 구성된 망구조를 모형화하고, 의사 결정 트리와 마찬가지로 과거에 수집된 데이터로부터 반복적인 학습 과정을 거쳐 데이터에 내재되어 있는 패턴을 찾아내는 모델링 기법이다. SVM은 명료한 이론적 근거에 기반하며 간단한 알고리즘을 통하여 학습을 성공적으로 수행하는데 미치는 요소들을 규명한다. 그러나 의사 결정 트리는 단일의 추천에 의해 수행되므로 유형을 찾는 데 있어서 정확도가 낮을 수 있으며, 신경망과 SVM은 일반화하기 위하여 부가적인 지식 추출 기술이 필요함에 따라 발견된 유형을 기술하는 단순한 방법을 제공하지 못한다.

반면, 연관 규칙 마이닝[7]은 상용적인 데이터 집합을 분석하는 데 효율적인 기술로, 이 기술을 대표하는 알고리즘인 Apriori 알고리즘은 대용량의 데이터 집합으로부터 유용한 연관 규칙을 찾는 데 계산적으로 유용한 방법을 제공한다. Apriori 알고리즘은 최대 k개의 경로로 구성된다. 첫 번째 경로는 1-아이템 집합의 지지도를 계산하는 단계로 최소 지지도보다 작은 지지도를 갖는 규칙을 무시하고, 고빈도 단어 항목(L)을 구성한다. 두 번째 단계는 2-아이템 집합을 발생하는 단계로, 첫 번째 경로로부터 추출된 1-아이템 집합의 고빈도 단어 항목의 쌍으로부터 추출할 수 있다. 이와 같은 과정을 지속적으로 진행하여 남은 모든 아이템 집합이 없을 때까지 아이템 집합을 추출한다. 각 단계에서 고빈도 단어 항목(L)을 구성하는 방법은 각 단계의 아이템 집합을 대상으로 모든 공집합이 아닌 부분집합들을 찾는다. 부분집합을 찾기 위해서는 신뢰도(confidence)와 지지도(support)를 결정해야 한다. 신뢰도를 결정하기 위한 식은 식 (1)이다. 식 (1)은 단어 W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 단어 W1의 항목을 포함하고 있는 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Confidence(W1->W2)=Pr(W2|W1) \quad (1)$$

지지도를 결정하기 위한 식 (2)는 전체 단어들의 쌍 중에 각 연관 단어의 출현 빈도를 나타낸다. 식 (2)는 단어 W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 데이터베이스 내의 전체 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Support(W1->W2)=Pr(W1 \cup W2) \quad (2)$$

Apriori 알고리즘은 주어진 임계점보다 더 큰 값의 지지도와 신뢰도를 갖는 모든 연관 규칙을 효과적으로 찾는다. 이러한 경우, 신뢰도와 지지도의 임계값에 따라 발생되는 규칙의 수는 크게 달라지므로 임계값을 적절하게 설정하는 것이 중요하다. 전처리가 시행된 1000개의 로그를 대상으로, 신뢰도를 0으로 지정하고 지지도를 0~1까지 0.1씩 증가시켜가며 실험하였다. 실험 결과, 지지도가 0.1인 경우 가장 많은 연관 규칙이 추출되었다. 또한, 가장 많은 연관 단어가 추출된 지지도 0.1을 고정시키고, 신뢰도를 0에서 1까지 0.1씩 증가시켜가며 실험한 결과, 신뢰도가 0.1인 경우 가장 많은 연관 단어가 추출되었고 점차적으로 추출된 연관 단어의 수가 감소되었다. Fig. 4는 신뢰도와 지지도 값의 변경에 따라 추출되는 연관 단어의 수를 나타낸다. Fig. 4에서 s_A는 지지도 변화에 따른 연관 단어의 수를 나타내고, c_A는 신뢰도 변화에 따른 연관 단어의 수를 나타낸다. 가장 최대수의 연관 단어가 추출되는 신뢰도와 지지도는 모두 0.1이므로 본 논문에서 제안한 방법에서는 신뢰도와 지지도를 모두 0.1로 설정하여 연관 단어를 추출한다.

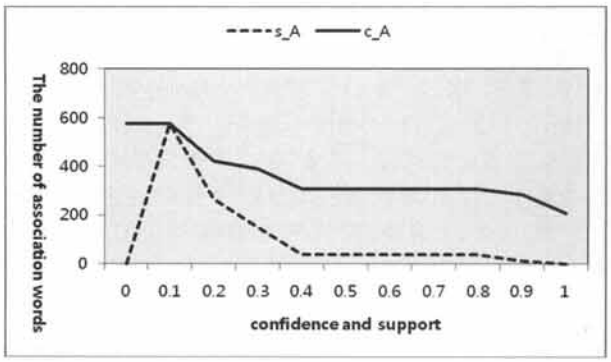

Fig. 4. The number of association words extracted by changing confidence and support

이와 같은 임계값을 사용한 이유는 해킹 흔적 로그를 추천하기 위해서는 빈도수가 큰 연관 단어를 찾는 경우도 중요하지만 희소 연관 단어가 해킹 흔적일 경우도 많으므로 빈도가 작은 연관 단어도 포함하는 것이 중요하기 때문이다.

4. 침입 흔적 로그 추천

본 장에서는 3장에서와 같이 추출한 연관 단어에 대해 침입 흔적 확률을 계산하고, 계산한 확률에 가중치를 추가하여 연관 단어 지식 베이스를 구축하는 방법을 기술한다. 또한, 포렌식 분석을 하는 조사관들에게 비구조화된 대량의 데이터로부터 적합한 정보를 효율적으로 감별하는 것을 도움으로써 시간을 절약하고 분류의 오류를 저하시킬 수 있도록 피셔의 역 카이제곱 분류 알고리즘을 이용하여 침입 흔적 확률이 높은 로그를 추출하는 방법을 기술한다. 추출한 로그 중 확률값이 높은 로그를 조사관에게 추천한다.

4.1 침입 흔적 연관 단어 지식 베이스 구축

Paul Graham은 메일이 특정 단어를 포함하는 경우 그 메일이 스팸인지 정상 메일인지를 판별하기 위하여 베이지안 통계를 이용하였다[8,9,19]. 또한, Gray Robinson[10]은 Paul Graham이 사용한 베이지안 통계를 이용한 스팸 여과 방식이 독립성 가정, 희소 단어 처리, 단어 확률의 계산, 그리고 비대칭의 부분에서 문제점이 있음을 지적하고 해결 방법을 제안하였다. 그가 제안한 방법은 모든 단어들의 확률을 계산하고, 피셔의 역 카이제곱 분류 알고리즘[11]을 스팸인 단어와 햄인 단어에 모두 적용하여 메일이 햄에 가까운지 스팸에 가까운지를 확률의 결합을 통해 판별하는 것이다.

