Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.4, August 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.4.667

NoSQL기반의 MapReduce를 이용한 방화벽 로그 분석 기법*

최 보 민,† 공 종 환, 홍 성 삼, 한 명 묵‡
가천대학교

The Method of Analyzing Firewall Log Data using MapReduce based on NoSQL*

Bomin Choi,† Jong-Hwan Kong, Sung-Sam Hong, Myung-Mook Han‡
Gachon University

요 약

방화벽은 대표적인 네트워크 보안 장비로서 대부분의 네트워크 내/외부에 설치되어 패킷의 입/출입을 통제하는 시스템이다. 때문에 이에 저장된 로그 데이터를 분석하는 것은 네트워크 보안연구에 중요한 기초 자료를 제공해 줄 수 있다. 그런데 최근 기술의 발달로 인터넷망의 속도가 향상되고, 네트워크가 대형화 되면서 방화벽에서 저장하는 로그 데이터들의 양도 대용량화 또는 빅데이터화 되어 가고 있다. 이러한 추세 속에서, 기존의 전통적인 RDBMS방식으로 로그데이터를 분석하는 데는 한계가 있다. 본 논문은 NoSQL 기반의 MapReduce 설계를 이용한 방화벽 로그 분석 기법을 통해 NoSQL방식을 도입하는 것이 대용량 로그 데이터를 더욱 효율적으로 분석할 수 있다는 점을 발견했다. 우리는 기존의 RDBMS방식과의 데이터 처리 성능을 비교하여 NoSQL방식 데이터베이스의 우수한 성능을 입증하였고, 이를 바탕으로 제안하는 설계 기법을 평가하기 위해 3가지 공격 패턴을 선정하고 이를 집계 하여 분석을 수행하는 실험을 통해 제안하는 분석 기법의 성능 및 정확성을 입증하였다.

ABSTRACT

As the firewall is a typical network security equipment, it is usually installed at most of internal/external networks and makes many packet data in/out. So analyzing the its logs stored in it can provide important and fundamental data on the network security research. However, along with development of communications technology, the speed of internet network is improved and then the amount of log data is becoming 'Massive Data' or 'BigData'. In this trend, there are limits to analyze log data using the traditional database model RDBMS. In this paper, through our Method of Analyzing Firewall log data using MapReduce based on NoSQL, we have discovered that the introducing NoSQL data base model can more effectively analyze the massive log data than the traditional one. We have demonstrated execellent performance of the NoSQL by comparing the performance of data processing with existing RDBMS. Also the proposed method is evaluated by experiments that detect the three attack patterns and shown that it is highly effective.

Keywords: NoSQL, Firewall, Log Analysis, MapReduce, BigData

접수일(2013년 3월 29일), 수정일(1차: 2013년 6월 13일, 2차: 2013년 7월 1일), 게재확정일(2013년 7월 1일)
* "이 논문은 2013년도 가천대학교 교내연구비 지원에 의한 결과임."(GCU-2013-R193)
† 주저자, cbm0728@gmail.com
‡ 교신저자, mmhan@gachon.ac.kr(Corresponding author)

I. 서 론

현대 사회는 과학기술 발전과 함께 예측 불가능한 위험이 증가하는 정보 위험사회(Information Risk Society)가 되고 있다[1]. 일례로 2013년 3월 20일에 발생한 해킹 사태에서 볼 수 있듯이 네트워크를 통한 정보의 위협은 단순히 기술적 측면의 문제를 넘어서 사회적 측면의 문제로 이슈화 되고 있다. 이에 대응하기 위해 많은 보안 전문가들은 침입 탐지 및 예방 기법에 대해 연구하고 있으며, 다양한 물리적 장비들 또한 개발되고 있다.

네트워크 보안 위협에 대한 대표적인 보안 솔루션 중 하나인 방화벽은 네트워크의 내/외부의 경계에 설치되는 보안 장비로서 네트워크 구성에 필수적인 요소 중 하나이다. 과거에는 대규모 네트워크에만 제한적으로 사용 됐으나, 현재에는 인터넷 뱅킹 또는 온라인 게임 등과 같은 개인 정보 보호가 필요한 분야가 증가함에 따라 개인용 컴퓨터에도 필수적으로 설치되고 있다[2]. 방화벽을 통과하는 다양한 데이터들의 패킷들은 로그파일 형식으로 기록되는데, 일반적으로 이러한 로그파일은 공격한 해커가 있는 진원지, 네트워크 대역폭, 네트워크 부하량 등 다양한 보안 정보를 담고 있다. 따라서 방화벽 로그 분석은 공격자의 침입을 탐지하고 이를 예방할 수 있는 기술 등에 유용한 정보를 제공해 줄 수 있다[3].

하지만 최근 네트워크 기술의 발달로 인한 전송망의 속도 향상과 함께 방화벽이 처리하는 패킷의 수도 급격히 증가하게 되어 이를 기록한 방화벽 로그의 양 또한 기하급수적으로 늘어나고 있다. 이렇게 방대해진 보안 로그 데이터는 이를 효율적으로 관리하고 처리해 줄 데이터베이스가 필요한데, 최근 빅데이터의 이슈와 함께 등장한 대용량 데이터 처리에 유용한 NoSQL(Not Only SQL) 기술은 이에 대한 새로운 솔루션이 될 수 있다.

본 논문에서는 점차 대용량화 되어가고 있는 방화벽 로그 데이터의 특성을 분석하는 데 NoSQL 기반의 MapReduce 설계 방식을 적용했다. 빅데이터에 대한 유연하고 빠른 처리속도를 갖는 특징을 바탕으로, 기존 RDBMS(Relation Data Base Management System) 기반의 데이터베이스 모델 보다 빠르고 유연하게 대용량 로그 데이터를 처리 및 분석할 수 있는 방법을 제안함과 동시에, 3가지 DoS(Denial of Service)/DDoS(Distributed Denial of Service) 공격 탐지 실험을 통해 이에 대한 가능성을 입증하였다.

