Hadoop Ecosystem 기반 대용량 보안로그 수집 시스템 설계 및 구축

이종윤, 이봉환,
대전대학교 정보통신공학과
e-mail : heyyaki@naver.com

Design and implementation of a Large-Scale Security Log Collection System based on Hadoop Ecosystem

Jong-Yoon Lee and Bong-Hwan Lee
Dept. of Information and Communications Engineering,
Daejeon University

요 약

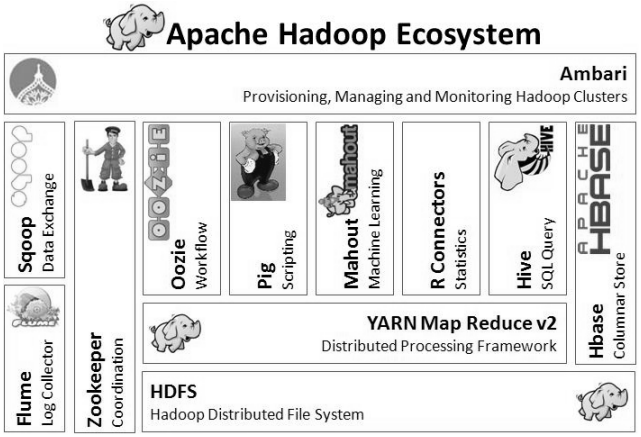
네트워크 공격이 다양해지고 빈번하게 발생함에 따라 이에 따라 해킹 공격의 유형을 파악하기 위해 다양한 보안 솔루션이 생겨났다. 그 중 하나인 통합보안관리시스템은 다양한 로그 관리와 분석을 통해 보안 정책을 세워 차후에 있을 공격에 대비할 수 있지만 기존 통합보안관리시스템은 대부분 관계형 데이터베이스의 사용으로 급격히 증가하는 데이터를 감당하지 못한다. 많은 정보를 가지는 로그데이터의 유실 방지 및 시스템 저하를 막기 위해 대용량의 로그 데이터를 처리하는 방식이 필요해짐에 따라 분산처리에 특화되어 있는 하둡 에코시스템을 이용하여 늘어나는 데이터에 따라 유연하게 대처할 수 있고 기존 NoSQL 로그 저장방식에서 나아가 로그 저장단계에서 정규화를 사용하여 처리, 저장 능력을 향상시켜 실시간 처리 및 저장, 확장성이 뛰어난 하둡 기반의 로그 수집 시스템을 제안하고자 한다.

1. 서론

최근 네트워크의 급속한 발전과 더불어 다양한 방법의 네트워크 및 어플리케이션 공격이 이루어지고 있다. 이러한 공격에 대비하기 위하여 IDS (Intrusion Detection System, 침입탐지시스템), IPS(Intrusion Prevention System, 침입탐지시스템), Firewall (방화벽) 등의 보안 시스템에서는 대용량의 보안 로그와 Server Agent에서의 각종 로그가 발생하고 있다. 이러한 대용량의 로그와 이벤트를 분석하여 공격에 대한 통제와 대비책, 보안 정책을 마련할 수 있도록 하기 위해 통합보안관리(ESM, : Enterprise Security Management) 시스템이 등장하였다.

시간이 지날수록 데이터의 양은 점점 더 증가하고 있기 때문에, 기존의 통합보안관리시스템은 대부분 관계형 데이터베이스를 이용하기에 이러한 대용량의 데이터를 저장하기에는 처리능력, 수용능력을 벗어나 결국은 성능 저하나 주요 정보 유실 등의 문제의 원인이 된다. 이에 따라 정보의 수용, 처리 능력 향상 및 데이터 유실없이 로그를 수집하는 필요성이 절실하게 된다. 이러한 필요성을 만족하기 위해 분산처리 환경을 요구하게 되는데 이에 적합한 프레임워크로 하둡을 고려할 수 있다[1].

