JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 19, No. 8 : 1797~1804 Aug. 2015

하둡 기반의 효율적인 보안로그 분석시스템 설계 및 구현

안광민 · 이종윤 · 양동민 · 이봉환*

Design and Implementation of a Hadoop-based Efficient Security Log Analysis System

Kwang-Min Ahn· Jong-Yoon Lee · Dong-Min Yang · Bong-Hwan Lee*

Department of Information and Communications Engineering, Daejeon University, Daejeon 300-716, Korea

요 약

통합로그관리시스템은 보안 위험 사항을 예측하고 기관의 보안성 향상에 기여하여 적합한 보안 정책을 마련할 수 있도록 도와준다. 본 논문에서는 대용량의 로그 데이터를 저장할 수 있는 분산 데이터베이스 모델과 로그 수집 절차를 자동화하여 분석 시간을 줄일 수 있는 하둡 기반의 로그 분석 시스템을 설계하고 구현하였다. 제안하는 시스템에서는 HBase를 사용하여 데이터 용량에 따라 Scale-Out 방식으로 유연하게 저장할 수 있게 하였고 정규식을 이용하여 분석에 용이한 저장 기법을 제안하여 기존 시스템 대비 분석 속도를 높일 수 있다.

ABSTRACT

Integrated log management system can help to predict the risk of security and contributes to improve the security level of the organization, and leads to prepare an appropriate security policy. In this paper, we have designed and implemented a Hadoop-based log analysis system by using distributed database model which can store large amount of data and reduce analysis time by automating log collecting procedure. In the proposed system, we use the HBase in order to store a large amount of data efficiently in the scale-out fashion and propose an easy data storing scheme for analysing data using a Hadoop-based normal expression, which results in improving data processing speed compared to the existing system.

키워드 : 로그 분석, 비관계형 데이터베이스, 통합보안관리시스템, 클라우드 컴퓨팅, 하둡

Key word : Log Analysis, Non-relational Database, Enterprise Security Management System, Cloud Computing, Hadoop

Received 22 April 2015, **Revised** 14 May 2015, **Accepted** 29 May 2015

* **Corresponding Author** Bong-Hwan Lee(E-mail:blee@dju.kr, Tel:+82-280-4780)
Department of Information and Communications Engineering, Daejeon University, Daejeon 300-716, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2015.19.8.1797 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

로그(Log)는 사용자의 활동이나 패턴 정보를 파악할 수 있고 네트워크 통신에 대한 발자취로 볼 수 있다. 다양한 공격에서도 선제적 대응을 위해서는 기본적인 자료임에도 불구하고 제대로 활용되지 못하는 로그의 활용성이 재인식되면서, 로그 데이터 정보는 보안적 측면에서 악의적 침입이나 공격에 대해 사전에 예측하거나 예방 할 수 있는 용도로 활용하여 차세대 보안의 핵심으로 재조명되어 로그를 통합 관리하고 분석할 수 있는 통합로그관리[1] 솔루션을 도입하는 사례가 늘어나고 있다.

로그는 네트워크의 규모가 커짐에 따라 방대한 양이 발생하기 때문에 이러한 로그 정보를 분석하기 위해 대량의 로그 데이터를 유실없이 저장하고 실시간 처리 및 분석하는 방법에 대한 필요성이 대두되고 있다. 이에 따라 빅데이터(Big Data) 기술을 활용한 데이터 분석이 조명받고 있으며, 기존의 관계형 데이터베이스 처리 방식으로는 해결할 수 없는 비정형화된 데이터를 저장하고 분석할 수 있게 되었다.

빅데이터 처리 기술로 오픈 소스인 하둡(Hadoop)이 주목받아 단순한 데이터 저장 능력뿐 아니라 분석 결과까지 처리가 가능한 높은 수준의 기술로 진화하고 있고 기업에서는 빅데이터 분석에 들어가는 초기 비용을 줄이고 호환성도 제공받을 수 있다.

