패킷 감시·분석을 통한 데이터베이스 보안 모델의 설계 및 구현

홍성진*, 조은애**
*고려대학교 소프트웨어공학과
**고려대학교 컴퓨터학과
e-mail : {jindigit, eacho}@korea.ac.kr

Design and Implementation of Database Security Model Using Packet Monitoring and Analysis

~

Sung-Jin Hong*, Eun-Ae Cho**
*Dept. of Software Engineering, Korea University
**Dept. of Computer Science & Engineering, Korea University

요 약

최근 금융, 공공기관 등에서 개인 정보 유출이 빈번해짐에 따라 사회적으로 심각한 문제가 발생하고 있다. 한국산업기술진흥협회의 조사에 따르면, 이런 정보 유출이 외부의 불법적 시스템 침입으로 인해 발생하는 것보다, 대부분 데이터 접근이 인가된 내부자 소행으로 나타나고 있다. 이는 데이터베이스의 보안 취약성으로 인해, 내부의 비인가자 또는 인가자의 데이터 접근에 대한 통제 정책이 제대로 이루어지지 않기 때문이다. 이에 따라, 본 논문에서는 클라이언트에서 데이터베이스 서버로 요청되는 네트워크상의 패킷 분석을 통한 데이터베이스의 접근통제방법을 제안한다. 제안된 보안모델에서는, 사용자 정보 및 SQL 의 위.변조를 방지하기 위해서 공개키 인증과 메시지 인증코드 교환으로 무결성을 확보하였다. 또한 권한별 테이블의 컬럼 접근통제를 확장하기 위해서 데이터 마스킹 기법을 구현하였다.

1. 서론

인터넷의 등장으로 그동안 오프라인에서 이루어지던 많은 정보교환의 행위들이 온라인상에서 이루어지는 사회구조로 변하고 있다. 이는 많은 정보들의 집중화 현상을 발생시켰고, 정보가 자산이 되는 정보화 사회로 발전하게 되었다. 이에 따라 사회의 급속한 정보화는 개인정보의 가치를 빠르게 상승시키고 있으며, 정보이용자 또한 정보에 접근하려는 다양한 접근 방식을 요구하고 있다.

정보의 적절한 접근통제가 부재한 상황에서의 정보에 대한 다양한 접근허용은, 개인정보의 악의적 대량 수집과 불법적인 이용 및 유출로 이어질 수 있으며 이와 관련하여 최근 국.내외적으로 고객의 정보가 대량 유출되는 개인정보침해사고가 빈번히 발생하고 있다. 또한 유출된 정보들이 누군가에 의해서 다시 수집되고 악의의 목적으로 불법적인 곳에 이용되며, 뿐만 아니라 또 다른 유출로 이어지면서 그 폐해는 점점 심각해지고 있다. 한국산업기술진흥협회의 조사에 따르면 이 같은 정보 유출 사고의 90% 이상이 내부자 소행으로 나타나 내부 정보 유출 방지를 위한 보안강화가 요구된다.

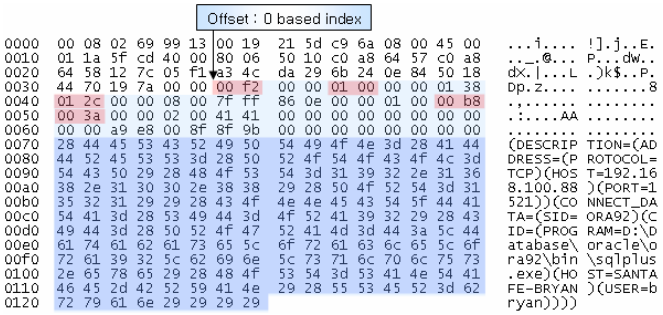
이런 일련의 사건들은 개인정보보호에 대한 인식의 부재가 문제이다. 그러나 더욱 중요한 것은, 데이터에 접근할 수 있는 인가자 및 비인가자의 접근통제가 올바르게 이루어지지 않고 있다는 점이다. 이는 데이터베이스에서 제공하는 보안방식으로 정보 이용자들의 다양한 접근 요구사항을 통제하는데 한계가 있기 때문이다.