2장에서는 관련연구로서 기존 연구들의 취약점과제안하는 기법에 대한 전반적인 이론적 배경들을 소개하고, 3장에서는 NoSQL기반의 MapReduce 설계 기법을 제안하였다. 4장에서는 기존의 RDBMS와 NoSQL 기반의 MongoDB의 성능을 비교하고, 이를 바탕으로 실제 방화벽 보안 로그의 분석을 통해 공격 유무를 판단한다, 마지막 5장에서는 이에 대한 결과를 도출하고, 향후 연구 방향을 제시한다.

II. 관련 연구

2.1 방화벽 로그 분석의 필요성

방화벽 로그는 다양한 보안 정보를 담고 있기 때문에 이를 분석하는 것은 네트워크 보안사고 발생의 원인들에 대하여 추적할 증거자료를 제공해 주거나, 사고 발생 전 이상증후 혹은 외부의 해킹 시도를 탐지할 수 있다[4]. 방화벽에서 수집된 로그들은 다양한 정보들을 담고 있지만, 각각의 침입 탐지 유형에 따라 그 척도가 되는 정보가 다르다. [표 1]은 한국 전자통신연구원에서 개발한 상황인식 기반의 침입탐지 방법인 NASA(Network Attacks Situation Analy-

[표 1] 이용하는 로그 데이터 형식

로그 필드 타입	내용	
Date Time	로그발생 날짜/시간	
	YYYYMMDD HHMMSS 포맷	
Source IP	로그발생 진원지의 IP 주소	
	2 * 2byte unsigned String	
Destination IP	로그발생 목적지의 IP 주소	
	2 * 4byte unsigned String	
Destination Port	로그발생 목적지의 Port	
	2 * 2byte unsigned String	
Protocol	프로토콜	
	2 * 2byte unsigned String	
	- IP	: 0
	- ICMP	: 1
	- GGP	: 3
	- TCP	: 6
	- EGP	: 8
	- PUP	: 12
	- UDP	: 17
Action Code	로그 메시지의 상세 정보	
	최대 256 Byte의 문자열	
	- 1(허용)	: Allow/Accept/Close
	- 2(거부)	: Drop/Reject

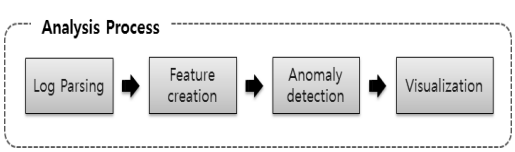
sis)[5]에 기반 하여 DoS나 DDoS 등과 같은 침입을 탐지하는데 주요 척도가 되고 있는 로그 데이터들로 정리한 표로, 4장의 실험에 이용될 로그 데이터들이 이에 해당한다.

이러한 로그 데이터들을 기반으로 특정 Source IP의 비정상적인 발생 횟수, Protocol 및 Action Code의 이벤트 발생 속성이나 반복 횟수에 따라 공격에 대한 패턴을 생성해 낼 수 있으며, 이를 통해 설정된 룰을 기반으로 DDoS, DoS, Port Scanning 등과 같은 다양한 네트워크 공격을 분류하고 탐지 할 수 있다[6].

2.2 전통적 RDBMS 기반 로그 분석의 취약점

일반적으로 로그 분석은 [그림 1]과 같은 단계를 거쳐 이루어지고 있으며 각 단계에 대한 역할은 다음과 같다[11]. 첫 번째 처리 단계인 Log Parsing 단계 에서는 보안 장비로부터 흘러들어오는 로그들을 수집하거나 분석 대상인 로그 파일 또는 로그 데이터의 선택을 수행한다. 분석할 대상이 수집 또는 선택 된 후에는 이를 이용하여 상관 분석이나 각 시스템 마다 적용되는 분석 기법들을 이용하여 특정 공격 유형의 반복적인 특징 및 패턴 등을 추출하여 이를 규칙 등으로 설정하는 것을 수행해야 하는데, 이러한 과정들을 처리 하는 것이 Feature creation 단계이다. 다음으로, 이전 단계에서 설정된 공격 규칙을 기반으로 Anomaly detection 단계가 수행된다. 본 단계에서는 특정 시간에 일어난 공격을 탐지하고 이에 대한집계 량이나 유형 등을 추출하여 추후 이와 같은 보안사고의 발생을 방지 하거나 예방할 수 있는 방안이 수립될 수 있다. 마지막 단계인 Visualization 단계는 시각화 하는 단계로써 악의적인 로그의 패턴을 모니터링 하거나 분석 결과를 판단하기에 용이하도록 정보를 가시적으로 표현해 주는 역할을 수행한다.

이와 같은 기존의 로그 분석 단계들은 일반적으로 관계형 데이터베이스인 RDBMS를 기반으로 처리되어 왔다. 그러나 기존의 방식은 빠르게 변화하고 거대화 되어 가고 있는 보안 로그 데이터들의 현 특성을



[그림 1] 로그 분석 단계

반영하지 못하고 있어 이를 분석하는 데 다음과 같은 한계들이 있다[12].

먼저 성능 문제이다. 기존의 RDBMS는 자체적인 Scale-Out 방식의 확장이 어려워 저장 공간 부족 현상이 일어날 수 있다. 이는 점차 늘어나는 많은 양의 방화벽 로그 데이터들을 수집하고 처리하는 데 있어 속도 저하 및 성능적인 결함을 발생시킬 수 있다. 이러한 문제들을 해결하기 위해 상용 데이터 솔루션을 통해 데이터 저장 공간을 확장하거나, 바이너리 형태로 저장하여 파일을 경량화 시켜 이를 저장 공간을 확보하려는 시도가 이루어져 왔다[12][13]. 그러나 현재 방화벽에서 서비스 중인 대용량의 데이터들은 고가의 솔루션에 저장할 수준의 의미 있는 데이터가 아닐 뿐만 아니라, 바이너리 형태의 파일 정장 방안은 별도의 파일 변환 과정이 필요하기 때문에 비실시간 로그 분석이라는 또 다른 취약점이 생기기 때문에 제시되고 있는 방안들은 현실적인 해결 방안이 되지 못하고 있다.

