韓國 컴퓨터情報學會 論文誌
第10卷 第3號, 2005. 7.

2005-10-3-2-9

패킷 분석을 이용한 내부인 불법 질의 탐지

장 경 옥*, 구 향 옥**, 오 창 석***

Detection of Internal Illegal Query Using Packet Analysis

Gyong-Ohk Jang *, Hyang-Ohk Koo **, Chang-Suk Oh ***

요 약

정보 통신의 확산으로 일반 사용자도 정보 제공 매체 등을 통해 데이터베이스의 정보를 주고 받는 일련의 행위가 일상화되도록 변화되었다. 이러한 환경 변화는 데이터베이스에 대한 사용이 증가되고 사용의 편리성에 상반된 보안의 취약성을 야기한다. 본 논문에서는 이러한 문제점을 해결하기 위하여 패킷 분석을 통해 내부인 불법 질의를 탐지하는 방법을 제안한다. SQL 구문에 관련된 패킷을 분석 자료로 구축하기 위해서 네트워크 상의 패킷을 읽어 들여 각 프로토콜별로 헤더를 분석하였다. 패킷 중에서 TCP 세그먼트의 데이터부분에 SQL 구문이 있는 경우 SQL 구문을 사용자 권한 정보와 사용자 하드웨어 정보를 이용하여 분석하므로 사용된 SQL 구문이 사용자의 접근 통제 범위내의 질의가 입력되었는지를 탐지할 수 있는 방법을 제안하고자 한다.

Abstract

The purpose of this study is for designing a illegal query detection system using Winpcap library for unauthorized access by internal person. The illegal query detection can be possible detecting the data in out of access control or searching illegal data by plagiarizing other user ID.

The system used in this paper collects packets and analyzes the data related to SQL phrase among them, and selects the user's basic information by comparing the dispatch of MAC address and user's hardware information constructed previously. If the extracted information and user's one are different, it is considered as an illegal query.

It is expected that the results of this study can be applied to reducing the snaking off unprotected data, and also contributed to leaving the audit records using user's access log which can be applied to the pattern analysis.

▸ Keyword : SQL(Structured Query Language), Packet Analysis, Illegal Query Detection

• 제1저자 : 장경옥
• 접수일 : 2005.06.07, 심사완료일 : 2005.07.10
* 충북대학교 전기전산공학과, ** 충북대학교 컴퓨터공학과, *** 충북대학교 전기전자컴퓨터공학부

Ⅰ. 서 론

과거 데이터베이스 환경은 분산되지 않고 중앙 집중식으로 관리하므로 권한이 부여된 특정 사용자와 관리자들만이 접근할 수 있도록 통제되어 보안에 관련된 문제는 거의 보고되지 않았다. 중앙 집중식 환경은 정보 통신의 확산으로 일반 사용자도 정보 제공 매체 등을 통해 데이터베이스의 정보를 주고 받는 일련의 행위가 일상화되도록 변화되었다. 이러한 환경의 변화는 데이터베이스에 대한 사용이 증가되고 사용의 편리성에 상반된 보안의 취약성이 증가되고 있다. 이런 취약성은 조직 내부의 내부자 거래에 의한 정보의 흐름이 발생할 때 권한이 부여되지 않은 자료를 유출하는 일이 자주 발생하고 있다[1]. 인터넷상에서 쉽게 내려 받을 수 있는 특정 질의 도구를 이용하여 데이터베이스에 접속 후 관련된 모든 정보를 손쉽게 빼낼 수 있다. 비밀 자료가 포함된 데이터베이스에 대해 내부자 부주의 또는 고의적인 정보 유출과 오용에 대비한 데이터베이스 중심의 방어가 필요하다. 따라서 논문에서는 네트워크 상의 패킷을 읽어 들여 각 프로토콜별 헤더를 분석 후, TCP 세그먼트의 데이터 부분에 SQL 구문을 사용자 권한 정보와 사용자 하드웨어 정보를 이용하여 분석하므로 사용된 SQL 구문이 사용자의 접근 통제 범위내의 질의가 입력되었는지를 탐지할 수 있는 방법을 제안하고자 한다[2].

