Data Base 보안과 Oracle 보안 구현

노 시 춘* · 박 상 민** · 조 성 백*** · 김 귀 남***

요 약

데이타베이스 환경에서 보안을 성취한다는 것은 위협을 확인하고 이에 대응할 수 있는 정책(policy : 해야할 일)과 메커니즘(mechanism : 달성하는 방법)을 선택하는 것이다. 본 연구에서는 데이타베이스 보안위협의 유형과 성격을 살펴보고 이같은 위협을 예방하고 차단할 수 있는 방안을 데이타베이스 보안 일반원칙으로서 제시한다. 또한, Oracle의 사례를 들어 보안성 구현방법을 제시하고 구현결과를 검증하므로서 데이타베이스 보안에 대한 일련의 절차와 프레임 워크를 구성해 본다.

Computer Database Security and Oracle Security Implementation

SiChoon Noh* · SangMin Park** · SungBaek Cho*** · Kuinam J. Kim***

ABSTRACT

Under database system environment, to accomplish database security is to ascertain the security threats and to choose the policy and mechanism of treating them. This study suggests the type and character of security threat and general method of prevention and cutting off the threats. Also, this study suggests the method of realization of Oracle security and additionally shows the method of Oracle security framework implemention. As an example, the verification method of Oracle security implementions are shown.

* 경기대 대학원 정보보호 기술공학과
** 인천대학교 공과대학 산업공학화
*** 경기대 대학원 정보보호 기술공학과

1. 서 론

데이타베이스 보안은 허가 받은 사람이 허가된 권한으로 허가된 자료취급 및 할당된 자원만을 사용하게 하고, 이 과정에서 제 3자가 그러한 과정을 엿볼 수 없게 하며, 자료유실/자료변형과 시스템 불안정을 방지하는데 목적이 있다.

그러나, 데이타베이스 관리시스템은 일반적으로 운용체제 하에서 작동되는 응용소프트웨어이기 때문에, 보안체제 또한 작동되는 시스템의 네트웍 환경과 운용체제의 보안체계와 밀접한 관련이 있다. 그리고 업무성격과 조직의 특성에 따라 데이타베이스의 분산배치가 일반화되고, 분산배치에 따른 원격접속을 위하여 공중망에 노출되는 경우가 빈번하므로, 연계성의 중요성이 더욱 요구되고 있다.

따라서, 데이타베이스 보안체계 수립은 전체 시스템, 네트웍 구성이 함께 고려되어야 하며 종합 보안프레임워크 하에서의 데이타베이스 보안대책이 강구될 때 투자비용 절감 및 업무추진의 효율화를 기대할 수 있고, 데이타베이스 관리시스템에서 제공하는 보안기능 등을 제대로 활용할 수가 있다. 그러나 일반적으로는 시스템/네트웍/응용소프트웨어 구성이 끝나고 사용 중에 데이타베이스 보안을 고려하기 때문에 많은 비용과 기술적인 문제를 야기하고 있다.

데이타베이스와 관련된 보안대책을 살펴보면, 크게 네트웍보안차원, 시스템보안차원, 사용자보안차원, 사용자암호관리차원, 사용감사차원 등 크게 6가지 분야로 분류할 수 있다. 따라서, 연구를 통해 정책적 관점에서 각 데이타베이스 보안분야을 개괄 점검하고, 세부 기술적인 설명과 지침이 필요한 사용자 인증, 사용자권한 부여, 사용감사분야를 Oracle의 경우를 사례로 보안프레임워크를 구현한다.

2. 데이타베이스 보안의 위협

위협(threat)이란 의도적이든 우연이든 시스템이 관리한는 정보를 유출하거나 수정하는 적대적인 행위하고 정의할 수 있다.

데이타베이스에서 위반(violation)이란 데이타의 부당한 검색, 수정 혹은 삭제로 이루어지는데 데이타베이스에서 위반이 일어나게 하는 사건을 위협이라고 하며 다음과 같이 세 부류로 나눌 수 있다.

- 부당한 사용자가 의도적이거나 혹은 우연히 접근하여 데이타를 판독(read)함으로써 정보의 부당한 유출(improper release of information)이 발생한다. 권한 데이타를 관찰함으로써 비권한 정보를 추론할 수 있는 비밀 위반이 이 부류에 속한다.
- 데이타의 부당한 수정(improper modification of data)의 경우로서 이것은 부당한 데이타 취급이나 수정에 의한 데이타 무결성에 관련된 모든 위반이다. 부당한 수정은 비권한 판독(read)과 반드시 관련되지 않는다. 왜냐하면 데이타를 판독하지 않고도 부당하게 변경하 수 있기 때문이다.
- 또다른 경우는 서비스 거부(denial of service)인데 사용자가 데이타를 접근하거나 자원(resource)을 이용하지 못하게 하는 행위와 관련된다.