따라서 본 논문에서는 먼저 정보 이용자들의 다양한 데이터 접근 요구사항을 유연하게 통제하기 위해 네트워크상에서 패킷 취득 및 프로토콜 분석을 수행하고, 이를 통해 다양한 데이터베이스 접근통제방법을 제안하고자 한다[1][2].

2. 패킷분석

본 논문에서는 데이터베이스 접근통제를 수행하기 위해서, 네트워크상에서 통제대상 데이터베이스로 흐르는 패킷 중 TCP 데이터 부분을 취득하여 수집한다. 수집된 정보는 사용자에게 할당된 접근통제정책과 비교하여, 해당 데이터베이스로 접근할 수 있는지 판단하였다.

본 논문에서는 상용 DBMS 인 오라클을 대상으로 데이터베이스 접근통제를 구현하였다. 패킷의 내용 중 접근통제정책을 적용하기 위한 자료의 수집은 패킷의 프로토콜 분석을 통하여 수행하였다[3] [4][6]. 다음 (그림 1) 은 실험을 통해 오라클의 클라이언트 프로세스에서 데이터베이스로 연결요청 시 취득한 패킷의 프로토콜을 분석한 내용 일부를 도시한 것이다.



(그림 1) 클라이언트에서 데이터베이스로 연결요청시의 프로토콜 구조

<표 1> 은 클라이언트에서 데이터베이스 서버로 연결요청에 대한 프로토콜 정보를 기술하였다.

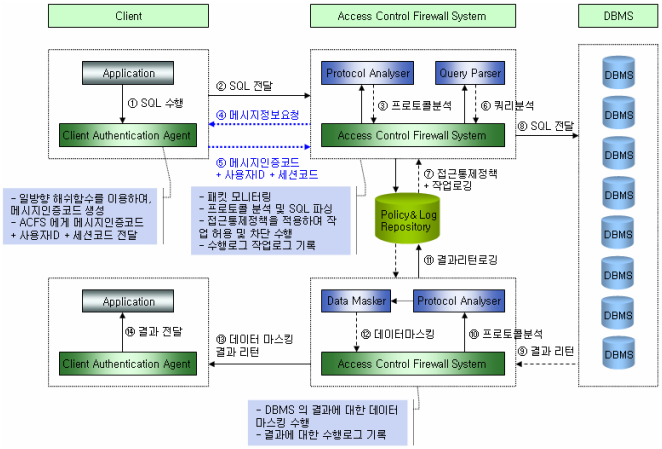
<표 1 > Offset Base : TCP 페이로드(0 based index)

Offset	길이 (byte)	설명
offset + 0	2	TCP 페이로드의 전체길이를 나타낸다.
offset + 4	2	클라이언트가 서버로 보내는 메시지 시작이다. offset + 4 : 0x01 offset + 5 : 0x00 이면 클라이언트에서 서버로 최초접속, 즉 리스너로의 연결요청을 의미한다. offset + 4 : 0x01 offset + 5 : 0x04 이면 리스너의 재지정에 의한 서버로의 연결요청이다.
offset + 10	2	두 바이트 값에 따라 SQL 을 구분하는 기준이 된다.
offset + 24	2	최초 연결요청시, 클라이언트 접속기술자 길이.
offset + 26	2	최초 연결요청시, 클라이언트 접속기술자가 시작되는 위치. offset : 0 based index 를 기준위치로 한다.
offset + 접속기술자 시작위치	접속기술자 길이	최초 연결요청시, 클라이언트 접속기술자가 시작되는 위치이다. offset + 26 에서 시작위치가 기술되었으며, offset + 24 에서 접속기술자의 길이가 기술되어있다.

위에서 기술한 연결요청 이후에, 클라이언트 프로세스와 서버 프로세스간에 여러 환경 정보를 주고 받은 후 정상적인 연결이 이루어진다. 최종 연결이 확립되기까지의 패킷 모니터링과 분석을 통하여, 사용자의 접근통제항목을 적용시킴으로써 데이터베이스로의 연결통제정책을 수립할 수 있다. 다음 본문에서는 제안된 데이터베이스 접근통제의 구조에 대해서 논의한다.