본 논문에서는 3장에서와 같이 추출한 연관 단어의 침입 흔적 확률 계산을 위해 Paul Graham이 사용한 베이지안 통계법을 사용하지 않고 신뢰도와 지지도를 병합하는 방법을 사용한다. 다음으로, 희소 연관 단어의 처리와 빈도가 매우 높은 단어의 처리를 위하여 Robinson이 제안한 방법을 이용한다. Robinson 방법에서의 메일을 로그로 적용하고, 단어를 3.2절에서 추출한 연관 단어로 정의하여 그 확률을 계산한다. 이와 같이 계산된 값을 연관 단어의 가중치로 정의하고, 가중치가 추가된 연관 단어 지식 베이스를 구축한다.

연관 단어의 침입 흔적 확률은 식 (3)과 같이 지지도와 신뢰도를 곱한 식으로 정의한다. 식 (3)에서 $p(A_k)$는 k번째의 연관 단어 A_k가 해킹 흔적 연관 단어일 확률값을 나타낸다. 부가적으로, 연관 단어 A_k가 다른 연관 단어의 부분 집합인 경우의 확률도 누적시킴으로써 침입 흔적 연관 단어이나 확률값이 작아서 침입 흔적 연관 단어에서 제외될 경우를 예방한다. 식 (3)에서 N은 전체 연관 단어의 수를 나타낸다.

$$p(A_k) = \sum_{\substack{i=1 \\ A_k \in A_i}}^{N} Support_{A_i} \cdot Confidence_{A_i} \tag{3}$$

반면, 식 (3)의 $p(A_k)$값은 0의 최소값부터 다양한 범위의 값을 나타내므로 확률로 사용하기 어렵다. 따라서 0부터 1 사이의 값으로 변환시키기 위해서는 정규화가 필요하다. 정규화란 일반적인 수학이나 물리에서 데이터를 일정 크기로 나누어 주어 단위화하는 것을 의미한다[20]. 정규화의 방법으로는 평균값을 이용한 정규화, 중간값을 이용한 정규화, 사분위수(Quantile) 정규화, 그리고 최소-최대(Min-Max) 정규화 등이 있다. 일반적으로 평균값을 이용한 정규화를 많이 이용하는데, 자료에 약간의 잡음이 있을 경우 이상치(outlier)에 영향을 받을 수 있다는 단점을 갖는다. 반면, 중간값을 이용한 정규화는 데이터가 과도하게 크거나 작은 값이 있을 때 적당한 방법이며, 평균값을 이용한 방법과는 달리 이상치에 영향을 받지 않는다는 특성을 갖고 있다. 사분위수 정규화는 데이터 집합이 서로 다른 잡음에 의하여 분포의 전체 위치가 변할 가능성이 있을 경우 사용하는 방법이다. 최소-최대(Min-Max) 정규화[21]는 공학 단위에서 측정된 데이터를 처리하여 0.0에서 1.0까지의 값으로 변환시킴으로써 서로 다른 규모로 측정된 값들을 비교하는 데 유용한 방법이다.

위와 같은 정규화의 방법 중 데이터의 특성에 상관없이 모두 적용할 수 있는 방법은 존재하지 않으므로 데이터의 특성을 면밀히 검토하여 어떠한 방법을 사용할 것인지를 결정해야 한다. 본 논문에서는 식 (3)에 의해 계산된 확률의 범위를 0.0부터 1.0의 값으로 변경하고 데이터간의 관계를 유지하도록 하기 위하여 정규화의 방법 중 최소-최대 정규화를 사용한다. 정규화의 최대값은 그대로 1로 지정하나 최소값은 0.5의 값으로 지정한다. 이러한 이유는 침입 흔적 로그 집합으로부터 추출된 연관 단어는 침입 흔적 연관 단어일 가능성이 50%이상이라고 할 수 있기 때문이다. Fig. 5는 최소-최대 정규화 알고리즘을 나타낸다. 반면, Fig. 5에서 Pmax(최고의 침입 흔적 확률)와 Pmin(최소의 침입 흔적 확률)의 차이가 0인 경우, 침입 흔적 확률의 값이 일정함을 의미하므로 정규화된 결과를 침입 흔적 로그 추천에 사용할 수 없다. 따라서 이러한 경우는 최소-최대 정규화 알고리즘을 사용하지 않는다. 또한, Pnew_max는 최대의 확률값이므

로 1의 값으로, Pnew_min은 최소의 확률값이므로 0.5의 값을 지정하여 최소-최대 정규화를 시행한다. Fig. 5에서 $p'(A_k)$는 정규화가 시행된 후의 침입 흔적 확률을 나타낸다.

```
minmax(p(Ak),Pmax,Pmin,Pnew_max,Pnew_min)
1       p(Ak)<-p(Ak)-Pmin
2       y <- Pnew_max-Pnew_min
3       z   <- Pmax-Pmin
4       p(Ak)<-p(Ak)*y/z
5       p'(Ak)<-p(Ak)+Pnew_min
6       return p'(Ak)
end minmax
```

Fig. 5. Min-Max normalization algorithm

Table 3은 식 (3)에 의해 계산한 $p(A_k)$에 Fig. 5의 최소-최대 정규화 알고리즘을 적용하여 계산한 $p'(A_k)$를 나타낸다.

Table 3. Example of normalization for probability of intrusion of association words

rule_No(k)	Association word	$p(A_k)$	Min_Max normalization $p'(A_k)$
1	{404}=>{passwd, GET}	0.35	0.575385
2	{404, GET}=>{passwd}	0.35	0.575385
3	{passwd}=>{404, GET}	0.6	0.642692
4	{passwd, GET}=>{404}	0.6	0.642692
5	{passwd, 404}=>{GET}	0.63	0.650769
6	{404}=>{etc, GET}	0.07	0.500000
7	{404, GET}=>{etc}	0.35	0.575385
8	{etc}=>{404, GET}	0.6	0.642692
9	{etc, GET}=>{404}	0.2	0.535000
10	{etc, 404}=>{GET}	0.21	0.537692
11	{etc}=>{GET}	1.53	0.660462
12	{GET}=>{etc}	0.79	0.580769
13	{404}=>{GET}	16.32	0.999999
14	{GET}=>{404}	9.84	0.968708
15	{top}=>{GET}	0.66	0.566769
16	{GET}=>{top}	0.26	0.523692
17	{GET}=>{구문, 근처}	0.33	0.531231
18	{근처}=>{구문, GET}	0.87	0.589385
19	{근처, GET}=>{구문}	0.87	0.589385
20	{구문}=>{근처, GET}	0.87	0.589385
21	{구문, GET}=>{근처}	0.87	0.589385
22	{구문, 근처}=>{GET}	0.87	0.589385
23	{근처}=>{잘못, GET}	0.16	0.512923
24	{500}=>{잘못, 근처}	0.07	0.503231
25	{500, 근처}=>{잘못}	0.16	0.512923
26	{500, 잘못}=>{근처}	0.17	0.514000
27	{구문}=>{500, 잘못}	0.16	0.512923
28	{잘못}=>{500, 구문}	0.85	0.587231