본 논문에서는 대용량의 로그 수집 및 분석을 위한 하둡 프레임워크를 구축하여 데이터의 양이 많아지면 저장 및 분석 처리 능력의 한계를 극복하기 위한 도구로 사용하여 대량의 보안로그 데이터 활용을 높여 효과적인 보안로그관리 시스템을 구현하였다. 제안하는 보안로그관리 시스템은 이기종 네트워크 장비에 Agent를 탑재하여 로그를 수집하고 분산 데이터베이스인 HBase에 분석이 용이할 수 있도록 저장한다. 또한, R-Hive를 사용하여 MapReduce 과정을 거쳐 사용자가 원하는 데이터를 조회하거나 분석할 수 있는 시스템을 구현하여 방대한 양의 로그를 처리함에 있어서 유연하고 처리 속도가 높은 효과적인 대용량의 로그 분석이 가능하도록 하였다.

Ⅱ. 관련 연구

2.1. 통합로그관리

통합로그관리시스템은 정보시스템에서 생성되는 이기종 디바이스에서 다양한 로그를 수집, 저장하여 필요한 정보를 검색하고 보고서를 생성하여 IT 인프라의 상태와 사용 현황을 알려주는 역할을 한다. 로그는 가장 기초적인 데이터로 생성부터 폐기에 이르는 로그의 생명주기를 관리하는 것으로 이를 통해 장애나 보안사고가 발생했을 시 원인을 추적하고, 다양한 규제 및 법규에 관한 감사 자료로 활용할 수 있도록 사후 활용의 성격이 강하다[2]. 로그는 실시간 대응력은 없지만, IT 인프라의 모든 부분에 대해서 발생하고 발자취를 남기기 때문에 정확하게 사고원인을 파악할 수 있는 장점이 있다. 로그 관리의 필요성과 가치에 대해 다시 중요시된 일은 최근에 일어났지만 상대적으로 수집한 로그 데이터들을 제대로 사용하지 못하고 있는 것으로 나타났다. 일례로 SANS가 2012년 60명 이상의 IT 전문가를 대상으로 한 설문조사 결과를 살펴보면, 22%의 응답자들은 보안 정보와 이벤트를 수집하는 시스템을 이용하여 데이터를 모아 분석하는 반면에, 58%는 로그관리 시스템을 이용하고, 나머지 응답자는 다른 방법에 의존하고 있는 것으로 나타났다. 대부분의 응답자들은 로그를 모으는 주요 이유로 규제를 준수하기 위함이라고 밝혔고, 거의 모든 응답자들은 의심스러운 행동을 감지하고 추적하는 행위는 매우 중요하다고 답했다. 하지만 정작 수집의 이유에 대한 답변으로 중요하다고 했지만 아이러니하게도 수집 후에도 실제 이용하기 가장 어려운 장애물로 조사됨에 따라 시간이 지날수록 방대한 데이터가 발생하게 되고 빅데이터 시대가 열리면서 로그 관리의 중요성과 활용이 확대되어 원본 로그 저장과 실시간 처리의 필요성이 동시에 중요하게 제기되고 있다[3].

ESM(Enterprise Security Management)의 경우에 보안 솔루션의 로그를 받아 상호 연관 관계를 파악하는 것이기 때문에 내부 사용자에 의한 위협에 한계가 있을 수 밖에 없으며, 이러한 약점은 보안 솔루션은 물론 IT 인프라의 전체 서버와 애플리케이션의 로그들을 취합, 저장, 분석하는 로그 관리를 통해 채워져야 한다는 것이다. 단순한 두 분야의 조합이 아니라 융합의 필요성이 나타나게 되어 SIEM(Security Information & Event Management)이 주목받게 되었다.

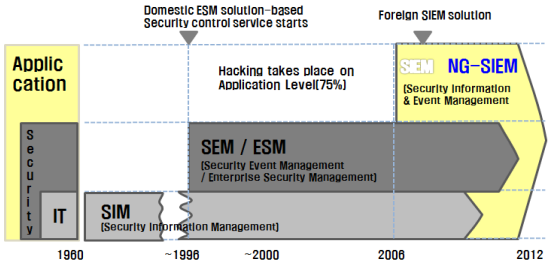

그림 1. 보안로그관리 발전 단계
Fig. 1 Evolution Step of Security Log Management

표 1. 보안로그관리 기능 분석
Table. 1 Features of Security Log Management Systems

	Use	Analysis Area
SIM	▪ ICT System Log Integration and Fault Monitoring	▪ ICT-oriented Integration and Analysis
ESM	▪ Security Control via ICT System and Security System Linkage Analysis	▪ ICT and Internal Security Area
SIEM	▪ General Analysis via ESM and Application Linkage	▪ Unknown Security Threat Analysis