3. 설계

본 논문에서 제안하는 보안모델이 데이터베이스 접근통제를 어떻게 수행하는지 살펴본다. 클라이언트에서 데이터베이스로 작업이 요청될 때, 접근통제시스템이 어떤 방식으로 프로토콜을 취득하고, 분석하여 접근통제를 수행하는지 확인하고자 한다.



(그림 2) 데이터베이스 접근통제시스템 구조

제안하는 보안모델은 클라이언트 인증 에이전트(Client Authentication Agent – CAA, 이하 CAA), 접근통제방화벽시스템(Access Control Firewall System – ACFS, 이하 ACFS), 접근통제정책저장소(Access Control Policy Repository – ACPR, 이하 ACPR) 로 구성된다.

다음은 (그림 2)에서 나타난 접근통제시스템의 전체적인 동작구조를 설명한 것이다.

① 애플리케니션이 데이터베이스로 작업을 시도하면, CAA 는 NDIS 영역을 감시하고 있다가 Network Adapter 로 전달되는 패킷을 취득한다. 취득한 패킷으로부터 이더넷 프레임 헤더, TCP 헤더, IP 헤더를 제거하고, 메시지를 추출한다[7]. 추출된 메시지는 일방향 해쉬함수를 통해 인증코드가 생성된다. 생성된 메시지 인증코드는 메시지 인증코드리스트에 보관된다.

② CAA 에서 작업이 완료된 패킷은 애플리케이션으로부터 전달받은 원본 그대로 네트워크를 통해 데이터베이스로 전달된다.

③ ACFS 는 네트워크를 감시하고 있다가, 접근통제 데이터베이스 서버로 전달되는 패킷을 취득한다. 취득된 패킷은 프로토콜 분석기를 통해 이더넷 프레임 헤더, TCP 헤더, IP 헤더로부터 IP, Mac Addresss 를 추출한 후, 사용자 인증을 거쳤는지 확인한다. 사용자 인증인 안된 접근이면 그 즉시 차단시킨다. 정상적인 접근으로 판단되면, 패킷으로부터 메시지를 추출하고 일방향 해쉬함수를 통하여 인증코드를 생성한다.

④ ACFS 는 메시지의 무결성 및 사용자 인증을 확인하기 위해 CAA 에게 메시지에 대한 정보요청을 한다.

⑤ CAA 는 ACFS 로부터의 메시지 정보 요청을 받고, 메시지인증코드, 사용자 ID, 세션고유코드를 세션키 암호화와 개인키 암호화화를 거쳐 ACFS 로 전달한다.

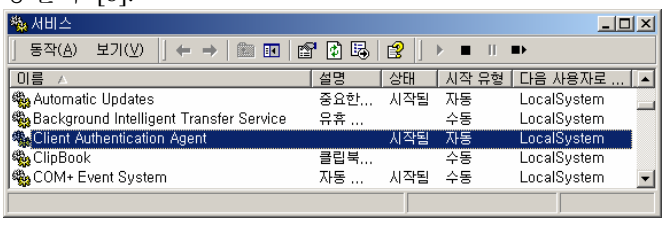
⑥ ACFS 는 CAA 로부터 전달받은 메시지인증코드와 사용자 ID, 세션고유코드를 가지고 인증받은 접근인지