반면, Table 3의 정규화된 침입 흔적 확률을 그대로 사용하는 경우 두 가지 원인으로 인하여 정확도 저하의 문제가 발생된다. 첫 번째 원인은 전체 로그에서 매우 높은 빈도로 나타나는 연관 단어로 인해 발생한다. 정보 검색 분야의 한 연구[22]에서는 단어가 전체 문헌에서 출현하는 빈도가 매우 높다면 그 단어는 낮은 가중치를 부여해야 하며, 특정 문헌에서는 출현 빈도가 높으나 전체 문헌에서는 출현 빈도가 낮다면 높은 가중치를 부여해야 한다는 연구를 기술하였다. 이와 같은 정보 검색의 이론을 침입 흔적 확률에 적용한다면 전체 로그에서 매우 높은 빈도로 출현하는 연관 단어의 가중치는 낮추고, 특정 로그들에는 출현 빈도가 높으며 전체 로그에는 출현 빈도가 낮은 경우의 연관 단어 가중치는 높이는 과정이 필요하다. 두 번째 원인은 빈도가 매우 희소함에도 불구하고 무조건 0.5이상의 확률을 부여함으로 인해 발생한다. 본 논문에서 제안한 방법에서는 이와 같은 두 가지 원인으로 인한 정확도 저하 문제를 해결하기 위하여 Robinson이 스팸 메일에 사용한 신뢰도 계산 방법을 제안한 방법에 적용한다. 이를 위하여, Fig. 5의 최소-최대 정규화가 완료된 침입 흔적 확률을 대상으로 신뢰도를 식 (4)와 같이 계산하고, 이 값을 가중치로 정의한다. 식 (4)는 Robinson이 스팸 메일에 사용한 신뢰도 계산식을 침입 흔적 로그에 적용한 식으로, 정규화된 침입 흔적 확률 $p'(A_k)$에 가중치를 추가함으로써 가중치가 추가된 침입 흔적 확률 $f(A_k)$를 정의한다.

$$f(A_k)=\frac{(s\cdot x)+(n_k\cdot p'(A_k))}{s+n_k} \tag{4}$$

식 (4)에서 n_k는 연관 단어 A_k와 이 연관 단어를 부분집합으로 하는 연관 단어의 총 수를 나타낸다. 또한, x는 배경 지식을 기반으로 하는 연관 단어의 초기 확률로 연관 단어가 침입 흔적 로그에 나타날 초기 확률을 의미한다. s는 배경 지식에 대한 신뢰 강도로 출현 빈도가 매우 낮은 희소 연관 단어가 다시 나타날 때 이를 포함하는 로그가 침입 흔적이 있는 로그로서 추천될 수 있는가에 대한 확률을 의미한다. 배경 지식으로부터 x를 설정하고, 설정된 x에 대한 신뢰의 강도를 나타내는 s의 값은 변화될 수 있으며, 최적의 s와 x를 식 (4)에 적용할 때 침입 흔적 로그 추천의 정확도는 높아진다.

식 (4)에서 최적의 s와 x의 값을 찾기 위해 검사 방법의 유용값 및 절단값(cut-off value) 판단을 하는 ROC (Receiver Operating Characteristic) 곡선[23]을 사용한다. ROC 곡선은 테스트의 각각 다른 가능한 절단점에 대한 위양성율(100-특이도(specificity), X축)과 그에 대한 실제 양성률(민감도(sensitivity) 또는 100-위음성률, Y축)을 그래프로 표현한 것이다. Table 4는 ROC 곡선을 그리기 위하여 초기값을 s=1, x=0으로 시작하여 0.1의 간격으로, s의 값은 감소시키고 x의 값은 증가시켜가면서, 총 10회에 걸쳐 침입 흔적 로그를 분류한 실험 결과를 나타낸다. Table 4에서 s의 값은 감소시키고 x의 값을 증가시켜가면서 실험한 이유는 s와 x가 상호 보완적인 관계이기 때문이다. 예를 들어 s의 값과 x의 값이 모두 1인 경우, 연관 단어가 한번 출현했으나 침입 흔적 로그에 속한다고 한다면 이 연관 단어는 한번 출현한 것만으로 무조건 1의 확률값을 갖는다.

Table 4. The result of classification rate by changing s and x

s	x	Normal log-> Normal log			Intrusion log -> Intrusion log		
		The number of classification	False positive (%)	False negative (%)	The number of classification	False positive (%)	False negative (%)
1	0	492	0	0.66	54	31.48	30.37
0.9	0.1	762	0.07	1.42	321	37.38	39.69
0.8	0.2	7658	0.02	2.74	1031	33.27	20.91
0.7	0.3	8642	0.02	4.45	7381	10.96	9.97
0.6	0.4	9921	2.15	9.51	8578	10.45	6.09
0.5	0.5	9724	8.78	14.98	7286	10.50	5.13
0.4	0.6	8772	10.50	37.2	4231	16.50	2.54
0.3	0.7	5643	41.25	67.35	5312	18.37	1.78
0.2	0.8	4764	20.40	70.82	4728	16.18	1.28
0.1	0.9	2554	38.61	83.52	453	0.22	0.78
0	1	53	0	99.65	0	0.00	99.76

이와 같은 결과는 추천의 정확도를 낮추는 결과를 갖는다. 따라서, s의 값이 클 경우 배경 정보에 대한 연관 단어의 초기 확률 x는 낮아야 하며, s의 값이 작을 경우 단어의 초기 확률 x를 높임으로써 희소 단어와 빈도가 매우 높으나 의미가 없는 단어에 대한 확률을 보완할 필요가 있다. Table 4에서는 배경 정보에 따른 신뢰 강도인 s가 0일 경우의 미분류율은 99.65%와 99.76%로 미분류 로그가 많아짐을 알 수 있다.

Fig. 6은 Table 4를 기반으로 작성된 ROC 곡선을 나타낸다. Fig. 6에 나타난 숫자의 단위는 백분율(%)이다.