2.2. Hadoop

하둡은 여러 개의 저렴한 컴퓨터를 마치 하나인 것처럼 묶어 대용량 데이터를 처리하는 기술로서, 하둡은 수천대의 분산된 x86 장비에 대용량 파일을 저장할 수 있는 기능을 제공하는 분산파일시스템(Distribute File System)과 저장된 파일 데이터를 분산된 서버의 CPU와 메모리 자원을 이용해 쉽고 빠르게 분석할 수 있는 컴퓨팅 플랫폼인 맵리듀스(MapReduce)로 구성되어 있다[4]. 하둡은 하나의 마스터와 다수의 슬레이브로 구성된 마스터/슬레이브 아키텍처를 갖고 있다.

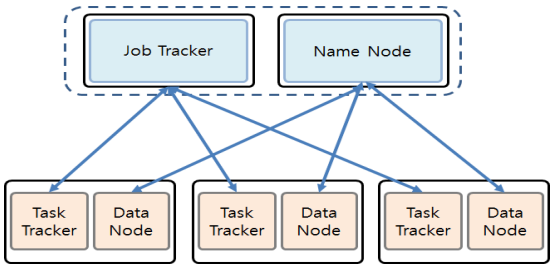

그림 2. 하둡 클러스터 구성도
Fig. 2 Configuration of Hadoop Cluster

하둡은 분산 데이터 저장 및 처리에 있어서 더욱 비즈니스에 효율적으로 적용시킬 수 있도록 다양한 서브 프로젝트를 제공한다. 이러한 서브 프로젝트들을 상용화시키면서 하둡 에코시스템이 구성되었으며, 하둡 생태계라고 표현되기도 한다.

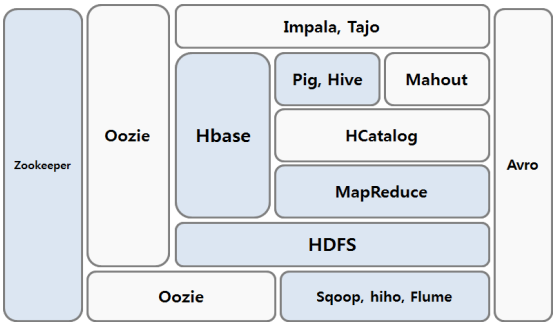

그림 3. 하둡에코시스템 구성도
Fig. 3 Configuration of Hadoop Ecosystem

Ⅲ. 하둡 기반의 보안로그분석 시스템

3.1. 시스템 구현

하둡 기반의 보안로그분석 시스템의 구성도는 그림 4와 같으며 이기종 디바이스에서 발생하는 로그를 하둡 기반의 NoSQL인 Hbase에 저장시키는 알고리즘을 통해 효과적인 로그 저장 기법과 저장된 대용량의 데이터를 R-Hive를 이용하여 분석 후 경고를 웹에 표시하는 과정으로 구성되어 있다.

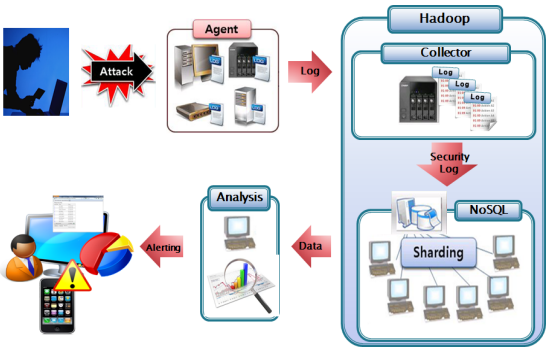

그림 4. 하둡 기반 보안로그분석 시스템 구성도
Fig. 4 Configuration of Hadoop-based Security Log Analysis System