Ⅱ. DBMS 접근 통제 방법

자료에 비밀이 포함된 시스템에서는 권한이 없는 사용자의 접속을 제어하기 위해 (그림 1)과 같이 DBMS의 접근 통제 정책을 사용하고 응용 프로그램에서 각 사용자별 접근 통제 정책을 사용하여 관리할 수 있다.

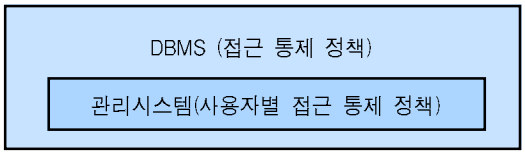

그림 1. 비밀이 포함된 시스템 접속 통제 방법
Fig. 1. Private system access control method

다수의 사용자가 공통의 데이터를 접근하게 되는 데이터베이스내의 데이터 내용을 임의의 사용자가 검색하거나 변경할 가능성이 존재한다. 이런 경우 DBMS에서는 데이터베이스를 사용할 수 있는 자격인권한을 부여해 줄 수 있다. 여러 부류의 사용자들이 여러 수준의 권한을 가질 때 DBA는 사용자별로 연결 권한 또는 테이블-레벨 권한을 사용하여 비밀 자료의 테이블이나 칼럼으로의 접근을 제한 할 수 있다. 그리고 비밀 테이블이나 칼럼으로의 제한된 접근을 제공하기 위하여 내장 프로시저를 사용할 수 있다. 일부 사용자에게 테이블에 대한 모든 접근을 제한할 수 있고 비밀 행이나 칼럼을 보여주지 않는 뷰를 통하여 접근을 허용할 수 있다. 권한이 주어지면 각각의 권한은 시스템 카타로그 테이블에 기록되며 사용자가 특정 질의를 하였을 때 무슨 권한이 부여되었고 누구에게 부여되었는지를 찾기 위해 시스템 카타로그 테이블을 참조하여 처리한다[3].

DBMS에서의 접근 통제는 (그림 2)와 같이 GRANT와 REVOKE 명령을 사용하여 권한을 부여하거나 취소할 수 있다[4].

```
GRANT 권한 <on 테이블명(컬럼명)> TO user명
REVOKE 권한 <on 테이블명> FROM user명
```

그림 2. GRANT/REVOKE 명령
Fig. 2. GRANT/REVOKE command

데이터베이스 또는 테이블을 생성하면 각 권한은 모든 사용자에게 허용된 상태이고 각 사용자별로 권한을 설정하려면 각 사용자별로 여러 개의 GRANT문을 실행하여야 한다. 그리고 사용자가 변경된 경우 이전 사용자의 권한을 취소하여야 하고 새로운 사용자에게 권한을 다시 설정해야 하는 지속적인 관리가 필요하다. 이러한 관리에 소홀함이 발생하면 이는 바로 정보의 유출과 직결될 수 있다는 문제점이 존재한다. 비밀 자료가 있는 시스템은 DBMS에서 제공

하는 권한을 이용하여 사용자별로 각기 다른 권한을 부여하여 관리되고 있다. 예를 들어 기록 권한이 있는 사용자에게는 선택, 갱신의 권한을 허용한 반면, 기록 권한이 없고 단지 자료의 조회에 대한 권한이 있는 사용자에게는 선택 권한만을 허용한다. 그리고 일부 특정 테이블에 대해서는 모든 사용자에게 권한을 취소하고 일부 사용자에게만 액세스 권한을 부여하는 등의 보안 정책을 사용한다.

자료의 접근 권한이 각 사용자별로 테이블 또는 칼럼 단위로 제어되지 않고 행 단위로 관리되어야 할 때 데이터베이스에 대한 접근 통제는 특정 범위의 데이터를 세분화해서 제한을 두는 데는 한계가 있다. 이런 경우 사용자 ID에 따라 세분화된 접근 권한 자료를 데이터베이스에 구축하고 이를 응용 프로그램과 연결하여 접근을 통제할 수 있다. 시스템의 초기 시작 단계에서 사용자 권한 정보를 참조하고 이 정보에 따라 모든 자료 조회 시 접근을 통제하고 접속 로그를 기록하므로 세분화된 접근 통제를 할 수 있다[5]. 그러나 만약 내부자 거래에 의해 시스템의 사용자 계정을 도용한다면 자료의 접근 범위에 벗어난 자료가 유출된다는 문제점이 있다. 또한 내부자의 정보 유출로 인해 특정 질의 도구인 SQL*Plus 또는 파워 빌더 등을 이용하여 불법적인 질의를 한다면 질의에 대한 대부분의 로그가 기록되지 않은 채 자료와 데이터베이스의 모든 정보가 유출될 수 있다는 문제점이 있다[6]. 이러한 문제는 DBMS에서 제공하는 자료의 접근 통제가 테이블 또는 칼럼의 단위로만 처리되고 행 단위로 세분화하여 접근을 제어하기에 한계가 있기 때문이다.