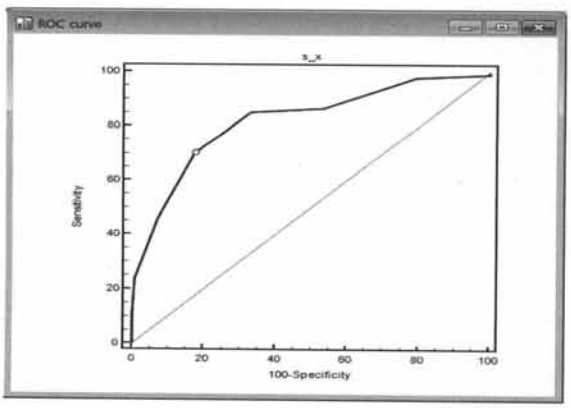

Fig. 6. ROC curve by changing s and x

Fig. 6의 ROC 곡선에서 절단값은 s가 0.7, x는 0.3이었다. 따라서 x와 s를 각각 0.3와 0.7로 결정하여 정규화된 침입 흔적 확률에 대해 가중치를 적용하였다.

Table 5는 정규화된 침입 흔적 확률 $p'(A_k)$를 식 (4)에 대입하여 계산된 가중치가 추가된 침입 흔적 확률 $f(A_k)$를 나타낸다. 침입 흔적 로그 집합을 대상으로 Table 5와 같이 가중치가 추가된 침입 흔적 확률을 계산한 후에 이를 기반으로 침입 흔적 연관 단어 지식 베이스를 구축한다.

Table 5. Example of calculating the probability of intrusion with weights

Association word(A_k)	Min-Max normalization $p'(A_k)$	s	x	n_k	$f(A_k)$
{404}=>{passwd, GET}	0.575385	0.7	0.3	3	0.523285
{404, GET}=>{passwd}	0.575385	0.7	0.3	3	0.523285
{passwd}=>{404, GET}	0.642692	0.7	0.3	3	0.577859
{passwd, GET}=>{404}	0.642692	0.7	0.3	3	0.577859
{passwd, 404}=>{GET}	0.650769	0.7	0.3	3	0.584407
{404}=>{etc, GET}	0.500000	0.7	0.3	1	0.417647
{404, GET}=>{etc}	0.575385	0.7	0.3	3	0.523285
{etc}=>{404, GET}	0.642692	0.7	0.3	3	0.577859
{etc, GET}=>{404}	0.535000	0.7	0.3	1	0.438235
{etc, 404}=>{GET}	0.537692	0.7	0.3	1	0.439819
{etc}=>{GET}	0.660462	0.7	0.3	9	0.634449
{GET}=>{etc}	0.580769	0.7	0.3	7	0.555245
{404}=>{GET}	0.999999	0.7	0.3	75	0.993526
{GET}=>{404}	0.968708	0.7	0.3	64	0.961473
{top}=>{GET}	0.566769	0.7	0.3	3	0.516299
{GET}=>{top}	0.523692	0.7	0.3	3	0.481372
{GET}=>{구문, 근처}	0.531231	0.7	0.3	4	0.496792
{근처}=>{구문, GET}	0.589385	0.7	0.3	5	0.553846
{근처, GET}=>{구문}	0.589385	0.7	0.3	5	0.553846
{구문}=>{근처, GET}	0.589385	0.7	0.3	5	0.553846
{구문, GET}=>{근처}	0.589385	0.7	0.3	5	0.553846
{구문, 근처}=>{GET}	0.589385	0.7	0.3	5	0.553846
{근처}=>{잘못, GET}	0.512923	0.7	0.3	1	0.425249
{500}=>{잘못, 근처}	0.503231	0.7	0.3	1	0.419548
{500, 근처}=>{잘못}	0.512923	0.7	0.3	1	0.425249
{500, 잘못}=>{근처}	0.514000	0.7	0.3	1	0.425882
{구문}=>{500, 잘못}	0.512923	0.7	0.3	1	0.425249
{잘못}=>{500, 구문}	0.587231	0.7	0.3	5	0.551957

4.2 역 카이제곱 알고리즘을 이용한 침입 흔적 확률 계산

테스트 로그 집합으로부터 침입 흔적 로그를 추천하기 위하여 우선적으로 Table 2의 전처리 과정을 완료한다. 다음으로, Apriori 알고리즘을 이용하여 테스트 로그 집합으로부터 연관 단어를 추출한다. 이와 같이 추출된 연관 단어를 기반으로 피셔의 역 카이제곱 분류 알고리즘을 이용하여 테스트 로그 집합의 로그가 침입 흔적 로그일 확률과 정상 로그일 계산한다.

그런데 역 카이제곱 분류 알고리즘에 적용하기 위해서는 우선적으로 역 카이제곱 검증을 거쳐야 한다[10]. 역 카이제

곱 검증이란 하나의 귀무가설을 세우고, 증명에 의하여 이 귀무가설을 기각하고 대립가설을 선택하는 과정을 의미한다. 검증을 위해 필요한 귀무가설은 "$f(A_k)$는 정확하고, 현재의 로그는 여러 연관 단어들의 임의의 선택일 뿐이며, 선택된 연관 단어는 나머지 연관 단어들과 독립적이다. 따라서 $f(A_k)$의 값은 균등분포라고 할 수 없다"이다. 이와 같은 가설은 실제 어떤 로그도 침입 흔적 로그나 정상 로그 중 어느 한 분류에 속하지 않을 수 없으며, 구성된 연관 단어가 완전히 무작위로 추출된 연관 단어라는 것을 의미이다. 그러나, 매우 특수한 경우를 제외하고는 대부분의 로그가 침입 흔적 로그나 정상 로그 중 하나의 범주로 분류될 수밖에 없으며, 구성된 연관 단어가 완전히 무작위로 추출될 수는 없고 서로 관련되는 연관 단어들로 구성될 수밖에 없다. 따라서 이와 같은 귀무가설은 기각되고, 로그로부터 추출한 연관 단어 집합 전체를 대상으로 식 (5)의 피셔의 역 카이제곱 계산식을 이용하여 그 침입 흔적 확률을 계산한다. 반면, 침입 흔적 연관 단어 지식 베이스에 포함되지 않은 연관 단어는 침입 흔적 확률이 무조건 0이 되고, 식 (5)의 수식에 의해 계산을 할 수 없다. 이를 보완하는 방법으로, 침입 흔적 확률이 0인 연관 단어는 정상 연관 단어로 간주하고, H의 값을 1로 정의한다.

$$H = C^{-1}(-2\ln\prod_{A_i \in A_k}^{N} f(A_k), 2n_k) \tag{5}$$

식 (5)에서 C-1()은 역 카이제곱 함수이며, 카이제곱 분포의 확률변수로부터 p-value(유의수준)를 추출하는 데 사용된다. n_k는 N개의 전체 연관 단어 중 연관 단어 A_k를 포함하는 연관 단어의 수이다. Fig. 7은 식 (5)의 역 카이제곱 함수의 알고리즘을 나타낸다.