본 시스템은 기본적으로 4가지의 요소로 구성된다. 첫 번째 요소는 로그를 발생시키는 에이전트에서 로그 수집기인 Flume을 사용하여 로그 수집 서버에 전송되도록 한다. 이때 Flume은 Agent와 Collector로 구조화되어 있고 로그 데이터를 전송받으면 HBase와 연동하여 HBase에 저장시키기 위한 전처리 과정으로 정규식 기법 사용한다. 두 번째 요소로 분산데이터베이스인 HBase의 구현을 통해 대용량의 데이터를 샤딩(Sharding) 기법을 이용하여 다수의 Region 서버에 저장할 수 있게 된다[5]. 세 번째 요소는 Hive를 이용하여 HBase의 데이터를 효과적으로 조회하기 위해 HBase와 연동된 가상테이블을 생성하여 SQL 질의어를 통해 분석 기반을 마련하고 맵리듀스를 통해 데이터를 분석한다. 마지막으로 분석된 데이터를 통해 RHive를 이용하여 시각적으로 로그 통계 리포트를 보여줄 수 있고 정책에 맞게 위협 탐지 룰셋(Rule-Set) 정보를 설정하여 위협 탐지에 관한 경고를 줄 수 있는 웹 인터페이스를 구현한다.

그림 5는 본 논문에서 제시한 하둡에코시스템에 대한 전체 아키텍처를 도식화한 것이다.

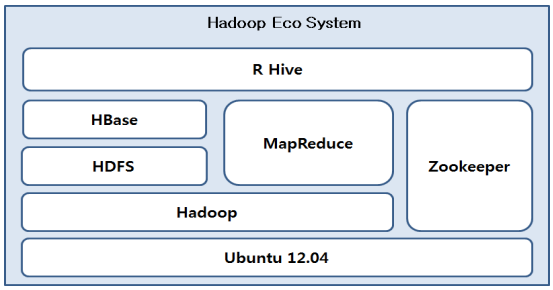

그림 5. 하둡에코시스템 아키텍쳐
Fig. 5 Architecture of Hadoop Ecosystem

3.2. Flume, HDFS 및 Hbase 연동

로그 수집기인 Flume, 하둡분산파일시스템인 HDFS, 그리고 NoSQL 분산데이터베이스 Hbase를 서로 연동시킴으로써 대용량의 로그 데이터를 분산저장 및 처리를 할 수 있는 기반이 마련된다[6].

Collector의 설정 파일의 샘플을 살펴보면 sinks의 타입을 hbase.HbaseSink로 설정하여 table명은 'snort', columnFamily는 'cf'라는 컬럼에 agent에서 로그 데이터를 받아와 Hbase에 저장되게 할 수가 있다. 로그마다 데이터 형식이 다르기 때문에 효율적으로 로그 수집기에 저장시키려면 전처리 과정을 거쳐야 각 컬럼 별로 로그 데이터의 정보를 체계적으로 저장할 수 있다[7, 8]. 이러한 전처리 과정은 Hbase에 저장할 때 전송된 로그 데이터를 정규식을 사용하여 각 컬럼에 저장할 수 있게 한다.

이기종 디바이스의 Agent에서 로그가 발생하여 Flume Agent를 통해 Flume Collector로 로그가 수집될 시, 실시간으로 HBase에 저장시키는 방법으로 구현하였다. HBase에 저장될 때 특성상 대용량의 데이터가 전송되었을 경우 HDFS로 저장함으로써 안정성을 도모한다. 그림 6은 Flume, HDFS, HBase의 연동 구조를 나타낸 것이다.

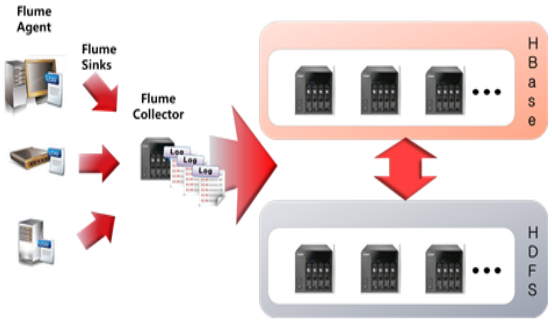

그림 6. Flume, HDFS 및 HBase 연동
Fig. 6 Interworking of Flume, HDFS, and HBase

3.3. 정규식을 이용한 전처리 과정

Agent에서 로그를 수집하는 Collector Server로 전송할 때 전처리 과정을 통해서 HBase의 확장형 칼럼에 저장해야 추후 분석을 효과적으로 할 수 있다. HBase는 NoSQL의 특성상 쿼리문을 사용하기가 힘든 단점이 있기 때문에 전처리 과정을 거쳐 HBase에서 각 칼럼별로 로그 데이터를 저장하고 Hive에서 가상 테이블을 생성하여 HBase와의 연동을 통해 쿼리문을 사용할 수 있게 구현하였다.