III. 패킷 분석을 이용한 내부인 불법 질의 탐지

본 논문에서 제안한 내부인 불법 질의 탐지 과정을 (그림 3)에 도시하였다.

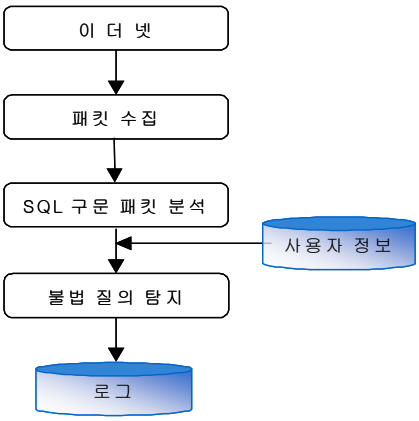

그림 3. 불법 질의 탐지 과정
Fig. 3. Illegal query detection process

제안한 내부인 불법 질의 탐지 과정은 네트워크상의 패킷을 읽어 들여 패킷의 데이터 부분에 SQL 구문이 있으면 불법 질의 탐지 분석을 위한 패킷을 수집한다.

그리고 사용자 정보와 SQL 구문을 비교 분석하여 이를 정상 질의와 불법 질의로 구분하여 로그를 생성하면서 불법 질의를 탐지한다.

제안한 방법을 적용하여 패킷을 획득하기 위해 본 논문에서는 Van Jacobson 등이 개발한 라이브러리 Libpcap (이하 pcap)을 사용하였다[7]. pcap은 간단하게 패킷을 수집하기 위해 사용하는 라이브러리이다. pcap 외에도 패킷을 수집하기 위한 도구들이 있지만 대부분 운영 체제에 종속적이라 운영 체제별로 프로그램을 재작성해야 하는 단점이 있다. 이에 비해 운영 체제에 상관없이 범용적으로 사용하는 라이브러리인 pcap을 사용하였다. pcap은 윈도우 플랫폼에서 호환성을 위해 Winpcap을 제공하며 본 논문에서 이를 활용하였다.

패킷의 내용 중 SQL 구문의 발췌는 애플리케이션 헤더 부분의 파라미터 분석을 통해 수집 자료로의 채택 여부를 결정하였다. 본 논문에서는 상용 DBMS인 인포믹스를 사용하여 애플리케이션과 통신시의 파라미터 값을 실험을 통해 취득한 내용을 (그림 4)에 도시하였다.

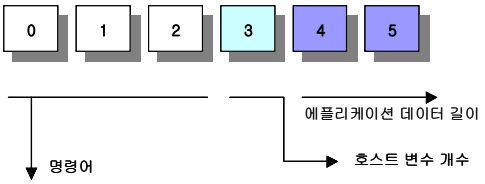

그림 4. 애플리케이션 헤더
Fig. 4. Application header

애플리케이션과 인포믹스가 통신시 앞 세 자리는 사용자 SQL 구문, 시스템 정보 교환, 데이터베이스 접속, 데이터베이스 접속 해제, 그 외의 정보 교환 등을 위해 사용된다. 그리고 네 번째 자리는 SQL 질의시 동적 파라미터인 호스트 변수의 개수를 나타내고, 다섯 번째와 여섯 번째 자리는 사용자가 실제로 질의한 SQL 구문의 길이를 나타낸다. 명령어 부분에서 0 1 0 로 시작하는 것은 실제 사용자가 질의한 SQL 구문이 아니고 애플리케이션과 인포믹스 사이에 정보 교환시 사용된 내용이다. 패킷의 내용을 분석해 보면 시스템 카타로그 테이블과의 정보 교환시 발생되는 SQL 구문이며 주로 자료의 접근 권한을 점검하는데 사용되고 있다. 그리고 실제 사용자가 질의 한 SQL 구문은 실험을 통해 명령어 부분이 0 2 0 으로 시작됨을 알 수 있었다. 본 논문에서는 애플리케이션 파라미터의 명령어 부분이 0 2 0 값을 가진 패킷만을 사용자가 실제 질의한 SQL 구문으로 발췌하여 <표 1>의 형식으로 불법 질의 분석용 자료를 수집하였다.