하둡은 아파치 오픈소스 프로젝트로 구글 파일시스템(Google File System)을 벤치마킹하여 하둡분산파일시스템(HDFS:Hadoop Distributed File System)과 맵리듀스(MapReduce)를 구현한 것으로, 사용하기 쉽고 편리하다는 장점을 내세워 널리 알려지게 되었다. 하둡은 대용량의 데이터를 분산저장과 다수의 서버 클러스터에서 일어나는 병렬처리 작업을 하는데에 최적화되어 있다. 하둡의 오픈소스 특성상 다양한 연관 솔루션과 도구들이 등장하는데 이러한 연관 관계를 하둡 생태계(Hadoop Ecosystem)라고 부른다. 하둡 생태계를 이용하여 다양하게 활용할 수 있는 분석 플랫폼을 구축할 수 있다.



(그림 1) Hadoop Ecosystem

그림 1과 같이 하둡 생태계를 활용하여 많은 양의 로그 데이터를 수집하는데 필요한 분산처리환경을 기반으로 실시간 처리 및 유연한 확장성을 목적으로 하는 로그 수집 시스템의 설계를 제안하고, 향후 연구에 필요한 사항을 확인하도록 한다.

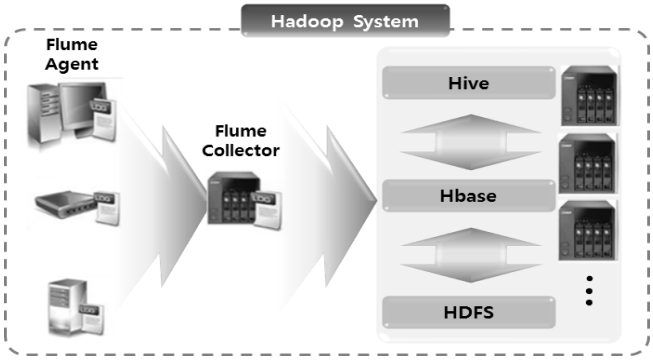
2. 대용량 보안로그 수집시스템 설계 및 구현

2.1 로그 수집 프레임워크

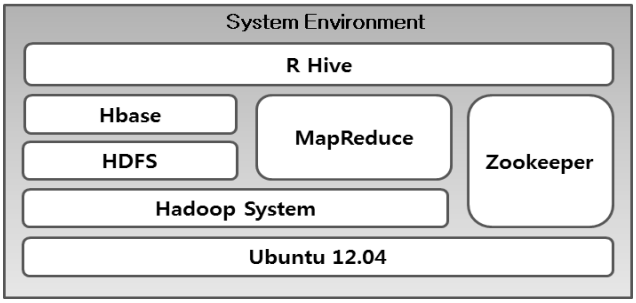
로그를 수집하기 위해서는 이기종 디바이스에 탑재 가능한 로그전송 에이전트와 수집기의 기능을 하는 Flume, 하둡분산파일시스템으로 대용량의 데이터를 저장할 수 있는 Hadoop System, NoSQL DBMS의 하나인 HBase, 사용자가 SQL 베이스의 쿼리를 통해 데이터 분석과 요약을 쉽게 할 수 있도록 하는 클라이언트 툴인 Hive를 사용하여 하둡 기반의 로그 수집 프레임워크를 구축한다.

2.2 로그 수집 시스템 설계

제안하는 하둡 기반의 로그 수집 시스템은 이기종의 디바이스에서 로그를 실시간으로 Collector로 전송할 수 있는 역할이 중요하다. 따라서 다양한 로그 수집 툴이 있지만 안정된 성능을 보이는 Flume을 활용하여 Collector로 실시간 전송하고 즉시 HBase에 저장하고 Hive를 이용하여 NoSQL DBMS에 저장되어 있는 데이터의 분석을 용이하게 만들기 위해 HBase와 연동하는 시스템으로 설계하였다. 하둡 기반 로그 수집 시스템의 구성도는 그림 2와 같다.