분석 단계에서 Hive를 활용하여 쿼리문을 사용하기 위해서는 HBase에 데이터를 저장시킬 때 비정형화된 로그를 분석하여 정형화된 로그 데이터를 만들기 위한 전처리가 필요하다. 이 과정에 사용하는 방법은 정규식을 이용하여 로그 데이터를 스플릿한 후 HBase에 칼럼별로 정형화된 로그 데이터를 저장하는 것이다.

로그 샘플을 살펴보면 날짜, 출발지 IP, 출발지 Port, 도착지 IP, 도착지 Port 등의 형식으로 되어 있는 것을 볼 수 있다. 이러한 로그를 기반으로 그림 7과 같은 정규식을 사용하여 로그를 분할할 수 있다.

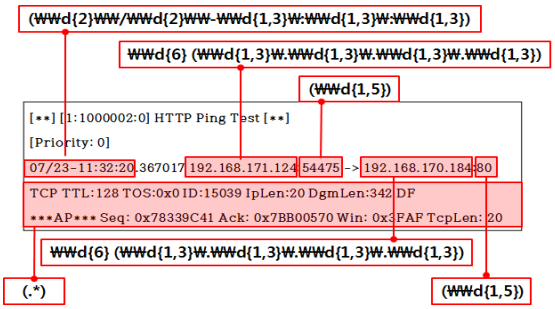

그림 7. 정규식을 사용한 전처리 방법
Fig. 7 Preprocessing of Normal Expression

3.4. 로그수집 관리시스템

사용자는 관리시스템을 이용하여 Hadoop 또는 HBase를 원격으로 구동시킬 수 있으며, 로그가 발생되는 Agent가 Collector Server로 로그를 전송하기 위한 설정과 Collector가 Agent로부터 로그를 전송받기 위한 절차를 자동 설정해준다. SSH 접속을 통해 노드에 명령어를 전송하여 해당 노드에서 명령어가 실행되도록 한다. 이를 통해 사용자는 관리의 용이함 및 시간 절감의 효과를 볼 수 있다.

하둡에코시스템에는 자동 실행 알고리즘과 로그 수집기 자동 설정 알고리즘이 포함되어 있다. 그림 8은 로그수집관리 시스템 화면을 보여준다.

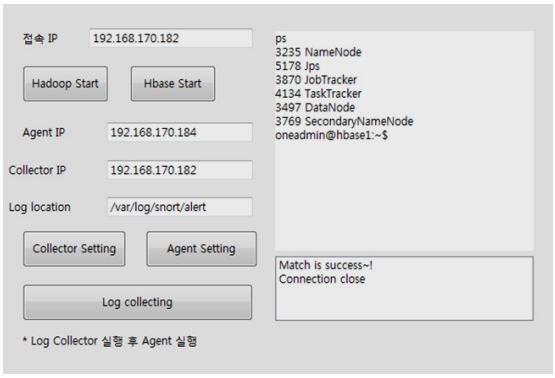

그림 8. 로그수집관리 시스템 화면
Fig. 8 Screenshot of Log Collection

하둡 또는 HBase의 동작 여부에 따라 하둡과 HBase의 실행을 자동화시켜주는 방법을 제안하였다. 프로그램에 IP를 입력하고 해당 노드에 접속하여 하둡에코시스템이 정상적으로 작동되고 있는지 판단하여 실행중이 아니면 하둡 에코시스템인 하둡과 HBase를 실행시킨다. 그림 9는 하둡에코시스템의 자동 실행 알고리즘에 대한 흐름도이다.