표 1. 패킷 수집 테이블
Table 1. Packet capture table

필드 명	데이터타입	속성	비 고
no	serial	PK	패킷 처리 순서
bd	datetime		패킷 수집 시간
id	char(6)		사용된 시스템 계정
sn	char(10)		사용자의 사번
mac_addr	char(24)		사용자의 MAC 주소
ip_addr	char(15)		접속한 사용자의 IP 주소
capture_sql	char(255)		사용자가 질의한 SQL 구문
gflag	char(1)		SQL 구문의 유효성 구분

수집된 SQL 구문을 분석하기 위하여 사용자 하드웨어 정보를 추가하였다. 내부자의 발신지 IP 주소 도용에 따른 정보 유출 가능성을 배제하기 위해 사용자가 사용하는 장비 정보를 <표 2>와 같이 구축하였다. 발신지 MAC 주소를 기반으로 사용자 하드웨어 정보와 사용자 권한 정보를 이용하여 사용자의 접근 범위를 찾아낼 수 있다. 접속된 사용자 IP 주소와 시스템 계정이 실제 등록된 사용자 하드웨어 정보와 비교하여 불법 접근 여부를 판단하였다.

표 2. 사용자 하드웨어 정보 테이블
Table 2. User H/W Information table

필드 명	데이터 타입	속성	비 고
sn	char(10)	PK	사용자의 사번
mac_addr	char(24)		사용자의 MAC 주소
ip_addr	char(15)		사용자의 IP 주소
id	char(6)		사용자의 시스템 계정

제안한 시스템에서 사용될 패킷은 TCP 패킷이므로 분석을 위해 모든 패킷을 수집하는 것은 불필요하다. 이를 위해 Winpcap에서 제공하는 pcap_compile과 pcap_setfilter 명령을 사용하여 필요한 패킷만을 필터링하였다. pcap_compile은 들어오는 패킷에 필터링 조건을 기술하기 위해 사용하였고 pcap_setfilter는 pcap_compile을 통해서 지정된 필터를 적용시키기 위해 사용하였다. (그림 5)는 실제 패킷 수집시 사용된 패킷 필터링이다.

```
// 필터 조건 기술 및 필터 적용
packet_filter[] = "필터조건" ;
IF (pcap_compile(fp,&fcode,packet_filter,0,netp)==-1) THEN
    printf("pcap_compile 에러 ");
    exit(1);
ENDIF
IF (pcap_setfilter(fp,&fcode)==-1)  THEN
    printf("pcap_setfilter 에러 ");
    exit(1);
 ENDIF
```

그림 5. 패킷 필터링 알고리즘
Fig. 5. Packet filtering algorithm

패킷에서 필요한 정보를 얻기 위해 패킷을 필터링하는 방법은 각 프로토콜의 헤더를 필터링 하는 방법과 응용 프로그램의 데이터를 필터링하는 방법이 있다. 먼저 각 프로토콜의 헤더 필터링 방법은 캡슐화 과정을 통해 추가되는 부가 정보인 이더넷 프레임 헤더, TCP 헤더, IP 헤더 등에