이 뿐만 아니라, 각 로그 데이터 테이블간의 복잡한 관계를 설정하여 이를 분석하고 있는 기존의 관계형 설계 방식은 변화하고 있는 공격의 패턴이나 새로운 공격 유형을 반영하는데 어려움이 있다. 즉, 변화된 공격 패턴이나 신규 공격의 발생 시에 이를 수정하거나 추가하기 위하여 매번 관계를 재설정해야 하고, 이러한 과정에서 점차 다양해지고 있는 공격의 모든 특징을 반영하고자 그 설계가 매우 복잡하고 거대해져 시스템의 효율성을 떨어뜨리는 원인이 될 수 있다.

이에 로그 분석에 있어서 기존 RDBMS가 가지는 한계점을 보완 및 극복할 수 있는 충분한 데이터 저장소와 구조의 유연성을 지닌 새로운 패러다임의 기술이 요구된다.

2.3 NoSQL 방식의 방화벽 로그 처리

본 논문에서는 2.2절에서 언급한 RDBMS 성능의 한계점을 보완하고 보다 효율적인 로그 분석을 구현하고자 NoSQL 방식에 기반한 로그 분석 방안을 제시한다. NoSQL은 기존 RDBMS의 비용이 높고 확장성이 떨어지는 문제점을 해결하기 위해 주목받기 시작한 기술 이다[14]. [표 2]는 이러한 NoSQL의 특징을 나열한 것으로, 일반적으로 이와 같은 속성을 지닌 데이터 저장소를 NoSQL 범주로 분류한다[6].

이러한 NoSQL이 특징들 중 로그 분석에 있어서 주목해 볼 점은 데이터 모델링을 위한 고정된 데이터

〔표 2〕 NoSQL의 특징

i. 관계형 데이터 모델이 아닌 키-값 또는 키-값을 응용한 데이터 모델
ii. 안정적이고 고가의 하드웨어가 아닌 다수의 값싼 하드웨어 이용
iii. 데이터는 분산된 노드에 파티션, 복제돼 저장
iv. 데이터의 정합성에 대한 요구 사항보다는 단절내성에 대한 요구사항이 더 중요
v. 2단계 커밋(commit) 수준의 트랜잭션 보다는 정족수(Quorum) 기반의 트랜잭션을 선호

스키마 없이 키(KEY) 값을 이용해 다양한 형태의 데이터에 접근이 가능하다는 것이다. 이러한 특성은 다양한 이기종 보안 로그들을 보다 용이한 방법으로 병합하여 Parsing 단계를 구현할 수 있으며, 데이터 모델링을 위한 고정된 데이터 스키마가 없어(schema-less) 새로운 공격 유형을 보다 신속하게 반영할 수 있다는 이점이 있다.

NoSQL은 데이터 모델에 따라 다양한 솔루션이 존재하고, 각 솔루션에 따라 사용 용도가 모두 다르므로 각 제품의 특징을 정확히 파악하여 활용 목적에 적합한 솔루션을 선택할 수 있도록 해야 한다. 일반적으로 NoSQL 솔루션의 분류는 데이터 모델 유형을 기준으로 Key-Value, Column, Document, Graph형식에 따라 분류되며, Hadoop의 HBase, Apache의 Cassandra, 10gen의 MongoDB가 대표적이다.

본 연구에서는 다양한 NoSQL 솔루션 모델 중에서도 차단 정책의 유동성을 보다 유연하게 반영할 수 있고, 유입되는 방화벽 로그 데이터를 보다 빠르게 수집 및 처리해야 하는 점을 고려하여 읽기/쓰기 처리 성능과 유연한 구조를 특징으로 확장성이 유리한 Document 지향 형식의 MongoDB를 채택하였다〔7〕.

2.4 MapReduce를 이용한 방화벽 로그 분석

MapReduce는 대용량 데이터를 병렬로 처리하기 위한 프로그래밍 모델로서, Map과 Reduce 두 함수의 조합을 통해서 분산/병렬 시스템을 운용한다. Map과 Reduce는 〔표 3〕과 같은 단계로 이뤄진다: Map 단계는 키(key)와 값(value) 쌍을 입력받아 새로운 키-값의 쌍 집합을 출력하고, Reduce 단계에서는 Map 단계에서 출력된 데이터를 입력받아 각 키

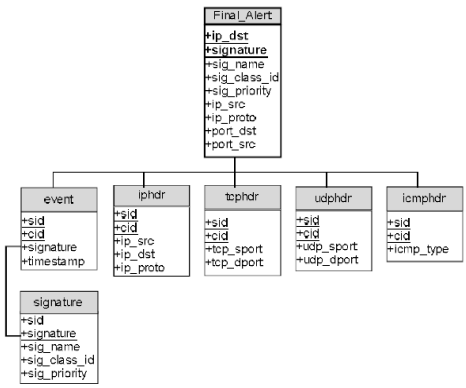
〔표 3〕 MapReduce 단계

Map	(k1, v1)	→ list(k2, v2)
Reduce	(k2, list(v2)	→ list(v2)

에 대한 값의 집합을 출력 한다〔9〕. 즉, Map 함수는 입력 데이터를 필터링 하고 Reduce가 처리할 수 있는 형태로 변형시키는 역할을 하고 Reduce 함수는 이러한 값들을 받아 통합하여 최종 결과를 구축해주는 역할을 한다고 볼 수 있다.

앞서 서론에서 언급했듯이 방화벽 로그데이터는 이를 통과하는 다양한 패킷데이터들의 정보를 담고 있기 때문에, 실제 침입 탐지 분석에 이용되는 주요 정보 이외에도 기타 불필요한 많은 정보필드를 담고 있다. 이에 효율적으로 공격을 탐지하려면 중복된 정보를 포함하여 불필요한 정보들은 제거하고 필요한 정보들만을 간추려 분석을 수행해야 한다. 〔그림 2〕은 Wei-Yu Chen의 이기종 보안 장비에서 발생한 다양한 보안 로그 데이터들을 MapReduce의 설계를 통해 다양한 형태의 보안 로그들을 각각의 특정 필드를 기준으로 병합한 모델을 제시한 것이다. 현재까지 이루어진 대부분의 MapReduce 설계 기반의 로그 분석 기법들은 Wei-Yu Chen의 연구를 전신으로, 이 모델을 연장하여 이루어져 왔다〔15〕〔16〕.