반면, 침입 흔적이 있는 연관 단어들은 $f(A_k)$의 값이 1에 가깝기 때문에 그들의 곱에 크게 영향을 받지 못한다. 이런 단점을 보완하기 위해 먼저 모든 확률의 역확률 의미인 $(1-f(A_k))$의 값을 계산하여 병합하는 방법을 사용한다.

식 (6)은 선택한 로그가 정상 로그일 가능성을 계산하는 식이다. 식 (5)에서 $f(A_k)$는 연관 단어 A_k가 침입 흔적이 있는 연관 단어일 확률을 나타내는 반면, 식 (6)에서 $(1-f(A_k))$는 연관 단어 A_k가 정상 연관 단어일 확률을 나타낸다.

```
Chi2q(x,  v)
1       i<-0
2       m<-0.0, s<-0.0, t<-0.0
3       m <- x / 2.0
4       s <-  Math.Exp(-m)
5       t <- s
6       FOR   i <- 1 to i < (v / 2) DO
7          t <- t x (m / i)
8          s <- s + t
9       END FOR
10      return   ((s < 1.0) ? s : 1.0);
end Chi2q
```

Fig. 7. Inverse chi-square function algorithm

$$S = C^{-1}(-2\ln\prod_{A_i \in A_k}^{N}(1 - f(A_k)), 2n_k) \tag{6}$$

반면, 침입 흔적 연관 단어 지식 베이스에 포함되지 않는 연관 단어는 이들의 $(1-f(A_k))$값이 동일하게 1의 값으로 계산되므로, 다양한 훈련 로그 집합으로부터 침입 흔적 연관 단어 지식 베이스를 구축하는 연구가 필요하다.

4.3 테스트 로그 집합에서의 침입 흔적 로그 추천

테스트 로그 집합에 대해 4.2절에서와 같이 역 카이제곱 분류 알고리즘을 이용하여 침입 흔적 로그일 확률과 정상 로그일 확률을 계산하였다. 다음으로, 이들을 병합하여 그 결과를 기반으로 침입 흔적 로그를 추출한다. 최종적으로, 추출된 로그의 확률값이 높은 순서에서 낮은 순서로 정렬한 후 정렬된 로그의 집합을 조사관에게 추천한다.

식 (7)은 식 (5)의 H와 식 (6)의 S를 병합하여, 선택한 로그가 정상 로그인가 침입 흔적이 있는 로그인가를 판단하는 식이다.

$$I = \frac{1 + H - S}{2} \tag{7}$$

식 (7)에서 I는 해당 로그가 침입 흔적이 있는 로그에 가까울수록 1의 값에 근접하여, 정상 로그에 가까울수록 0의 값에 근접함을 알리기 때문에 지시자라고 할 수 있다. 매우 희박한 경우이나 I가 0.5의 값을 나타낼 때 이 로그는 정상 로그인지 아닌지를 판별할 수 없는 확신이 없는 로그를 나

Table 6. Example of a test log set

No	date	time	c-ip	cs-method	cs-uri-stem	cs-uri-query	sc-status
1	2010-02-16	1:58:13	xxx.xxx.249.99	GET	shop/shop_topview.asp	top=7	200
2	2010-02-16	5:04:13	xxx.xxx.249.83	GET	shopping_cart/cart_add.asp	|12|80040e14|'And'_키워드_근처의_구문이_잘못되었습니다.	500
3	2010-02-16	5:10:34	xxx.xxx.249.85	GET	shop_board/shop_board_del.asp	page=11&num=226	200
4	2010-02-16	7:23:30	xxx.xxx.249.196	POST	login_check.asp	|18|80040e14|줄_1:_'admin'_근처의_구문이_잘못되었습니다.	500
5	2010-02-16	6:14:22	xxx.xxx.249.196	GET	webplus	script=/../../../../etc/passwd	404

타낸다. 이와 같은 경우는 분류에서 제외하여 정확하지 않은 분류로 인해 정확도가 저하되는 경우를 예방한다. Table 6은 침입 흔적 로그를 추출하기 위한 테스트 로그 집합의 예를 나타낸다.

Table 6의 로그를 대상으로 전처리를 실시한 후, 신뢰도와 지지도를 각각 0.1로 지정하고 연관 단어를 추출한다. Table 7은 추출된 연관 단어의 예를 나타낸다.

Table 8은 각 로그로부터 Table 7과 같이 추출한 연관

Table 7. Example of association words extracted after preprocessing

No	Association word
1	{GET, 200, top}=>{topview.asp, shop}{topview.asp,GET}=>{top, shop}{top}=>{topview.asp, shop, 200}{top, shop}=>{GET, 200}{GET}=>{200, top}{GET}=>{200}{top}=>{200}
2	{잘못}=>{구문}{키워드}=>{잘못}{And}=>{잘못, 키워드}{잘못, And}=>{키워드}{And}=>{잘못, 근처}{잘못}=>{키워드, And, 구문}{근처, 구문}=>{키워드, 잘못}{And, 근처}=>{키워드, 잘못, 구문}{키워드, And, 잘못, 구문}=>{근처} {500}=>{근처}
3	{GET}=>{page}{page}=>{num}{num}=>{200}{GET, page, num}=>{xxx.xxx.249.85, 200}{num,xxx.xxx.249.85}=>{200, GET}
4	{admin}=>{POST}{login_check.asp}=>{줄}{admin}=>{login_check.asp}{login_check.asp}=>{admin,POST}{admin,500}=>{login_check.asp}{줄, l ogin_check.asp}=>{POST}{login_check.asp, 줄}=>{POST, admin}{login_check.asp, POST}=>{500, 줄} login_check.asp, POST, admin}=>{줄, 500}
5	{passwd}=>{etc}{GET}=>{passwd, script}{etc}=>{webplus, GET}{GET}=>{script, passwd, etc}{script, GET, passwd}=>{etc} GET}=>{webplus, passwd, etc, script}{webplus, passwd, etc, script}=>{GET}