확인한다. 메시지에 대한 무결성이 완료되면 쿼리분석기를 통해 쿼리를 분석한다.
⑦ ACFS 는 ACPR 로부터 접근통제정책을 수집 후, 분석된 쿼리로부터 통제정책을 적용하여 결과에 따라 작업허용 및 차단을 수행한다. 허가되지 않은 접근을 시도한 경우, 관리자에게 통보된다. 일련의 작업들은 ACPR 에 기록된다.
⑧ ACFS 로부터 작업이 완료된 패킷은 클라이언트에서 전달받은 원본 그대로 네트워크를 통해 데이터베이스에 전달된다.
⑨ 데이터베이스가 작업처리를 하고 결과를 클라이언트에 전달할 때, ACFS 는 네트워크를 감시하고 있다가 패킷을 취득한다.
⑩ 취득된 패킷은 프로토콜 분석기를 통하여 패킷을 분석한다.
⑪ 분석된 패킷으로부터 정보를 추출하고 ACPR 에 필요한 정보를 기록한다.
⑫ ACPR 에 기록을 완료한 후, 패킷으로부터 데이터 마스킹에 필요한 정보들을 추출한다. 해당 결과에 마스킹이 필요한 데이터가 있다면 데이터 마스킹 작업을 수행한다. 마스킹이 필요가 없는 데이터는 원본 그대로 전달된다.
⑬ 데이터 마스킹 작업이 완료된 패킷은 변경된 정보를 가지고 패킷을 재생성하여 클라이언트에 전달한다.
⑭ CAA 는 ACFS 로부터 전달받은 패킷을 취득하지 않고 그대로 통과시켜 애플리케니션으로 보낸다.

CAA 는 클라이언트에 위치하면서 사용자에 대한 공개키인증과 데이터베이스로 수행되는 SQL 의 무결성을 보장한다. ACFS 는 접근통제대상이 되는 데이터베이스 서버로 접근하는 패킷을 취득하여 무결성을 검증하고, 패킷에 대한 접근통제정책을 적용하여 데이터베이스로의 접근 허용 및 차단을 수행하게 된다. ACPR 은 사용자에 대한 모든 접근통제정책과 로그인 및 작업에 대한 모든 로그를 가진다.

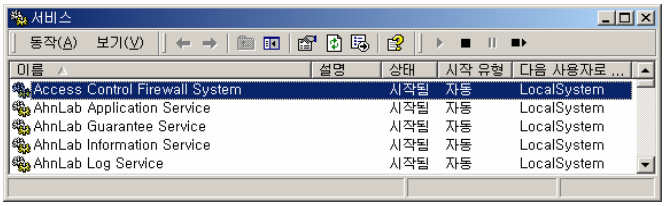
4. 구현

본 논문에서는 상용 DBMS 인 오라클을 대상으로 데이터베이스 접근통제를 구현하였다. 본 논문에서 제안하는 방법중 네트워크상에서 패킷의 취득 및 데이터 마스킹을 하기 위해서, Vadim V.Smirnov 가 개발한 Windows Packet Filter Kit 을 이용한다. Windows Packet Filter Kit 은 윈도우의 TCP/IP Layer 와 Network Adapter 사이의 NDIS 후킹을 통하여 패킷을 취득할 수 있는 라이브러리이다 [7]. 또한 취득된 SQL 의 파싱을 위해서 sqlparser.com 의 General SQL Parser 를 이용한다 [8].



(그림 3) CAA 실행 화면

(그림 3) 은 CAA 가 서비스 형태로 실행된 모습이다. 클라이언트에 위치하면서 사용자인증 및 메시지에 대한 무결성을 보장한다.



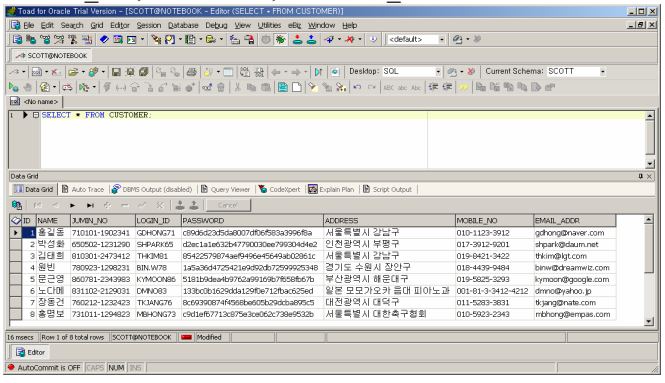
(그림 4) ACFS 실행 화면

(그림 4) 는 ACFS 가 서비스 형태로 실행된 모습이다. ACFS 는 물리적으로 데이터베이스 서버의 앞단에 위치하면서, 데이터베이스로 접근하는 모든 패킷을 통제한다. 데이터베이스로 접근이 허용된 작업 이외에는 모든 작업을 차단시킨다.