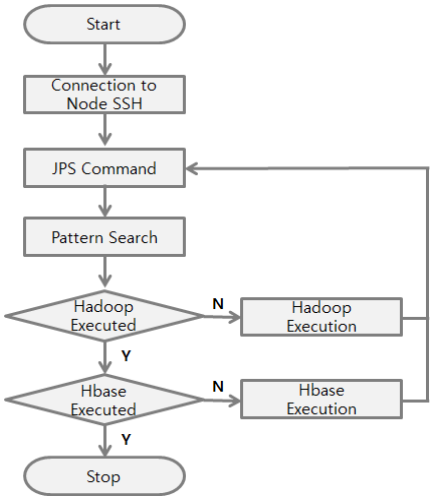

그림 9. 하둡에코시스템 실행 알고리즘
Fig. 9 Hadoop Ecosystem Execution Algorithm

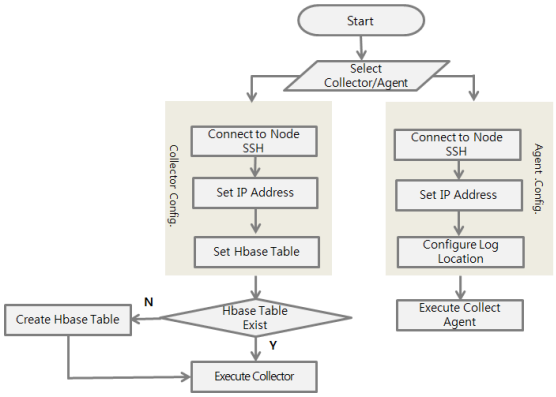

그림 10. 로그 수집기 설정 알고리즘
Fig. 10 Log Collector Configuration Algorithm

기존의 방법과 달리 본 연구에서는 로그가 발생하는 Agent와 로그를 수집하는 Collector의 로그를 수집하는 설정파일을 SSH 접속을 통해 자동으로 설정하는 알고

리즘으로써, 사용자가 프로그램을 통해 간편하게 로그를 수집할 수 있도록 한다.

로그를 수집할 Collector는 IP 설정과 HBase 테이블을 설정하면 HBase에 테이블이 존재하는지 판단할 수 있다. 판단 후 존재하지 않으면 테이블을 자동 생성하고 Collector Server를 실행하여 로그 수집을 시작한다. Agent는 Collector Server의 IP 설정과 수집할 로그 파일의 위치를 입력하면 Agent가 실행되어 Collector Server와 연결되어 로그가 수집되기 시작한다.

3.5. 분석시스템

HBase만을 사용하여 저장된 로그 데이터를 검색할 때는 SQL의 사용에 제한이 있기 때문에 Hive를 사용하여 하둡 데이터를 SQL을 이용한 쿼리를 이용하여 다룰 수 있어, 분석을 용이하게 만들고 HBase 테이블 기반으로 Hive에서 가상의 테이블을 생성하여 Hive 쿼리를 사용하여 HBase의 Table 내용 검색이 가능하다. 가상 테이블을 생성하면 Hive에 접속하여 HBase의 칼럼 별로 데이터를 조회할 수 있으며, 맵리듀스 작업으로 대용량의 데이터를 분석할 때 빠른 속도로 처리가 가능하게 된다.

Ⅳ. 실험 및 성능평가

4.1. 로그수집 관리시스템

성능 실험을 위한 대량의 보안 로그를 발생시키기 위해 공격자는 Ping of Death 공격을 이용하여 Agent에 지속적으로 패킷을 전송하고 Agent에서는 오픈 소스인 Snort를 이용하여 침입탐지시스템(IDS)를 구현하였다. 그림 11은 실험 환경 구성도이다.

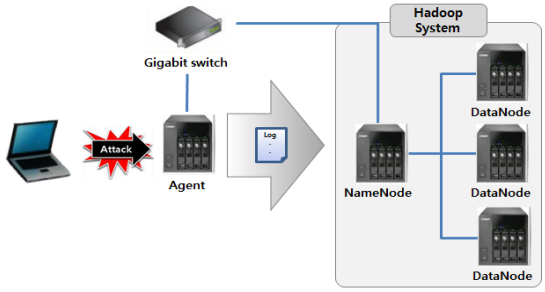

그림 11. 실험 환경 구성도
Fig. 11 Configuration of Experimental Environment

공격자가 65,000바이트 크기의 ICMP 패킷을 지속적으로 보내면 수신측 서버에서 네트워크 트래픽이 급증하는 것을 확인할 수 있으며, Wireshark로 패킷 덤프 분석을 한 결과 숫자 데이터 58이 반복되어 나오는 의미없는 과도한 데이터를 유발하여 Agent의 CPU 사용률을 높여 마비시키는 공격임을 확인할 수 있다.