Table 8. H, S, I by calculated based on association word knowledge base

No	Extracted association word	$f(A_k)$	H	$1-f(A_k)$	S	I
1	{topview.asp, GET}=>{shop, 200, top}	0.21678	0.0004	0.78322	0.999	0.00018
	{GET, 200, top}=>{topview.asp, shop}	0.37369		0.62631		
	{topview.asp, GET}=>{top, shop}	0.26579		0.73421		
	{top}=>{topview.asp, shop, 200}	0.32068		0.67932		
	{top, shop}=>{GET, 200}	0.11677		0.88323		
	{GET}=>{200, top}	0.00168		0.99832		
	{GET}=>{200}	0.01766		0.98234		
	{top}=>{200}	0.00235		0.99765		
2	{500}=>{근처}	0.85289	0.9679	0.14710	0.334	0.81696
	{키워드}=>{잘못}	0.81984		0.18015		
	{And}=>{잘못, 키워드}	0.58184		0.41815		
	{And}=>{잘못, 근처}	0.68184		0.31815		
	{잘못}=>{키워드, And, 구문}	0.54738		0.45262		
	{근처, 구문}=>{키워드, 잘못}	0.58615		0.41385		
	{And, 근처}=>{키워드, 잘못, 구문}	0.51292		0.48708		
	{키워드, And, 잘못, 구문}=>{근처}	0.58184		0.41815		
3	{GET}=>{page}	0.00007	0.0005	0.99993	0.987	0.00677
	{page}=>{number}	0.07880		0.92120		
	{number}=>{200}	0.35770		0.64230		
	{GET, page, number}=>{xxx.xxx.249.85, 200}	0.16687		0.83313		
	{num, xxx.xxx.249.85}=>{200, GET}	0.52270		0.47730		
4	{admin}=>{POST}	0.65521	0.9520	0.34479	0.528	0.71185
	{login_check.asp}=>{줄}	0.71312		0.28688		
	{admin}=>{login_check.asp}	0.58172		0.41828		
	{login_check.asp}=>{admin,POST}	0.72611		0.27389		
	{admin,500}=>{login_check.asp}	0.54332		0.45668		
	{줄, login_check.asp}=>{POST}	0.62114		0.37886		
	{login_check.asp, 줄}=>{POST, admin}	0.51351		0.48649		
	{login_check.asp, POST}=>{500, 줄}	0.56242		0.43758		
	{login_check.asp, POST, admin}=>{줄, 500}	0.49372		0.50628		
5	{passwd}=>{etc}	0.61320	0.8408	0.3868	0.696	0.57200
	{GET}=>{passwd, script}	0.62730		0.3727		
	{etc}=>{webplus, GET}	0.52341		0.4766		
	{GET}=>{script, passwd, etc}	0.56210		0.4379		
	{GET}=>{webplus, passwd, etc, script}	0.45120		0.5488		
	{webplus, passwd, etc, script}=>{GET}	0.49380		0.5062		
	{script, GET, passwd}=>{etc}	0.47620		0.5238		

단어를 대상으로 식 (5), 식 (6), 그리고 식 (7)에 대입하여 계산된 H, S, I의 값을 나타낸다. Table 8에서 로그1과 로그3은 I의 값이 0.5보다 작으므로 침입 흔적이 없는 로그로 판단하여 추천하지 않는다. 반면, 로그2, 로그4, 그리고 로그5는 침입 흔적이 있는 로그로 판단하여, 이들을 I의 같이 높은 순서에서 낮은 순서로 정렬한다. 최종적으로, (로그2, 로그4, 로그5)를 조사관에게 추천한다.

5. 성능 평가

제안한 디지털 포렌식 텍스트 마이닝 기반 침입 흔적 로그 추천 방법(AK_Robinson)의 성능을 평가하기 위하여 기존의 텍스트 마이닝 기술을 로그 파일의 추천 방법에 적용하여 이들을 비교함으로써 그 성능을 평가하였다. 텍스트 마이닝 기술을 디지털 포렌식에 적용한 기존의 방법은 수상한 사용자를 찾아내기 위하여 로그들을 결합하고 그 결과를 기반으로 의사 결정 트리를 이용하는 기술(Decisiontree)[5], 디지털 포렌식 분석에 텍스트 군집을 사용하는 방법(Textclustering)[4], 그리고 로그 데이터로부터 프로파일을 생성하고 그 규칙 집합을 사용하는 방법(AR_Profile)[6] 등이 있다. 추천 시스템의 정확도는 추천 시스템이 사용자에게 얼마나 높은 질의 추천이 가능한가의 정도를 측정하며, 그 종류로는 예측값의 정확도를 측정하는 MAE(Mean Absolute Error), 분류의 정확도를 측정하는 정확도(precision)와 재현율(recall), ROC 곡선 등이 있다[24]. 본 논문에서 제안한 방법은 예측값의 정확도보다는 분류의 정확도가 추천에 더욱 큰 영향을 주기 때문에 정확도와 재현율, 그리고 ROC 곡선의 척도를 사용하여 성능을 평가하였다.

5.1 성능 평가 자료

성능 평가를 위하여 2010년 2월 16일부터 2010년 3월 1일까지 총 14일간 마이크로소프트의 IIS로부터 웹로그를 수집하여, 훈련 로그 집합과 테스트 로그 집합으로 구성하였다. 침입 흔적 로그 추천의 실험을 위해 여러 유형의 웹 해킹을 시도하였으며, 이들을 누적한 총 로그 수는 48,231이다. 전처리를 위하여 데이터 정제과정을 거쳤을 때 48,231개의 로그 중 이미지 자료는 7,800건, 요청 실패는 764건, 그리고 불완전한 자료는 232건이었다. 이와 같은 자료들을 제외한 총 로그 수는 40,384이다. 총 40,384 로그 중 25,000 로그를 훈련 로그 집합으로 구성하였으며, 나머지는 테스트 로그 집합으로 구성하였다. 또한, 훈련 로그 집합을 침입 흔적이 있는 로그 집합과 정상 로그 집합으로 수작업으로 분류하였다.

5.2 Roc 곡선을 이용한 성능 평가

Roc 측정은 곡선을 이용하여 추천 시스템이 침입 흔적 로그를 얼마나 정확하게 많이 추천할 수 있는가를 측정한다. 이 방법은 분류된 로그의 정확도를 평가하기 위하여 민감도와 특이도를 사용한다. 민감도는 실제로 해킹 흔적이 있는 로그를 얼마나 잘 찾아내는가의 기준을 나타내며, 특이도는 해킹 흔적이 없는 로그를 얼마나 잘 분류하는가의 기준을 나타낸다. AUC(Area Under the Curve)는 ROC 곡선의 아래면적을 나타내며 추천 시스템이 '좋은(good) 로그'을 보유할수록 증가한다. 여기서 '좋은 로그'와 '나쁜(bad) 로그'을 결정하는 것이 필요하다. 본 논문에서는 '좋은 로그'와 '나쁜 로그'를 구분하기 위해 침입 흔적 로그를 침입 흔적 로그로 추천한 경우는 '좋은 로그', 정상 로그를 침입 흔적 로그로 추천한 경우는 '나쁜 로그'로 정의한다. 특히, AUC의 값이 1인 경우는 완전 정확한 추천 방법이라고 할 수 있으며, 0.5이하는 정확하지 않은 추천 방법이라고 할 수 있다[23].