또한, 보안위협은 그것이 발생하는 동기의 의도성 여부에 따라 우연적(non-fraudulent : accidental)위협과 의도적(fraudulent : intentional)위협으로 나눌 수 있다. 우연적 위협은 손상을 주려는 의지가 아닌 뜻밖의 사고로서 다음과 같은 유형이 있다.

- 지진이나 수해 혹은 화재와 같은 천재지변이나 우발 재해.

이런 사고는 시스템 하드웨어나 저장 데이타를 손상시킬 수 있으며 이런 사고들은 무결성 위반이나 서비스 거부를 일으킬 수 있다.

- 하드웨어나 소프트웨어에서의 오류나 버그 이것은 보안정책에 올바르지 못한 응용에 이르게 해서 데이타의 비권한 접근, 판독, 수정, 혹은 권한 있는 사용자가 접근거부를 당할 수 있다.
- 인간 오류는 응용에서 올바르지 못한 입력이나 올바르지 못한 사용과 같은 비의도적 위반을 일으키게 한다. 결론적으로 오류나 버그에 의해 발생한 것과 유사하다.

우연적 위협에 대비되는 기념으로 의도적 위협이 있다. 이는 목적물인 정보에 손상을 주려는 확실한 의지에 의해 발생하는 위반을 말하는데, 이런 위반은 두 부류의 사용자와 관련성이 있다.

- 자신의 권한을 남용할 수 있는 권한 사용 자(authorized users)
- 적대 행위자(hostile agents) 즉, 소프트웨어나 시스템 하드웨어를 파괴하려거나 데이타의 부당한 판독이나 수정을 하는 부당한 사용자(내부자이든 외부자이든)을 말한다.

두 경우에는 외형적으로 정당한 업무수행이나 정당한 이용형태를 보여 실제적인 의도를 감출 수 있다. 예를 들면, 트로이 목마(Troyjan Horse) 그리고 트랩도어(trapdoor)들은 전형적인 적대 행위자들의 공격이다.

3. 데이타베이스 보안 일반원칙

3.1 네트웍 차원의 보안

네트웍 차원에서는 데이타베이스가 탑재된 시스템에 대한 외부에서의 침입 가능성과 데이타베이스 접근과정에서 사용자계정/사용자 암호가 제 3자에 노출되지 않는 방안('snooping' 차단)이 고려되어야 한다.

- 주요 데이타베이스관련 서버 프로세스와 응용프로세스가 통신하는 TCP/IP 회선은 사내망이나 공중망이 아닌 전용회선 사용을 고려한다. 특히, 서버 프로세스를 구동시키는 Listener 프로세스 설치 포트는 이 점을 특히 유의하여야 한다.
- 데이타베이스가 설치된 시스템이 부득이 외부 공중망에 접속되어야 할 경우에는 방화벽 설치를 고려한다.
- 데이타베이스가 설치된 시스템의 TCP/IP 포트가 Dummy Hub에 연결되어 암호나 계정이 타 시스템으로 Broadcasting 될 가능성이 없는지 사전 환경조사가 필요하다.
- 데이타베이스가 설치된 시스템의 Async.포트가 외부와 연결되어 있는지를 확인하고 이에 대한 관리대책 수립이 필요하다.

3.2 시스템차원의 보안

시스템 보안차원에서는 데이타베이스 규모, 데이타베이스 사용자의 종류/특징/규모, 데이타베이스 응용서비스의 특징 및 종류 등을 고려하여야 하며, 주요사항은 다음과 같다.

- 데이타베이스의 규모가 작을 경우에는 데이타베이스 관리자를 시스템관리자가 겸임하는 것이 바람직하고, 클 경우에는 분리하여야 한다.
- 데이타베이스의 규모가 크고 데이타베이스 사용자 및 서비스 규모가 다양하고 클 경우에는, 데이타베이스 관리자도 운영관리자, 보안관리자 등 기능적으로 세분하거나, 주관리자, 부관리자 등 관리적으로 계층을 두는 것이 바람직하다.
- 데이터베이스 사용자의 인증 방법의 선택은

데이타베이스의 특성 및 산재 정도에 따라, 분산 또는 중앙에서 단일 인증관리를 할 것인지를 판단하고 인증과정의 암호화 및 세부 인증방법(절차)도 복합 검토하여야 한다.