서 얻을 수 있는 자료를 필터링한다[8]. 그리고 응용 프로그램의 데이터를 필터링하는 방법은 TCP 세그먼트에 있는 데이터에서애플리케이션 헤더와 사용자 데이터 부분을 필터링하여 자료로 사용하였다.

```
// 애플리케이션 헤더 길이가 6이므로 애플리케이션 데이터 길이는 6
이하는 처리안함
IF ( 애플리케이션 데이터 길이 < 6) THEN
      EXIT;
IF(애플리케이션 헤더 == '245')     THEN
      // connection에 대한 처리
ELSE IF(애플리케이션 헤더 == '08')   THEN
     // disconnection에 대한 처리
ELSE IF (애플리케이션 헤더 == '020')   THEN
    // 실제적인 sql command가 들어오는 경우이므로 sql문을 처리
ELSE IF (애플리케이션 헤더 == '010')   THEN
    // 시스템카타로그 정보 검색
      EXIT;
ELSE IF (애플리케이션 헤더 == '040')   THEN
    // 동적 파라미터에 관련된 sql문
      EXIT;
ENDIF
```

그림 6. 애플리케이션 헤더에서 SQL 구문 필터링
Fig. 6. SQL sentence filtering from application header

(그림 6)은 제안한 시스템에서 애플리케이션 헤더에서 SQL 구문을 필터링하기 위해 사용한 알고리즘을 보여준다. 실제 질의를 수행하면 관련된 패킷은 한 개만 생성되는 것이 아니고 해당 애플리케이션에서 데이터베이스의 시스템 카타로그 테이블과 접근 통제 정보를 통신하는 등의 많은 패킷이 흐르게 된다.

그래서 TCP 세그먼트의 애플리케이션 헤더 정보를 조사하여 분석 작업에 불필요한 패킷을 제거한 후 실제 사용자가 질의한 SQL 구문만을 발췌하여 수집 자료로 구축하였다. 수집 자료에는 패킷 수집 시간, 발신지 MAC 주소, 발신지 IP 주소, 애플리케이션 헤더를 제거한 TCP 데이터 부분을 포함하였다.

SQL 구문이 포함된 패킷이 수집되면 불법 질의를 탐지하기 위해 4단계의 과정을 거쳐 판단을 내렸다. 첫 번째 단계로 수집된 패킷의 자료에서 사용자 하드웨어 정보와 비교분석하면서 검사를 진행하다.

발신지 MAC 주소를 갖고 사용자 하드웨어 정보에서 해당 장비의 발신지 IP 주소 도용 여부를 판단하였다.

두 번째 단계로 수집된 패킷의 사용자 계정을 가지고 사용자 하드웨어 정보와 비교하여 관리 시스템에 접근할 수 있는 사용자의 유효성을 판단하게 된다.

세 번째 단계로 현재 질의를 던진 사용자가 관리 시스템의 실무자라면 사용자 권한 정보를 참조하여 현재의 질의의 테이블 단위의 범위가 적법한지를 판단한다.

그리고 마지막 단계로 수집된 SQL 구문의 조건 항목에서 사번을 발췌하여 자료 접근의 범위가 행 단위 질의가 가능한지를 판단한다.

위의 과정에서 관리 시스템에서 공통으로 사용되는 SQL 구문 패턴은 비교 대상에서 제외시켰다. 공통으로 사용되는 SQL 구문 패턴은 제안한 시스템을 이용하여 일정 기간 동안 관리자가 관리 시스템을 사용하여 생성된 SQL 구문 중 공통적으로 사용하는 SQL 구문만을 발췌하여 구축하였다. 수집된 SQL 구문이 구축된 SQL 구문과 동일하면 불법 질의 분석에서 제외시켰다.

발신지 IP 주소 도용 여부를 비교하기 위해 수집된 패킷 데이터에서 발신지 MAC 주소를 키로 사용자 하드웨어 정보에서 발신지 MAC 주소에 해당하는 발신지 IP 주소, 시스템 계정, 사용자 사번을 발췌한다. 수집된 데이터의 발신지 IP 주소와 발췌된 IP 주소와 비교하여 상이한 경우 불법 질의 자료로 처리하였다.