이와 같은 초기 MapReduce 설계의 로그 분석 기법들은 로그 분석 단계에 있어서 그 범위가 Log Parsing 단계에 한정되어 있어, 단순 MapReduce를 이용하여 다양한 보안 로그들을 분류하고 정제하였을 뿐, 실질적으로 로그를 분석하는 과정을 Map-Reduce로 처리하여 구현하지는 가 적용되지는 못하였다. 이에 본 논문에서는 기존에 이루어져 있던 연구



〔그림 2〕 Wei-Yu가 제안한 로그 분석 모델

의 범위를 확장하여 분석단계에 있어서 MapReduce 설계 기법을 적용한 방법을 제시한다.

III. 제안하는 NoSQL 기반의 MapReduce를 이용한 효율적인 로그 분석 기법

3.1 제안하는 방법의 이점

최근 증가하고 있는 보안 장비에 따라, 여기서 발생하는 로그 데이터의 양 또한 기하급수적으로 증가하고 있다. 이에 기존 RDBMS 방식의 로그 분석시스템들은 수집 및 저장 공간이 부족하여 성능 저하를 일으킬 수 있다. 이러한 기존 방식의 취약점은 대용량 로그데이터 환경에서 효율적인 로그분석을 처리하는데 많은 어려움을 야기하고 있다[16].

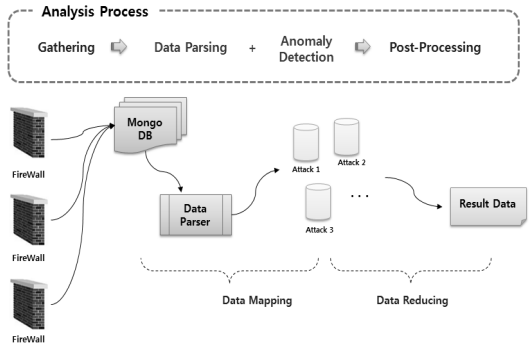
뿐만 아니라, 점차 다양화 되어가고 있는 공격패턴을 보다 신속하게 반영하여 이를 추가하거나 수정하는 것 또한 보안로그 분석에 있어 매우 중요한 부분이다. 그러나 기존 RDBMS 방식의 로그 분석 시스템들은 이러한 요구사항을 반영하고자 할 경우 별도의 테이블 및 이들 간의 관계를 다시 생성하고 새로 정의해야 하는 등 새로운 공격 유형을 반영하는 절차가 복잡하고 그 설계가 어렵다.

이에 본 논문에서는 이러한 취약점을 보완하고, 기존 연구의 범위를 확장하여 대용량화 되어가고 있는 이기종 방화벽 로그 데이터의 분석에 대하여 수집 공간의 제약이 없고 새로운 공격유형 등장에도 이를 유연하고 빠르게 추가 적용 하여 분석할 수 있는 방화벽 로그 분석 기법을 제안하고 있다. 즉, 기존RDBMS 방식의 속도 및 성능 저하 현상을 NoSQL 기반에 분산/병렬 처리에 최적화 되어 있는 MapReduce 프로그래밍 기법을 이용하여 이를 보완하였다. 또한, 다수의 NoSQL솔루션 모델 중에서도 유연한 확장성을 자랑하는 Document-Oriented 형식의 MongoDB를 이용하고, 이를 바탕으로 MapReduce설계를 적용하여 변화하는 공격 유형을 보다 신속하고 효율적으로 반영하여 로그 분석에 쓰일 수 있도록 하였다.

따라서 NoSQL 기반의 MapReduce 설계를 이용한 로그 분석을 통해 기존 로그 분석 기법의 속도 및 처리 성능 문제 개선, 변화하는 공격 패턴을 신속하게 반영하여 유연한 로그 분석을 할 수 있는 이점과 각각의 Data Parsing과 Anomaly detection 과정을 별도로 처리했던 로그 분석 과정을 이용되는 공통의 키(Key)를 추출하여 Map() 함수 수행 과정에서 한 번에 처리함으로써 전체 분석 프로세스를 간소화 할 수 있는 이점이 있다.

3.2 제안 하는 NoSQL기반의 MapReduce를 이용한 로그 분석 기법 처리 프로세스

본 논문에서 제안하고 있는 NoSQL 기반의 MapReduce를 이용한 로그 분석 기법 모델의 전체적 프로세스를 [그림 3]처럼 표현하였으며, 본 절에서는 각 단계에서 처리하는 프로세스에 대하여 소개하고 있으며, 다음은 이에 대한 설명이다.



[그림 3] 제안하는 로그 분석 기법의 처리 단계

- **Gathering 단계:** 데이터 수집 단계로서, 이기종 방화벽에서 발생하는 로그들을 수집하여 BSON 형태의 파일로 임시 저장한다. 수집방법은 앞선 3.1절에서 설명되었듯이 이전 연구기법[10]을 토대로, 본 연구를 위해 임의로 설정한 공격 패턴 3가지를 탐지하는 데 이용되는 로그 데이터만을 수집하였다. [표 4]는 수집한 로그 데이터들의 필드를 나열한 것이고 이에 대한 설명은 2.1절의 [표 1]을 참고 할 수 있다.