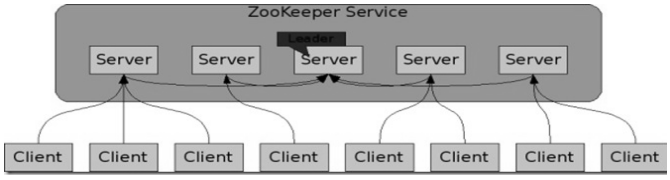


(그림 2) 제안하는 로그 수집 시스템 구성도



(그림 3) 제안하는 로그 수집 시스템 아키텍쳐

하둡 에코시스템을 사용하면서 안정성을 위해 Zookeeper도 같이 사용하게 된다[2]. Zookeeper는 다중의 서버 집합을 묶어서 관리해주는 시스템으로 서버의 수가 늘어나며 한 대 이상의 서버에 이상 발생이 생길 확률이 커지므로 데이터의 흐름 안정화에 큰 역할을 할 수 있다. 리더라고 불리는 서버는 모든 서버의 중심이 되는 곳이며 또한 하나의 서버에서 처리가 되어 데이터가 변경되면 모든 서버에 전달되어 동기화를 하게 된다. 그림 4는 Zookeeper의 구조를 나타낸 것이다.



(그림 4) Zookeeper 구조

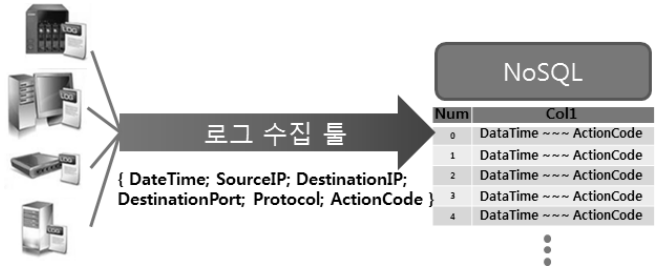
대용량의 데이터를 분산 저장할 수 있는 NoSQL DBMS 구축을 위해 HBase를 사용하는데 이는 구글의 Bigtable과 매우 유사한 Data mode를 사용한다. 그래서 데이터 구조는 Bigtable의 구조를 사용하며, data row는 정렬 가능한 row key와 column들의 임의의 숫자를 가진다. 형태는 <Family>:<label>로 Family는 column을 유동적으로 늘리어 정보가 각기 다른 로그 데이터를 유연하게 저장시키는데 목적이 있다[3].

이기종 장비에서 보안 로그 데이터는 각기 다른 형태로 정보가 저장될 것이다. 이에 따라 Flume에서 HBase로 데이터를 저장시키는데 상이한 데이터로 인해 저장 시에 문제가 발생하면 안되기 때문에 해결 방안으로 Agent에서 Collector로 Sink 설정할 시 Hbase serializer를 Regex HbaseEventSerializer API를 활용하여 문제 해결을 한다. 위와 같은 API를 사용하면 정규식을 통해 로그 데이터의 형식이 각기 다르더라도 HBase에 저장시키고자하는 모든 로그 데이터를 유실없이 저장시킬 수 있다.

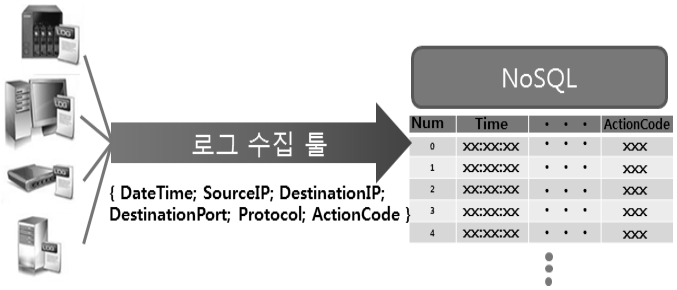
2.2 로그 수집 시스템 구축

제안하는 로그 수집 시스템 구축에 사용된 노드는 총 4대로 메인노드 1대와 확장노드 2대, 그리고 로그 발생 서버로 구성하였다. 각 노드의 성능은 CPU Intel i7, Memory 12G, HDD 1T을 갖추고 있으며, 노드 구축 환경은 Ubuntu 12.04 운영체제에 Flume-1.4.0, Hadoop-1.2.1, Hbase-0.94.12, Rhive-0.0.7을 사용하여 구축하였다.