수집된 SQL 구문 데이터에서 데이터 정의 문장 (DDL) 및 데이터 제어 문장 (DCL)이 있는지 점검한다. DDL문과 DCL문은 관리자만이 실행하는 문장이므로 수집된 SQL 구문에 이 문장이 포함되면 불법 질의로 처리한다. 그리고 수집된 SQL 구문 데이터에 데이터 조작 문장 (DML) 중 기록(Insert, Update, Delete)에 관련된 구문이 있으면 기록 권한을 점검하고 조회(Select)에 관련된 구문이 있으면 조회 권한에 대한 접근 권한 여부를 판정한다. (그림 7)은 테이블 단위 권한을 점검하기 위한 알고리즘을 나타내고 있다.

```
IF (수집된 SQL 구문에 DDL 문장 또는 DCL 문장) THEN
      //시스템권한위반에 대한 불법 질의 처리
      EXIT;
ENDIF
IF (수집된 SQL 구문에  DML문장)  THEN
   IF (수집된 SQL 구문에 DML문장 중 Select문장 ) THEN
         //조회 및 행 단위 접근 권한 처리
    ELSE
         // 기록에 대한 접근 권한 처리
    ENDIF
ENDIF
```

그림 7. 테이블 단위 권한 점검 알고리즘
Fig. 7. Table based authority inspection algorithm

수집된 SQL 구문 데이터 중에서 사번이 포함된 경우 이 사번과 발췌된 권한 정보를 조합하여 행 단위의 접근 권한을 점검한다. (그림 8)은 행 단위 접근 권한을 점검하는 알고리즘을 나타난다.

```
//수집된 SQL 데이터에서 사번 발췌
발췌사번 = ""
IF (수집된 SQL 구문에 사번 칼럼명 존재) THEN
    사번처음위치 = 수집된 SQL구문의 사번 컬럼명 위치 + 5 ;
     FOR i = 1 , 수집된 SQL구문길이 DO
          IF 숫자(수집된 SQL구문의 i번째 문자열) THEN
             발췌사번 = 발췌사번 +  i번째 문자열 ;
          ELSE
              EXIT ;
          ENDIF
     NEXT
ENDIF
//발췌된 사번과 접근 권한을 키로
// 사용자 권한 정보 테이블에서 자료 발췌
IF (자료가 발췌) THEN
   //접근 권한 있는 자료로 처리
ELSE
   //접근 권한 없는 자료로 처리
ENDIF
```

그림 8. 행 단위 권한 점검 알고리즘
Fig. 8. Row based authority inspection algorithm

Ⅳ. 실험 및 결과 고찰

본 논문에서 제안한 패킷 분석을 이용한 내부자 불법 질의 탐지의 성능을 평가하기 위한 실험은 실험용 질의 패킷을 일정량 전송한다. 질의 패킷은 특정 질의 도구와 관리 시스템을 사용하여 혼용 생성하여 기존 방법을 사용한 경우와 제안한 시스템을 사용한 경우를 비교 분석한 결과를 <표 3>에 나타내었다. <표 3>에서 보는 바와 같이 사용자 A가 관리 시스템을 사용하는 경우는 시스템에서 권한이 없는 자료에 대해서는 정상적으로 조회가 되지 않았다. 비밀 자료에 대한 접근 통제는 관리 시스템의 메뉴에서 원천적으로 막고 있어 접근할 수 없으므로 두 시스템에서 탐지가 되지 않았다.

그러나 특정 질의 도구를 사용하는 경우, DBMS는 행 단위의 접근 통제를 할 수 없으므로 권한이 없는 자료에 대해서도 자료가 검색되어지고 있다. 그러나 제안한 시스템은 권한이 없는 사용자가 질의한 것에 대해서는 불법 질의로 분류시켰다. 내부자 거래에 의해 타인의 관리 계정 또는 시스템 계정을 도용해서 질의를 한 경우에는 기존 시스템은 표면상 접근 통제 정책이 정상적으로 작동하는 것처럼 보인다.

표 3. 불법 질의 탐지 성능
Table 3. Performance of illegal query detection

사용자	질의 구분	자료 건수 (미권한)		기존 시스템		제안 시스템	
		일반	비밀	일반	비밀	일반	비밀
사용자A	관리 시스템	10(5)	5(2)	5		5	
	질의 도구	10(5)	5(2)		5	5	5
사용자B (관리계정 도용)	관리 시스템	10(5)	5(2)			5	
	질의 도구	10(5)	5(2)			5	5
사용자C (ID도용)	관리 시스템	10(5)	5(2)		5	5	5
	질의 도구	10(5)	5(2)		5	5	5
사용자C (IP 주소/ ID 도용)	관리 시스템	10(5)	5(2)			5	5
	질의 도구	10(5)	5(2)			5	5