[표 4] 수집한 로그데이터 필드 구조

{DateTime; SourceIP; DestinationIP; DestinatioPort; Protocol; ActionCode}

- **Data Parsing 및 Anomaly Detection 단계:** 본 분석 기법에서 이용하는 MapReduce 처리에서의 Map에 해당되는 단계이다. 이전 연구들과는 차별적으로 Map() 함수 안에서 Data parsing과 Anomaly detection을 한

〔표 5〕 Map()함수의 의사코드

```
while(LogList != Null){
map:
function(){
    this.attack.forEach(function(attack_type){
    emit(attack ,{date, count: 1});
  }
```

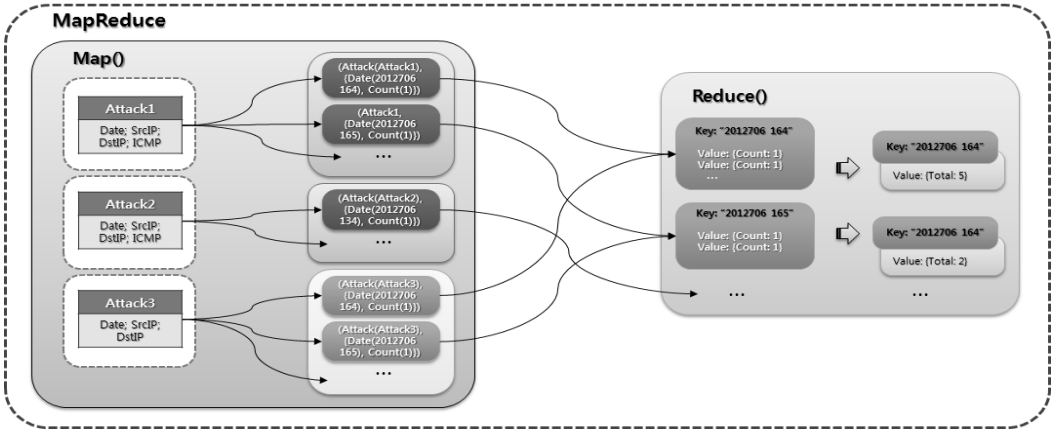
〔표 6〕 Reduce()함수의 의사코드

```
while(LogList != Null){
reduce:
function(key, values){   //(date, count)
total = 0;
for(n = 0; n<values.length; n++){
total += values[n].count;}
return{ attack: total};}
```

번에 처리하여 분석단계를 간소화 시킬 수 있었으며, 추후 이를 시스템화 하였을 때 구성비용이 절감 될 수 있다. Map()함수는 각 해당 공격 유형을 분석하는 데 이용하게 될 필드정보필드를 key값으로 Data Parsing을 수행하고, Parsing 된 데이터들을 가지고 사전에 정의된 Attack 의심 패턴들에 따라 비정상적 행위를 보여주는 로그 데이터들을 판별하게 된다. 〔표 5〕는 본 단계에서 이루어지고 있는 Map함수에 대한 의사코드이다. Map() 함수에서 이용될 (key, value)는 각 유형에 공격 탐지 패턴에 이용될 필드(key)와 이에 해당하는 실제 로그 값(value)으로 구성되며, 다음 Reduce() 함수에서 이용하게 될 새로운 (key, value)를 emit() 한다. Map() 함수에서 이용되는 키 값은 각 공격 유형마다 다르게 구성되어 있으며 〔그림 4〕를 참고하여 이에 대한 정보를 얻을 수 있다.

- **Post-Processing:** 사후 분석에 해당하는 단계로 MapReduce 처리에서 Reduce 함수에 해당하는 단계이다. 즉, Map()에서 emit()된 로그시간 정보(DateTime)와 공격에 대한 집계정보(count)를 Reduce()의 (key, value) 값으로 이용하여 시간별 공격횟수를 측정한 결과 데이터를 산출한다. 〔표 6〕은 Reduce()의 처리 과정을 의사코드로 표현한 것이다.

〔그림 4〕는 본 제안하는 로그 분석 모델의 MapReduce의 전체적인 처리 과정을 그림으로 표현한 것이다. Map() 함수에서 키가 되는 각 공격 유형마다의 이용되는 서로 다른 field의 구성을 가지고 있으며, Map() 함수를 거쳐 불필요하거나 중복된 로그 정보들을 정제하고, 사전에 설정된 공격 패턴 규칙들을 이용하여 Attack을 탐지한다. 여기서 산출되는 emit() 정보를 Reduce() 함수의 (key, value)쌍으로 전달하여 공격이 일어난 로그 데이터에 대한 집계를 산출하고, 특정 시간에 집계된 공격 횟수나 공격 유형 등을 통하여 방화벽 로그에 대해 다양한 분석을 할 수 있다.



〔그림 4〕 MapReduce 처리 과정

IV. 실험 및 결과

4.1 실험 환경 및 실험 시나리오

본 실험은 Intel Core i5 3GHz의 CPU와 8G의 RAM을 사용하였고, Windows7 64bit의 운영체제를 사용하였다. 실험에 사용한 데이터베이스는 NoSQL 기반의 MongoDB-2.4.1 버전과 기존의 관계형 데이터베이스에서 가장 많이 이용되어지는 MySQL-5.6.10 버전을 사용하여 SQL 질의 성능을 테스트하고, 공격의심 시나리오에 대한 탐지 및 분석을 수행하였다. 실험에 사용된 데이터는 실제 기업의 보안 로그 데이터 12,847,649건 중 10,000,000 건을 이용하여 실험하였다. 실험에 사용된 공격의심 시나리오는 〔표 7〕과 같다.

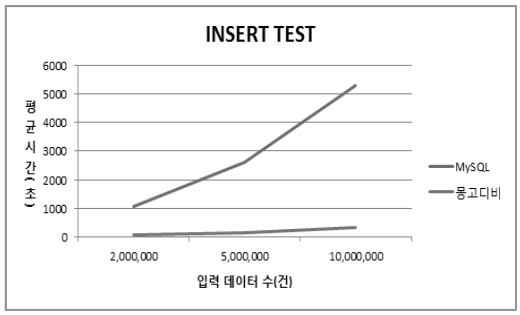
〔표 7〕 공격 의심 시나리오

i. Attack 1 : 동일 근원지에서 동일 목적지로 5분간 5회 이상의 ICMP 메시지 발생
ii. Attack 2 : 50개 이상의 근원지에서 동일 목적지로 5분간 50회 이상의 ICMP 메시지 발생
iii. Attack 3 : 동일 근원지에서 동 목적지로 1분간 10000회 이상의 통신 발생

4.2 성능테스트

본 성능 테스트는 크게 두 가지의 실험으로 이루어진다. 첫 번째는, NoSQL 방식이 기존의 RDBMS 방식에 비해 얼마나 빠른 속도로 대용량의 로그를 수집하여 분석할 수 있는가를 판단하기 위한 실험으로, 각각의 RDBMS와 NoSQL 기반의 프레임워크에서 200만, 500만, 1000만 건의 로그 데이터의 입력 속도를 측정하여 비교하였다. 두 번째로는 분산 컴퓨팅 환경에 최적화 되어 있는 MapReduce기법에 기반하여 1대의 PC에서 프로세스가 처리된 결과와 여러 대의 PC에서 프로세스가 처리된 결과 값을 비교하여 분산처리 환경 구현을 통해 제안하는 분석 모델의 성능이 개선되었음을 보였다.