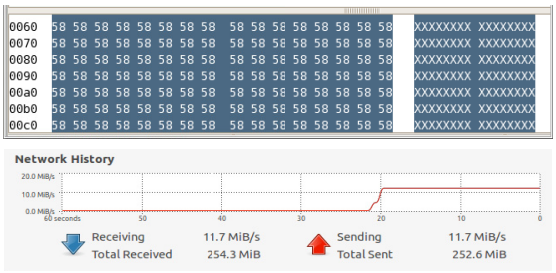

그림 12. 네트워크 공격 패킷
Fig. 12 Network Attacking Packet

로그 분석시스템에서는 RHive 명령어 rhive.query()를 이용하여 HBase 가상테이블을 생성한 Hive에서 대용량의 데이터를 분석할 수 있게 하였다.

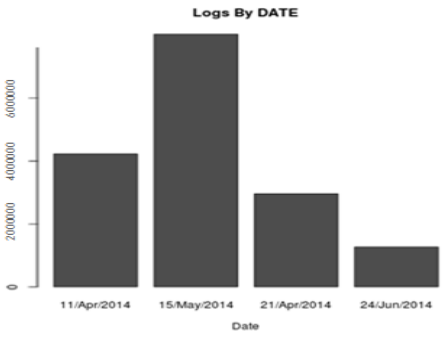

그림 13. 날짜별 로그 데이터
Fig. 13 Log Data by Date

로그 데이터의 내용 중에 시간 정보 가운데 날짜 부분만 추출하여 해당 날짜에 로그의 발생량을 알 수 있으며, 이를 이용하면 그림 13와 같은 그래프를 얻을 수 있다.

4.2. 성능 테스트

로그 수집 단계에서 전처리 과정을 통한 데이터 분석 성능을 평가하기 위해 데이터 양을 늘려가며 HBase에

전처리 과정의 실행 유무에 따른 테이블을 생성하였다. Flume에서 HBase로 로그를 저장시키는 단계에서 정규식을 이용하여 각 칼럼의 로그에서 의미있는 데이터를 추출하여 저장시키는 테이블과 Flume에서 HBase로 로그를 전처리 과정 없이 저장시키는 테이블로 데이터베이스를 구성하였다.

HBase에 저장된 로그에서 데이터를 조회하기 위해 쿼리문이 필요하기 때문에 Hive로 가상 테이블을 생성하여 SQL이 가능하도록 하였다. 전처리 적용 테이블과 미적용 테이블의 저장된 로그 데이터에서 같은 결과값을 얻기 위한 쿼리문을 사용하여 처리 속도를 비교하였다.

그림 14는 정규식을 적용한 테이블과 적용하지 않은 테이블의 Row의 수에 따른 쿼리 속도를 비교한 그래프이다. 테스트 결과를 확인해 보면 10만의 Row의 데이터를 처리할 때는 정규식을 적용한 테이블과 미적용한 테이블이 많은 차이가 나지 않았지만 Row의 양이 증가할수록 차이가 많이 나는 것을 확인할 수 있다. 따라서 Row의 수가 더욱 늘어날수록 격차는 더욱 커질 것으로 예상할 수 있다. 즉, 데이터의 양이 커질수록 정규식을 적용한 방식이 높은 격차로 빠른 속도를 낼 수 있다. 쿼리 속도가 빠를수록 분석 속도도 빨라지게 되기 때문에 대용량의 데이터를 분석함에 있어서 정규식을 적용한 분석법이 소요 시간을 단축시키는데 효과가 있음을 확인할 수 있다.

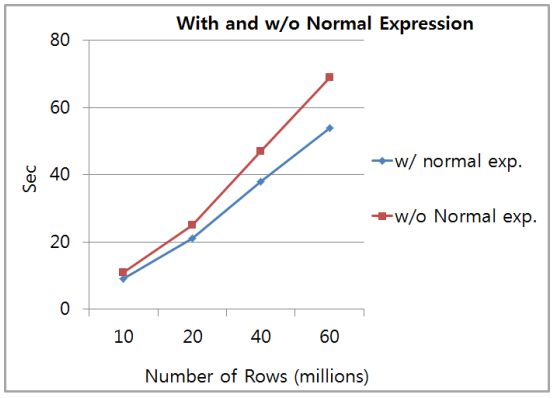

그림 14. 정규식 적용 VS 미적용 성능 비교
Fig. 14 Performance with Normal Expression vs without Normal Expression