그러나 엄격히 구분하면 본인의 자료 검색 범위를 벗어난 질의이므로 불법적인 질의로 처리되어야 한다. 관리 시스템은 이를 정상적인 질의로 처리하였지만 제안한 시스템은 이를 불법 질의로 검출해 냈다. 다음으로 권한이 없는 사용자가 자료 유출의 발신지를 속이기 위하여 발신지 IP 주소를 기존 권한이 있는 사용자의 정보를 도용하고 SQL 질의를 한 경우이다. DBMS의 접근 통제 정책은 사용된 시스템 계정이 표면상으로는 권한이 허용된 사용자이므로 이를 불법적인 질의로 처리하지 않고 자료의 검색을 허용하였다. 그러나 제안한 시스템은 불법 질의를 탐지할 때 시스템 계정뿐만 아니라 사용자의 하드웨어 정보를 같이 비교 분석함으로서 발신지 IP 주소 도용이나 시스템 계정 도용에 따른 자료 유출 및 오용에 대한 문제점을 해결하였다.

V. 결 론

본 논문에서는 비인가자의 불법적인 질의 또는 인가자의 자료 오용에 대한 불법적인 질의를 탐지하기 위하여 Wimpcap을 이용하여 네트워크상의 패킷을 분석하였다. 그리고 사용자 권한 정보 및 사용자 하드웨어 정보를 활용하여 불법 질의를 탐지하기 위한 시스템을 제안하고 구현하였다. 제안한 불법 질의 탐지 시스템은 사용자가 질의 한 SQL구문에 대한 정확한 로그 기록과 함께 특정 질의 도구를 이용한 인가되지 않은 질의 및 자료의 접근 범위를 벗어난 질의를 탐지할 수 있었다. 향후, DBMS에 관계없이 불법 질의를 탐지할 수 있도록 알고리즘을 추가한다면 효과적인 불법 질의 탐지를 할 수 있을 것이다.

참고문헌

[1] 한국 산업 기술 진흥원, 산업기밀 정보의 유출자, 2002.

[2] 홍승필, 김영철, 정보보호의 이해, 길벗출판사, 2004.

[3] 강석훈, 문송천, “데이터베이스 보안에 대한 최신 연구 동향”, 통신정보보호학회논문지, 제2권, 3호, 1992.

[4] INFORMIX. 한글 Informix SQL 실습안내서, 2004.

[5] 박석, 양지혜, “데이터베이스 보안을 위한 모델”, 통신정보보호학회 논문지, 제6권, 1호, 1996.

[6] Tomohiro Odaka. 기초부터 배우는 TCP/IP 애널라이저 작성과 패킷 분석, 성안당, 2003.

[7] 김수용 외 2인. “DB Application Firewall과 Web Application Firewall의 연동을 통한 불법적인 SQL 질의 차단기법”, 정보보호학회논문지, 제13권, 2호, pp.686-690 2003.

[8] 오창석, 데이터 통신 수정판, 영한출판사, 2001.

저 자 소 개

장 경 옥

1986년 2월 충남대학교 계산통계학과(이학사)

2003년 2월~현재 충북대학교 전기전산공학과 석사과정

<관심분야> 컴퓨터 네트워크, 정보보호

구 향 옥

1999년 8월 한발대학교 전자계산학과(이학사)

2002년 2월 충북대학교 컴퓨터공학과(공학석사)

2002년~현재 충북대학교 컴퓨터공학과 박사과정

2003년 8월~현재 백석대학 겸임

<관심분야> 컴퓨터네트워크, 뉴로컴퓨터, 정보보호

오 창 석

1978년 2월 연세대학교 전자공학과 (공학사)

1980년 2월 연세대학교 전자공학과 (공학석사)

1988년 8월 연세대학교 전자공학과 (공학박사)

1985년~현재 충북대학교 전기전자 컴퓨터 공학부교수

1982년~1984년 한국전자 통신연구원 연구원

1990년~1991년 Stanford 대학교 객원교수

2001년~2004년 한국콘텐츠학회 논문지편집위원장

2004년~현재 한국콘텐츠학회 상임 고문

<관심분야> 컴퓨터네트워크, 뉴로컴퓨터, 정보보호