〔그림 5〕은 삽입 테스트에 대한 결과이다. 삽입 테스트는 로그 데이터를 얼마나 빠른 시간 안에 수집할 수 있는가, 혹은 이미 수집된 로그 데이터를 얼마나 빠르게 가져 올 수 있는가를 시험하는 테스트이다. 그



〔그림 5〕 삽입 테스트 결과

래프를 살펴보면, 비교적 작은 크기의 200만 건의 데이터 삽입과 1000만 건의 데이터 삽입 결과를 비교해 볼 때, MySQL과 MongoDB 간의 속도차가 눈에 띄게 벌어진 것을 볼 수 있다.

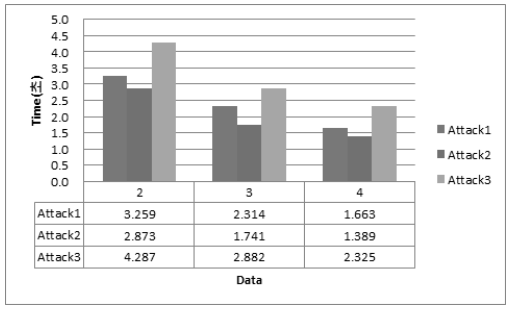
이에 대한 결과를 정리한 것이 〔표 8〕이다. 속도차이를 살펴보면 200만 건의 데이터를 삽입하는 두 데이터베이스 간의 속도차이는 936.275초이고 1000만 건의 데이터를 삽입할 경우에는 4946.733초로 속도차가 약 5배 정도가 증가하였는데, 이는 데이터가 늘어날수록 데이터를 삽입하는 속도 간의 차도 비례하게 증가하고 있음을 알 수 있다.

〔표 8〕 삽입 테스트 결과

구분	200만	500만	1000만
MongoDB(초)	59.022	152.107	340.533
MySQL(초)	995.298	2470.203	4946.733
속도차(초)	936.275	2318.096	4606.2

(단위 시간: 초)

다음으로는 두 번째 실험인 분산 컴퓨팅 기술에 기반 하여 분산 컴퓨팅 환경을 구현하고 분산 노드의 개수에 따른 제안하는 분석 모델의 공격 탐지 수행시간을 측정하여 비교한 것이다. 각 공격 유형마다 동일한



〔그림 6〕 분산테스트 결과

작업을 10번씩 수행하여 이에 대한 평균을 산출하여 다음 〔그림 6〕과 같이 결과 그래프로 나타내었다.

〔그림 6〕을 살펴보면 기존의 MapReduce를 처리하는 노드의 개수가 늘어남에 따라 수행시간이 점차 단축되고 있는 것을 볼 수 있다. 이는 분산 환경을 구현함으로써 제안하는 분석 기법의 성능이 향상될 수 있음을 알 수 있으며, 추후 보다 섬세하고 확장된 분산 처리환경이 조성된다면 보다 실시간적인 로그 분석이 수행될 수 있을 것이다.

또한, 〔표 9〕는 각 유형의 공격 수행시간에 대한 표준편차를 정리한 것으로, 노드의 수가 많을수록 표준편차의 값이 작게 산출 되고 있다. 이는 분산된 환경에서의 로그 분석이 보다 안정적으로 수행되었음을 보여 준다.

〔표 9〕 분산 환경 처리 수행시간에 대한 표준편차

구분	2	3	4
Attack 1	0.068	0.016	0.019
Attack 2	2.256	1.625	1.163
Attack 3	4.287	2.882	2.325

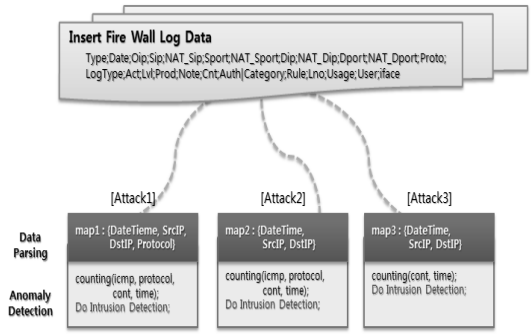
(단위 시간: 초)

4.3 NoSQL기반의 MapReduce를 이용한 로그 분석 테스트

본 실험은 앞서 수행되었던 성능테스트 과정에서 수집된 로그 데이터를 이용하여 실제 분석을 처리 하는 과정으로 3.2절에서 언급했던 본 논문에서 제안한 방화벽 로그 분석 기법의 'Analysis'와 'Post-Processing' 단계에 해당된다. 이는 'Gathering'단계에서 수집된 데이터를 Map()함수에서 각 공격 유형을 판단하는 데 이용되는 로그 필드(key)와 이에 해당하는 실제 로그 값(value)을 (key, value)로 설정하여 로그 데이터를 Parsing하고, Parsing된 데이터를 가지고 사전에 설정된 공격 탐지 규칙에 적용하여 공격을 판단한다.

〔그림 7〕은 이에 대한 처리 과정을 보여주는 그림이다. 즉, Map()의 기능을 가시화 한 내용으로, 여기서 emit()되는 로그 시간(key)과 해당 시간에 대한 공격 탐지 발생 카운트(value)를 Reduce()의 (key, value) 값으로 이용하여 'Post-Processing'을 수행하게 되는 것이다.