Table 9는 초기값을 s=1, x=0으로 시작하여 0.1의 간격으로 s의 값은 감소시키고 x의 값은 증가시켜가면서 총 10회의 ROC 민감도를 테스트한 결과를 나타낸다.

Table 9. Result o f ROC sensitivity test

s	x	AK_Robinson		Decisiontree		Textclustering		AR_Profile	
		sensitivity (%)	specificity (%)	sensitivity (%)	specificity (%)	sensitivity (%)	specificity (%)	sensitivity (%)	specificity (%)
1	0	0	100	0	100	0	100	0	100
0.9	0.1	28.77	99.00	1.140	99.93	11.19	89.26	35.26	88.81
0.8	0.2	46.15	92.84	2.410	99.28	23.77	79.34	67.81	76.23
0.7	0.3	69.94	82.53	31.08	92.74	46.15	59.51	72.84	53.85
0.6	0.4	85.32	67.05	61.92	82.26	69.94	42.50	77.86	30.06
0.5	0.5	86.72	46.42	81.78	66.56	85.32	28.34	83.35	14.68
0.4	0.6	88.32	45.20	83.59	46.41	86.72	17.00	90.60	13.28
0.3	0.7	97.90	20.63	98.10	20.94	97.90	8.500	98.79	2.100
0.2	0.8	99.30	0	99.91	0	99.30	2.830	99.72	0.700
0.1	0.9	99.70	0	99.93	0	99.90	0	99.87	0.100
0	1	100	0	100	0	100	0	100	0
AUC		0.822677		0.777905		0.561457		0.68412	

Table 9에서 본 논문에서 제안한 AK_Robinson 방법의 AUC는 0.822677로, 다른 방법들의 AUC보다 높다. 이와 같은 결과는 AK_Robinson 방법이 완벽한 검사라고는 할 수 없으나 중등도의 정확한 검사라는 것을 증명한다.

Fig. 8은 Table 9를 기반으로 그려진 ROC 민감도 곡선을 나타낸다. Fig. 8에 나타난 숫자의 단위는 백분율(%)이다.

Fig 8에서 제안한 AK_Robinson, Decisiontree, Text lustering, 그리고 AR_Profile의 곡선 형태를 비교해 보면 대용량의 로그 파일을 대상으로 학습을 통해 연관 단어 지식베이스를 구축하고 테스트를 통해 침입 흔적 로그를 분류한 AK_Robinson 방법이 가장 정확한 형태의 곡선 모양을 보이며, 학습을 한 후 의사 결정 트리를 이용하여 분류하나 IP 필드만을 대상으로 하는 방법인 Decisiontree 방법은 AK_Robinson 방법 다음으로 정확한 형태의 곡선 모양을 나타내었다. 반면, 학습 방법만을 적용하고 있는 AK_Profile 방법과 Textclustring 방법의 곡선 모양은 침입 흔적 로그의 추천 방법에 사용하기에는 다소 미흡한 곡선의 형태를 보였다.

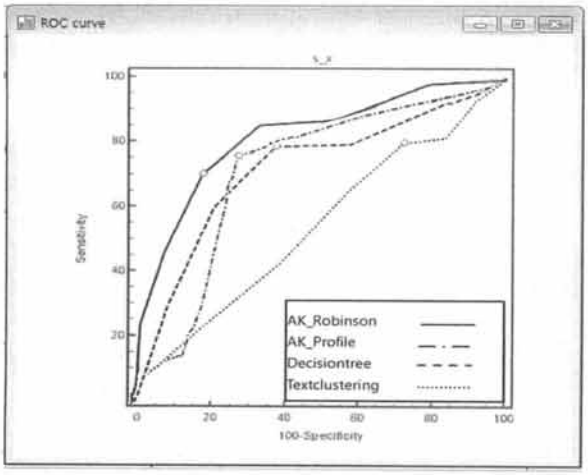

Fig. 8. ROC curve of AK_Robinson, Decisiontree, Textclustering, and AR_Profile

5.3 재현율과 정확도를 이용한 성능 평가

침입 흔적 로그 분류의 재현율(LR)과 정확도(LA)를 정의하는 식은 각각 식 (8)과 식 (9)이다. 식 (8), 식 (9) 에서 C_{I_Log}는 침입 흔적 로그로 분류된 로그, C_{N_Log}는 정상 로그로 분류된 로그를 나타낸다.

$$LR = \frac{C_{I\_Log->I\_Log}}{C_{I\_Log}} \tag{8}$$

$$LA = \frac{C_{I\_Log->I\_Log}}{C_{I\_Log->I\_Log} + C_{N\_Log->I\_Log}} \tag{9}$$

침입 흔적 로그의 재현율은 침입 흔적 로그로 분류한 비율을 나타내고, 침입 흔적 로그 분류의 정확도는 침입 흔적 로그로 분류한 로그가 실제로 침입 흔적 로그인 결과를 측정한 정도이다. 또한, 로그에 대한 분류의 정확도를 평가하기 위하여, 오분류율(False Positive, FP)과 미분류율(False Negative, FN)로 구분하여[25], 각각 식 (10)과 식 (11)로 정의하였다.

$$FP = \frac{C_{N\_Log->I\_Log}}{C_{N\_Log}} \tag{10}$$

$$FN = \frac{C_{I\_Log->N\_Log}}{C_{I\_Log}} \tag{11}$$

Table 10은 식 (8), 식 (9), 식 (10), 그리고 식 (11)를 이용하여 침입 흔적 로그를 추출하는 방법들의 성능을 계산한 결과를 나타낸다.

Table 10. Performance of methods for extracting intrusion logs

method	FN(%)	FP(%)	LA(%)	LR(%)
AR_Profile	15.32	28.8	84.02	72.3
AK_Robinson	9.97	12.52	98.79	77.5
Decisiontree	10.76	15.6	94.33	75.8
Textclustering	18.98	30.8	82.02	71.3

Table 10에서 AK_Robinson 방법과 Decisiontree 방법의 성능을 비교할 때 AK_Robinson 방법은 IP 필드뿐만 아니라 여러 개의 필드를 기반으로 학습을 하고 추천을 하므로 Decisiontree 방법보다 높은 정확도를 나타낸다. 반면, AR_Profile 방법과 Textclustring 방법은 단서가 없는 상태에서 포렌식 분석을 하는 분석가에게 증거를 추출하는 데 도움을 줄 수는 있으나 추천까지 진행했을 경우의 정확도는 제안한 방법과 Decisiontree 방법보다는 낮음을 알 수 있다.