Flume에서 정규식을 활용하여 Hbase로 로그 데이터를 즉시 컬럼별로 수집하게 만들어 데이터 분석하는데 더 빠르고 용이하게 고안하였다. 기존의 NoSQL을 사용하는 로그 데이터 수집은 온전한 로그를 레코드에 쌓아두는 형식이었지만 제안하는 시스템의 로그 데이터 수집은 Flume에서 Hbase로 저장이 되는 과정에서 정규화를 사용해서 Column 별로 원하는 데이터를 가공하여 저장할 수 있기 때문에 차후 데이터 분석 단계에 용이하게 사용할 수 있다.



(그림 4) 기존 NoSQL 로그 수집 시스템 저장 방식



(그림 5) 정규식 활용한 로그 수집 시스템 저장 방식

정규식을 사용한 Flume의 로그 수집 설정 과정에 해당 소스가 사용된다.

```
public void configure(Context context){
 String regex = context.getString(REGEX_CONFIG,
REGEX_DEFAULT);
 regexIgnoreCase=context.getBoolean(IGNORE_CASE_CONFIG,
INGORE_CASE_DEFAULT);
 inputPattern = Pattern.compile(regex, Pattern.DOTALL
+ (regexIgnoreCase ? Pattern.CASE_INSENSITIVE : 0));
 String colNameStr = context.getString(COL_NAME_ COLUMN_
NAME_CONFIG, COLUMN_NAME_DEFAULT);
 String[] columnNames = colNameStr.split(",");
 for (String s: columnNames){
  colNames.add()s.getBytes(Charsets.UTF_8));
 }}
```

위 소스를 적용시켜 설정을 하면 정규식을 활용하여 Hbase의 각 Column별로 데이터를 가공하여 저장시킬 수 있게 된다. 이를 활용하여 분석, 통계 처리를 할 때 기존의 로그 수집 방법보다 데이터를 가져오는데 더욱 용이하게 사용될 수 있다.

3. 결론 및 향후연구

본 논문에서는 기존의 시스템에서 성능한계를 벗어나지 못하여 로그 데이터가 유실되거나 성능 저하를 일으키는 것을 해결하기 위해, 하둡의 분산처리, 저장기능을 활용하여 대용량의 데이터를 저장시키는 하둡 기반의 로그 수집 시스템을 설계 및 구축을 하였다. 또한 기존 로그 수집 시스템에서 분석하는 것 보다 분석 단계를 수월하게 진행시키기 위해 로그 수집단계에서 데이터를 가공하여 Column별로 저장시키는 방안을 제안하였다.

향후 연구로는 설계 및 구축한 바탕으로 데이터의 로그 데이터 분석·통계 처리 과정을 거쳐 시각화에 관한 방법론을 연구해 볼 예정이며, 본 논문에서 제시한 시스템과 기존의 로그 수집 시스템을 사용하여 대용량의 로그 데이터를 저장해보고 분석 단계까지의 비교평가를 통해 제시한 시스템의 성능과 활용 가능성을 입증하고자 한다.

Acknowledgement

본 논문은 미래창조과학부의 고용계약형 SW석사과정 지원사업으로 수행한 결과임.

참고문헌

[1] 한국인터넷진흥원, “2013년 인터넷 및 정보보호 10대 이슈 전망”

[2] 김형준, 조준호, 안성화, 김병준, “클라우드 컴퓨팅 구현 기술”, 에이콘, 2010.

[3] 최대수, 문길종, 김용민, 노봉남, “MapReduce를 이용한 대용량 보안로그 분석”, 한국정보기술학회 논문지, 제 9권 제 8호, 2011년 08월

[4] Karen kent, Murugiah Souppaya, “Guide to Computer Security Log Management”, NIST SP 800-92, pp. ES-1~ES-3, 2006

[5] 최보민, 공종환, 홍성삼, 한명묵, “NoSQL기반의 MapReduce를 이용한방화벽 로그 분석 기법”, Journal of The Korea Institute of Information Security & Cryptology(JKIISC), VOL.23, NO.4, August 2013