실험 데이터로는 실제 기업에서 운영 중인 방화벽 장비에서 발생한 1000만 건의 로그 데이터 중의 일부



〔그림 7〕 Data Gathering 및 Parsing 과정

를 실험을 위해 사전에 설정한 Attack 시나리오를 바탕으로 이에 공격 시나리오를 설정하여 구성한 데이터로, 이에 대한 공격 탐지 하고 이를 바탕으로 사후 분석에 이용될 데이터를 추출함으로써 로그 분석 기법에 대한 설계가 잘 수행되었음을 을 평가했다. 〔그림 8〕은 실제로 분석 결과가 출력된 화면으로, 출력 형식은 다음과 같다

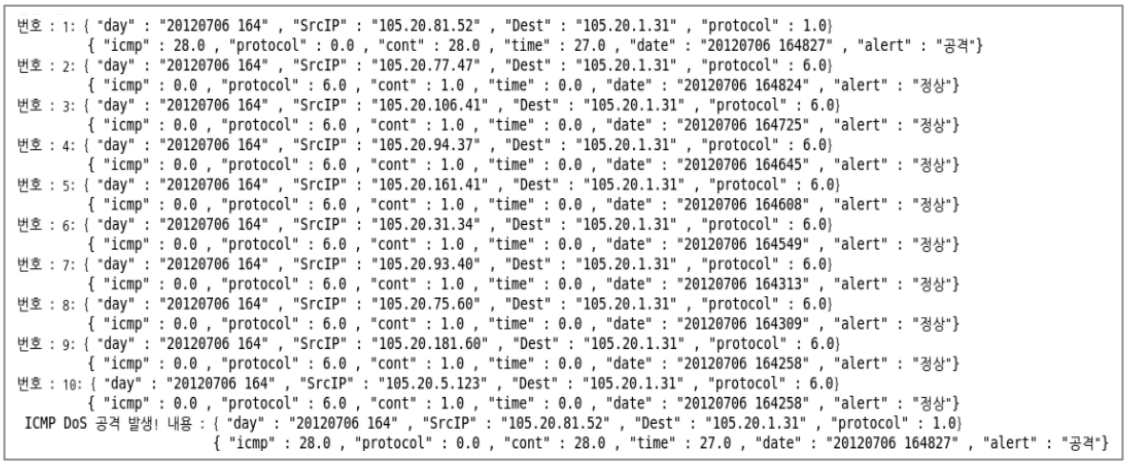
- Day: 발생시간(10분 간격으로 분할)
- SrcIP: 근원지 IP
- Dest: 목적지 IP
- Protocol: 프로토콜 번호(〔표 1〕 참조)
- ICMP: ICMP 발생 횟수
- Cont: 해당 key에서 발생한 이벤트횟수
- Time: 해당 key의 총 접속 시간
- Date: 해당 key의 최초 이벤트 발생 시간
- Alert: Reduce 내 공격 판단 결과

(1) Attack1 의심: 동일 근원지에서 동일 목적지로 5분간 5회 이상의 ICMP 메시지 발생

- 공격 시나리오
  - 소스 ip(srcIP): 105.20.81.52
  - 목적지 ip(dstIP): 105.20.1.31
  - 프로토콜: 1(ICMP)
- 실험 결과 출력 내용

```
ICMP Attack1 발생! 내용 :
{ "day" : "20120706 164" , "SrcIP" :
"105.20.81.52" , "Dest" :
 "105.20.1.31" , "protocol" : 1.0}
```

(2) Attack2 의심: 50개 이상의 근원지에서 동일 목적지로 5분간 50회 이상의 ICMP 메시지 발생



〔그림 8〕 공격 분석 결과 출력 내용

- 공격 시나리오
  - 근원지 ip(srcIP): 임의의 SrcIP 생성
  - 목적지 ip(dstIP): 105.20.1.250
  - 프로토콜: 1(ICMP)
- 실험 결과 출력 내용

> ICMP Attack2 공격 발생! 목적지 IP : "105.20.1.31" 로 ICMP 160회 발생

(3) Attack3 의심: 동일 근원지에서 동 목적지로 1분간 10000회 이상의 통신 발생

- 공격 시나리오
  - 근원지 ip(srcIP): 105.20.53.112
  - 목적지 ip(dstIP): 105.20.1.17
- 실험 결과 출력 내용

> Attack3 발생!
> 내용 (10000회 이상 접근시도):
> { "SrcIP" : "105.20.53.112" , "Dest" : "105.20.1.17" , "day" : "20120706 164"}

각 실험결과 출력 화면을 살펴보면 사전에 정의된 공격 의심 패턴을 기반으로 실험을 위해 고의적으로 설정해 놓은 각 실험용 공격 로그를 정확히 탐지해 냈음을 알 수 있다. 이는 MapReduce를 통한 방화벽 로그 분석이 수행되어 공격 분석 시에 해당 공격을 정확히 추출해 냈음을 의미한다.

이러한 실험을 바탕으로 〔표 10〕 연구 진행 과정동아 축적된 경험을 바탕으로 RDBMS와 비교하여 정성적인 분석을 하였다. 실제 로그 분석을 위해 NoSQL방식의 MapReduce를 설계해 봄으로써, 비교적 단순한 코드 구현이 가능하였다. 따라서 MapReduce를 이용한 공격 분석이 유연하고 높은 생산성을 제공할 수 있다고 판단하였다. 그 이유는, NoSQL은 비 관계형 데이터모델이기 때문에 복잡한 관계 연산 없이도 MapReduce와 같은 기법을 통해 데이터들 간의 연관성을 단순하고 용이하게 표현할 수 있기 때문이다. 즉, 실제 새로운 공격이나 변화된 공격 패턴에 대한 요구사항을 반영하는데 NoSQL 방식의 데이터 모델을 이용했을 때 더 단순한 설계로 설계 생산 비용이 감소 될 수 있다는 것이다.