본 논문에서는 보안요소의 하나로 데이터 마스킹을 제안하는데, 데이터베이스 서버의 수행결과에 대한 처리결과를 분석해서, 마스킹이 필요한 데이터인지를 판단한다. 마스킹이 필요한 데이터의 경우, 첫번째 문자만 '*' 로 처리하고, 나머지는 널 스티링으로 처리한다.

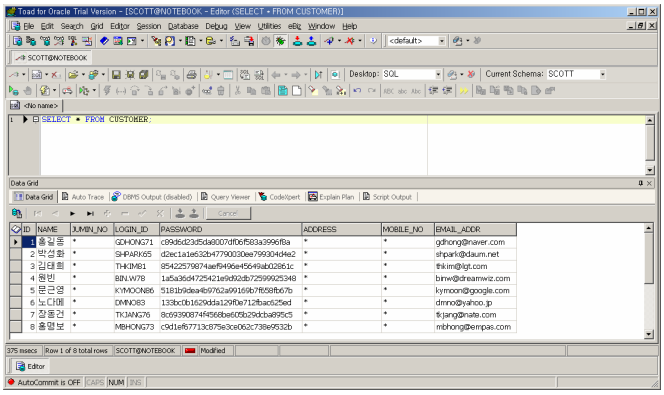
본 논문에서는 수행결과에 대해서 결과 데이터를 암호화 하지 않고, 첫번째 문자에 대해서' *' 로 처리하고, 나머지 길이에 대해서는 널스트링 처리를 한다. 데이터 마스킹은 컬럼에 대한 접근권한이 없는 사용자에게 데이터를 숨기는데 목적이 있다. 따라서, 데이터 암호화나 데이터 마스킹이나 그 목적을 이루는데는 차이가 없다. 또한, 암호화된 데이터는 암호화 알고리즘에 따라 원래 데이터 길이보다 길어지는 경우가 발생한다. 이는 오라클 데이터베이스 서버에서 구조화한 프로토콜의 내용변경을 의미한다. 암호화로 인해서 길이가 변경된다면, ACFS 에서 오라클의 프로토콜에 맞도록 패킷자체를 완전히 새로 만들어야 하는 비용이 든다. 또한 TCP 헤더의 페이로드 길이정보도 변경되어야 한다. 이는 마스킹이 필요한 데이터에 대해서, 단순히 첫문자의 ' *' 처리와 나머지 길이의 널 스트링 처리 후 TCP 의 체크섬만 변경하는것에 비해서 처리비용이 너무 많이 소요되는 문제가 발생하기 때문에 데이터 마스킹 방식을 채택하였다.

클라이언트 테스트 프로그램으로는 세계적으로 유명한 토드를 이용하였다. 마스킹 적용 컬럼으로는 JUMIN_NO, ADDRESS, MOBILE_NO 를 선택하였다.



(그림 5) 컬럼마스킹 적용 전 수행결과

(그림 5) 은 제안된 보안모델 적용 전 토드에서의 SQL 수행 결과이다. 쿼리 결과로서, 모든 컬럼에 대한 정보를 확인할 수 있다



(그림 6) 컬럼마스킹 적용 후 수행결과

(그림 6) 는 제안된 보안모델 적용 후 토드에서의 SQL 수행 결과이다. 쿼리 결과로서, JUMIN_NO, ADDRESS, MOBILE_NO 컬럼 데이터가 ' *' 처리되었음을 확인할 수 있다.

5. 비교

본 논문에서 제안하는 접근통제방식과 데이터베이스에서의 권한에 따른 데이터베이스 접근통제를 비교하였다. 비교방법은 임의의 DB 계정과, SQL 을 생성하여 보안조건과 권한구분 항목을 두고, 일정건수의 작업을 발생시키는 방식으로 진행하였으며, 비교 분석한 결과를 <표 2>에 나타내었다.