V. 결 론

본 논문에서는 하둡 기반의 효율적인 보안 로그분석 시스템을 설계하고 이를 오픈 소스를 활용하여 구현하였다. 기존의 NoSQL 기반의 로그 분석시스템에서는 로그 수집과 저장하는 방식에서 추후 분석을 위해 의미있는 데이터를 추출하는 과정이 필요하여 분석 시간이 많이 소요되는 문제점이 있었다. 본 논문에서는 이러한 문제를 해결하기 위해 자동화된 로그 수집 과정과 정규식을 이용하여 전처리 과정을 거쳐 로그 분석 시간을 단축할 수 있는 효율적인 시스템을 구현하였다. 이를 위해 대용량의 로그 데이터를 수용하고 처리하기 위해 오픈소스인 하둡 프레임워크를 이용하여 사용자가 로그 분석 보안 정책에 따라 악의적 공격을 분석할 수 있는 기존 시스템보다 처리율, 응답시간, 데이터 확장성에서 우수한 성능을 제공함을 확인하였다.

ACKNOWLEDGMENTS

This research was financially supported in part by the Ministry of Science, ICT, and Future Planning and IITP through the Innovative Human Resource Training Project for Regional Innovation. Also, this research was financially supported in part by Korea Sanhak Foundation.

REFERENCES

[1] W. J. Kim and H. Y Yeum “Integrated Management and IT Compliance for Heterogeneous Log”, *Journal of Korea Institute of Information Security & Cryptology*, Vol.20, No.5, pp.73-86, 2010.10.

[2] H. W. Lee “Design and Implementation of Web Attack Detection Based on Integrated Web Audit Data”, *KSII Transactions on Internet and Information Systems*, Vol.11, No.6, pp.73-86, 2010.12.

[3] D. H. Kim, “SIEM Trend Evolving into Intelligent Log Management Platform in Bigdata Environment”, *NIPA, ITFIND*, 2013. 8.

[4] B. M. Choi, J. H. Gong, S. S. Hong, and M. M. Han, “The Method of Analyzing Firewall Log Data using MapReduce based on NoSQL”, *Journal of Korea Institute of Information Security & Cryptology*, Vol.23, No.4, pp. 667-677, 2013.
[5] M. J. Kim, S. H, Han, W. Choi, and H. G. Lee, “Design and Implementation of MongoDB-based Unstructured Log Processing System over Cloud Computing Environment”, *KSII Transactions on Internet and Information Systems*, Vol.14, No.6, pp.71-84, 2013.12.
[6] D. S. Choi, J. J. Moon, Y. M. Kim, and B. N. Noh, “An Analysis of Large-Scale Security Log using MapReduce”, *Journal of KIIT*, Vol.9, No.8, pp. 125-132, 2011.8.
[7] Fengying Yang, "Research on Cloud-Based Mass Log Data Management Mechanism", *Journal of Computers*, Vol. 9, No. 6, June 2014.
[8] H. J. Jeong, “Integration of Large-scale Security Log based on NoSQL in Cloud Computing Environment”, Chosun University Master’s Thesis, 2014.

안광민(Kwang-Min Ahn)

2002년 대전대학교 정보통신공학과 졸업(학사)
2004년 대전대학교 정보통신공학과 졸업(석사)
2014년 대전대학교 대학원 박사과정 수료
※ 관심분야 : 클라우드 컴퓨팅, 포그 컴퓨팅, 보안

이종윤(Yoon-Jong Lee)

2013년 대전대학교 정보통신공학과 졸업(학사)
2015년 대전대학교 정보통신공학과 졸업(석사)
※ 관심분야 : 클라우드 컴퓨팅, 보안, 네트워크

양동민(Dong-Min Yang)

2000년 POSTECH 컴퓨터공학과(공학사)
2003년 POSTECH 컴퓨터공학과 졸업(석사)
2011년 POSTECH 컴퓨터공학과 졸업(박사)
현재 대전대학교 정보통신공학과 교수
※관심분야 : 무선이동통신, 인지 라디오 네트워크, 무선 센서 네트워크, 이동 애드혹 네트워크 등

이봉환(Bong-Hwan Lee)

1985년 서강대학교 전자공학과 졸업(학사)
1987년 연세대학교 대학원 전자 공학과 졸업(석사)
1993년 Texas A&M 대학교 대학원 전기 및 컴퓨터공학과 졸업(박사)
현재 대전대학교 정보통신공학과 교수
※ 관심분야 : 클라우드 컴퓨팅, 포그 컴퓨팅, 네트워크보안 등