6. 결 론

본 논문에서는 디지털 포렌식에서 텍스트 마이닝을 이용한 침입 흔적 로그 추천 방법을 제안하였다. Apriori 알고리즘과 Robinson의 신뢰도 계산법에 의해 학습함으로써 연관 단어 지식베이스를 구축하였으며, 이를 기반으로 피셔의 역 카이제곱 분류 알고리즘을 이용하여 테스트 로그 집합의 로그 중 침입 흔적이 있는 로그를 추출하고 그 순위를 부여하여 조사관에게 추천하였다. 제안한 방법은 포렌식 분석을 하는 조사관들에게 믿을만한 증거를 추천하였다. 이와 같은 증거는 단서가 없는 경우 비구조화된 대량의 데이터로부터 적합한 정보를 효율적으로 추출하는 것을 도움으로써 시간을 절약하고 데이터 분석에서의 번거로움을 줄일 수 있다는 장점을 갖는다. 또한, 제안한 방법은 침입 흔적 연관 단어 지식 베이스를 기반으로 테스트 로그의 침입 흔적 확률을 계산하므로 데이터의 모호성으로 인해 발생하는 정확도 저하 문제를 보완할 수 있었으며 역 카이제곱 분류 알고리즘을 이용한 확률의 결합으로 인하여 오분류율을 감소시킬 수 있었다.

향후, 웹로그 뿐 아니라 다른 유형의 로그에 적용하여 그 성능을 분석하는 방법이 필요하며, 미분류 로그에 대해 처리하는 연구가 필요하다.

참 고 문 헌

[1] G. Mohay, A. Anderson, B. Collie, O. De Vel, and R. McKemmish, Computer and Intrusion Forensics, Artech House, Norwood, MA, 2003.

[2] Linda Volonino, "Computer forensics and electronic discovery: The new management challenge," Computer & Security, Vol.25, No.2, 2006.

[3] Rayman D. Meservy and James V. Hansen, "Forensic Data Mining: Finding Intrusion Patterns in Evidentiary Data," In Proceedings of Americas Conference on Information Systems(AMCIS), 2010.

[4] Sergio Decherchi, Simone Tacconi, Judith Redi, Alessio Leoncini, Fabio Sangiacomo, and Rodolfo Zunino, "Text Clustering for Digital Forensics Analysis," Journal of Information Assurance and Security 5, 2010.

[5] Nikhil Kumar Singh, Deepak Singh Tomar, and Bhola Nath Ray, "An Approach to Understand the End User Behavior through Log Analysis," International Journal of Computer Application, Vol.5, No.11, 2010.

[6] Tamas Abraham and Olivier de Vel, "Investigative Profiling with Computer Forensic Log Data and Association Rules," In Proceeding of IEEE International Conference on Data Mining(ICDM), 2002.

[7] Jiawei Han, Data Mining:Concepts and Techniques, Morgan Kaufmann, 2001.

[8] Frederic P. Miller, Agnes F. Vandome, and John McBrewster(Ed.), Bayesian spam filtering, Alphascript Publishing, 2010.

[9] Jonathan A. Zdziarski, Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification, No starch press, 2005.

[10] Gary Robinson, "A statistical Approach to the Spam Problem," Linux Journal, Vol.107, 2003.

[11] Ramon C. Littell and J. Leroy Folks, "Asymptotic Optimality of Fisher's Method of Combining Independent Tests," Vol.66, No.336, 1971.

[12] Hyukgyu Cho, Heum Park, Hyukchul Kwon, "The Method of Verification for Legal Admissibility of Digital Evidence using the Digital Forensics Ontology," The KIPS transactions:Part D, Vol.16-D, No.2, 2009.

[13] Shaimaa Ezzat Salama and Mohamed I. Marie, "Web Server Logs preprocessing for Web Intrusion Detection," Computer and Information Science, Vol.4, No.4, 2011.

[14] M. Malarvizhi and S. A. Sahaaya Arul Mary, "Preprocessing of Educational Institution Web Log Data for Finding Frequent Patterns using Weighted Association Rule Mining Technique,"European Journal of Scientific Research, Vol.74, No.4, 2012.

[15] Juan Jose Garcia Adeva and Juan Manuel Pikatza Atxa, "Intrusion detection in web applications using text mining," Engineering Applications of Artificial Intelligence, Vol.20, No.4, 2007.

[16] Pang-Ning Tanch, Michael Steinbach, and Vipin Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

[17] G. V. Nadiammai, S. Krishnaveni, and M. Hemalatha, "A Comprehensive Analysis and study in Intrusion Detection System using Data Mining Techniques," International Journal of Computer Applications, Vol.35, No.8, 2011.

[18] Sandhya Peddabachigari, Ajith Abraham, and Johnson Thomas, "Intrusion Detection Systems Using Decision Trees and Support Vecter Machines," International Journal of Applied Science and Computations, Vol.11, No.3, 2004.

[19] Paul Grahamm, "A Plan for Spam," http://paulgraham.com/spam.html, 2002.

[20] B. M. Bolstad, R. A. Irizarry, M. Astrand and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," Bioinformatics, Vol.19, No.2, 2003.

[21] T. Jayalakshmi and Dr.A.Santhakumaran, "Statistical Normalization and Back Propagation for Classification," International Journal of Computer Theory and Engineering, Vol.3, No.1, 2001.

[22] S. E. Robertson and K. S. Jones, "Relevance Weightting of Search Terms," Journal of the American Society for Information Science, Vol.27, No.3, 1976.

[23] Sang Wook Song, "Using the Receiver Operating Characteristic (ROC) Curve to Measure Sensitivity and Specificity," Korean Journal of Family Med., Vol.30, No.11, 2009.

[24] Herlocker, J., Konstan J., Terveen L., and Riedl J. "Evaluating Collaborative Filtering Recommender Systems," ACM Transactions on Information Systems, Vol.22, No.1, 2004.

[25] J. Kim and S. Choi, "An Improved Bayesian Spam Mail Filter based on Chi-Square Statistics," Proceedings of KFIS Spring Conference 2005, Vol.15, No.1, 2005.

고 수 정

e-mail : sjko@induk.ac.kr

1990년 인하대학교 전자계산학과(학사)

1997년 인하대학교 정보컴퓨터교육(석사)

2002년 인하대학교 전자계산공학(박사)

2003년~2004년 Post Doc at University of Illinois at Urbana Champaign

2004년~2005년 Research Scientist at Colorado State University

2005년~현 재 인덕대학교 컴퓨터소프트웨어과 교수

관심분야 : Information security & Data mining & Machine Learning