<표 2> 데이터베이스 접근통제 검출 비교

보안조건	권한구분	접근건수		기존시스템		통제시스템	
		일반컬럼	통제컬럼	일반컬럼	통제컬럼	일반컬럼	통제컬럼
일반접근	인가된 User	100	100				
	비인가된 User	100	100	100		100	100
시간대별접근	인가된 User	100	100				
	비인가된 User	100	100			100	100
DB 계정도용 접근	인가된 User	100	100			100	100
	비인가된 User	100	100			100	100
DB 계정 / IP 주소도용접근	인가된 User	100	100			100	100
	비인가된 User	100	100			100	100
통제컬럼접근 (데이터마스킹)	인가된 User	100	100				100
	비인가된 User	100	100				100

<표 2> 에서의 비교 결과를 보면, 기존시스템의 경우, 인가된 User 와 비인가된 User 의 구분없이 보안조건에 대해서 통제를 하지 못했다. 그러나, 제안된 접근통제시스템의 경우, 보안조건에 대해서 정확한 통제를 수행하였다. 특히, 시간대별접근 보안조건을 보면, 각 사용자별로 할당된 자료 접근 시간이외의 접근에 대해서 접근통제를 정확히 수행한 것을 볼 수 있다. 또한 컬럼별 통제가 불가능한 일반 질의 도구의 통제컬럼 접근인 경우에도, 데이터 마스킹이 수행된 결과를 확인할 수 있다.

6. 결론 및 향후 연구

본 논문에서는 사용자들의 다양한 데이터베이스 접근을 통제하기 위해서 네트워크상의 패킷을 모니터링 및 분석함으로써 접근통제을 구현하였다. 이와 관련하여, 데이터베이스에 접근하는 사용자 정보의 위변조를 방지하기 위해서 공개키 인증방식을 사용하였고, 클라이언트로부터 요청된 SQL 의 무결성을 확보하기 위해서 클라이언트 인증 에이전트(CAA) 와 접근통제 방화벽시스템(ACFS) 사이에, 암호화된 SQL 의 인증코드를 교환하였다. 또한, 강력한 컬럼별 접근통제정책에 따른 SQL 작성의 불편함을 최소화 시키고, 권한별 데이터 접근을 수행하기 위해서 데이터 마스킹 기법을 구현하였다. 이런 접근통제의 결과로 데이터베이스 접근에 대한 모든 로그를 남김으로써, 사후추적에 대한 정보를 제공하도록 하였다. 향후에는, 데이터베이스의 종류에 관계없이 모든 데이터베이스에 적용할 수 있는 표준적인 접근통제정책 수립과, 데이터베이스에 대한 강력한 접근통제을 완성시키기 위한 완벽한 프로토콜 분석이 필요하다. 또한, 클라이언트의 작업수행의 성능향상을 위해서, 패킷 분석과 데이터 마스킹에 대한 알고리즘의 개선 연구가 필요하다.

참고문헌

[1] 김수용, 남건우, 김상천, DB Application Firewall 과 Web Application Firewall 의 연동을 통한 불법적인 SQL 질의 차단기법, 한국정보보호학회:학술대회지 , 한국정보보호학회 03 동계학술대회 , pp.686-690 , 2003

[2] 장경옥, 구향옥, 오창석, 패킷 분석을 이용한 내부인 불법 질의 탐지. 한국 컴퓨터정보학회 논문집, 2005.7

[3] Donna Keesling, James Womack, ORACLE, Oracle9i Database Administration Fundamentals II(한글판), D11297KR20, 제품 2.0, 2002 년 8 월, D37492, Oralce Corporation, 2001, 2002

[4] Deborah Steiner, Valarie Moore, Oracle9i Net Services Administrator's Guide, Release 2 (9.2), Part No. A96580-02, Oralce Corporation, 2001, 2002

[5] Carlisle Adams, Steve Lloyd, Understanding PKI (Concepts, Standards, and Deployment Considerations), Addison-Wesley, 2002.11.06

[6] Kozierok, Charles M., Tcp/ip guide, Oreilly & Associates Inc, 2005.07

[7] Vadim V.Smirnov, http://www.ntkernel.com

[8] www.sqlparser.com