V. 결론

네트워크 기술의 발달로 방화벽을 지나는 패킷 데이터의 양은 엄청난 속도로 증가하고 있다. 본 논문에서는 기존의 전통적인 데이터베이스 모델인 RDBMS의 한계를 지적하면서 최근 이슈가 되고 있는 빅데이터와 함께 이를 처리하는데 유리한 NoSQL방식의 데이터베이스를 이용하여 대용량 로그를 보다 효율적으로 분석하는 기법을 제안 하였다. 앞의 실험을 통해 알 수 있었듯이 다량의 방화벽 로그 분석을 위해서 비관계 지향적인 데이터 모델링 방식인 NoSQL, 특히 그중에서도 사용이 용이하고 확장성이 좋은 MongoDB의 적용은 대용량 로그 데이터 분석에 긍정적인 가능성을 보였다.

또한 대용량 빅 데이터의 분산 처리 프로그래밍 기

〔표 10〕 NoSQL과 RDBMS 비교 분석

	NoSQL	DBMS
단순성	단순한 비 관계형 데이터 모델링	복잡한 관계형 데이터 모델링
유연성	- 공격 패턴의 다양화에 대응 가능 - 비정형 데이터 관리 및 지원 가능	특정 데이터모델 및 스키마, 질의 언어에 의존적
확장성	대용량 데이터 처리 시, 저가형 PC들을 이용한 Scale-out방식의 구현이 용이	대용량 데이터 처리 시, H/W성능 개선을 통해 처리량을 향상시키는 Scale-up방식
효율성	스키마, 인덱스, 고차원 언어 등의 미지원	스키마, 인덱스, 고차원 언어 등의 지원

법인 MapReduce를 활용한 방화벽의 공격탐지의 구현을 통해, Map을 어떻게 구현 하는 가에 따라 Reduce성능이 어떻게 나타나는지 알 수 있었다.

따라서 향후에는 다양한 공격 유형별 탐지 목적에 맞게 동적(dynamic)으로 Map의 키를 설정하는 방법과 이를 이용하여 다양한 공격 패턴에 대응할 수 있는 실시간 침입 탐지 시스템에 대한 연구를 하고자한다.

참고문헌

〔1〕 민경식, "네트워크 시대의 사회적 위험과 정보보호," 전자공학회지, 35(12), pp.52-63, 2008년 12월.

〔2〕 행정자치부, 전자정부전문위원회, "정보시스템 구축 운영 기술 가이드라인 2.0," 2005년 10월.

〔3〕 Shaimaa Ezzat Salama, Mohamed I. Marie, Laila M. El-Fangary, Yehia K. and Helmy, "Web Sever Logs Preprocessing for Web Intrusion Detection," Computer and Information Science, vol. 4, no. 4, pp.123-133, July. 2011.

〔4〕 방화벽 관리 및 침입 기록 분석방법, NCSC-TR050016, 국가사이버안전센터, 2005년.

〔5〕 윤성종, 김정호, "방화벽 로그를 이용한 침입탐지기법 연구," Journal of Information Technology Applications & Management, 13(4), pp.141-153, 2006년 12월.

〔6〕 김형준, 조준호, 안성화, 김병준, 클라우드 컴퓨팅 구현 기술, 에이콘, 2011년 1월.

〔7〕 Kyle Banker, MongoDB in Action, Oreilly & Associates, Aug. 2010.

〔8〕 http://nts_story.blog.me/50116614473

〔9〕 Jeffrey Dean and Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM - 50th anniversary issue, vol. 51, no. 1, pp.107-113, Jan. 2008.

〔10〕 최대수, 문길종, 김용민 ,노봉남, "MapReduce를 이용한 대용량 보안 로그 분석," 한국정보기술학회 논문지, 제9권, 제8호, pp.125-132, 2011년 8월.

〔11〕 Wei Xu, Ling Huang, Armando Fox, David Patterson, and Michael I. Jordan, "Detecting Large-Scale System Problems by Mining Console Logs," *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pp. 117-132, Oct. 2009.

〔12〕 천준호, 신동규, 장근원, 전문석, "DDoS 공격에 대한 방화벽 로그 기록 취약점 분석," 한국정보보호학회논문지, vol. 17, no. 6, pp.143-148, 2007년 12월.

〔13〕 Karanjit and Chris Hare, Internet Firewalls and Network Security, Second Ed, New Readers Pub, Dec. 1996.

〔14〕 이미영, 최완, "빅데이터 처리 및 저장기술동향 및 전망," 한국정보통신학회지, vol. 13 no. 1, pp.33-39, 2012년 6월.

〔15〕 Wei-Yu Chen, Wen-Chieh Kuo, and Yao-Tsung Wang, "Building IDS Log Analysis System on Novel Grid Computing Architecture," WoGTA, 2009.

〔16〕 Wei-Yu Chen and Jazz Wang, "Building a cloud computing Analysis System for Intrusion Detection System," CLOUD SLAM, Apr. 2009.

〈저자소개〉

최 보 민 (Bomin Choi) 정회원
2012년 2월: 경원대학교 컴퓨터미디어학과 졸업(공학사)
2012년~현재: 가천대학교 일반대학원 전자계산학과(석사과정)
〈관심분야〉 Security, Algorithm, BigData

공 종 환 (Jong-Hwan Kong) 정회원
2012년 2월: 경원대학교 컴퓨터소프트웨어학과 졸업(공학사)
2012년~현재: 가천대학교 일반대학원 전자계산학과(석사과정)
〈관심분야〉 Network Security, Information Security

홍 성 삼(Sung-Sam Hong) 정회원
2009년: 경원대학교 전자거래학과 졸업(공학사)
2011년: 경원대학교 전자계산학과 석사과정 졸업(공학석사)
2011년 2월~현재: 가천대학교 전자계산학과 박사과정
〈관심분야〉 Security, Network, Optimization, Algorithm

한 명 묵(Myung-Mook Han) 종신회원
1980년: 연세대학교 공과대학 졸업 (공학사)
1987년: 뉴욕공과대학교 컴퓨터공학과 석사 졸업 (공학석사)
1997년: 오사카시립대학교 정보공학부 졸업(공학박사)
1998년~현재: 가천대학교 IT대학 컴퓨터공학과 교수
〈관심분야〉 Security, Algorithm, Data Mining