- 데이타베이스 사용자가 해당 운용체제의 사용자일 경우, 데이타베이스의 물리적 조작권한(파일삭제, 파일수정, 파일열람)과 데이타베이스상의 조작권한을 복합검토하여 부여하고, 운용체제상의 데이타베이스 파일 조작권한을 조정하여야 한다.
  - 운용체제의 관리자(root)는 데이타베이스의 물리적 파일등을 모두 조작할 수 있고, 데이타베이스 관리자의 모든 권한을 대행할 수 있으므로 문제가 될 경우 인증서버 독립 설치 등을 고려해야 한다.
  - 데이타베이스 관리자는 데이타베이스와 관련된 운용체제상의 물리적 파일이 위치한 디렉터리의 모든 사용권한이 부여되어야 한다.
  - 데이타베이스 일반사용자는 데이타베이스와 관련된 운용체제상의 물리적 파일이 위치한 디렉터리와 파일들에 대한 접근이 운용체제상에서 제한이 가해져야 한다.

3.3 데이타차원의 보안

데이타 보안정책은 테이블과 같은 Schema Object에 대하여 어떠한 데이타베이스 사용자가 접근하여 어떠한 작업(열람, 삭제, 수정, 삽입 등)을 할 수 있는지를 설계하는 것이다. 이때, 어떠한 세부적인 데이타 보안정책을 시행할 것인지는 데이타의 민감성, 정책결정자가 시행하려는 의지적인 보안수준 및 서비스의 특성에 따라 결정되어야 한다. 왜냐하면, 세부 데이타 수준의 보안기법은 서비스 품질과 사용자 편의성을 좌우하기 때문이다. 즉, 세부 데이터(schema object별, schema object의 열별)에 대하여 보안기능을 적용할 경우 서비스 품질과 사용자 편의성이 크게 저하될 수 있다.

데이타 보안 구현방법은 크게 다음과 같이 분류할 수 있다.

- 데이타베이스 관리시스템이 제공하는 권한(privileges)과 역할(role) 관련 기능을 사용하는 방법
- 세부 데이타(schema object, schema object의 열) 생성시 'where' 절을 사용하여 보안조건을 정의하는 방법
- 응용 소프트웨어에서 데이타 수준의 보안 기능을 구현하는 방법(예 : 자료의 주요 필드를 암호화하여 입/출력하는 방법)

3.4 사용자 관리 차원의 보안

데이타베이스와 관련된 사용자는 크게 최종사용자, 데이타베이스 관리자, 응용소프트웨어 개발자 및 응용소프트웨어 관리자로 구별할 수 있고, 각 사용자관리와 관련한 보안정책 수립시 공통적으로 고려할 사항은 다음과 같다.

- 암호보안
  - 주기적으로 암호변경을 권고 또는 강제하여 불법 접근 가능성을 약화시킬 필요가 있다
  - 보다 중요한 데이타베이스에 대한 불법 접근 가능성을 차단하기 위해서는 서버와 클라이언트, 서버와 서버 사이의 연결과정을 encryption화 하여야 한다.

- 권한(Privilege) 관리
  데이타베이스 상에서 특정 작업을 할 수 있는 것을 권한(Privilege)이라하며, 그러한 권한을 여러 개 모아 놓은 것을 역할(role)이라 한다. 따라서 데이타베이스와 관련한 사용자, 응용소프트웨어 또는 객체(object)가 적을 경우는 사용자들에게 특정 권한을 일일이 명시하여 부여하는 것이 좋고, 그 규모가 클 경우에는 유사 특정 권한들을 역할로 정의하여 부여하

는 방법이 관리 효율상 바람직하다.

각 개별 사용자 그룹별 보안관리상 고려사항은 다음과 같다.

- 최종 사용자(End user)
  최종 사용자의 이용상/조직상 특성 등을 고려하여 권한을 도출/부여하여야 하며, 지역적인 분산 특성이 있을 경우, 'directory service' 개념을 사용한 중앙 집중관리도 함께 고려해야 한다.

- 관리자(Database Administrator)
  데이타베이스의 규모가 적고, 보안의 중요성이 덜 한 경우 데이타베이스 관리자를 여러명 두는 것은 관리 효율이 떨어지나, 규모가 크고 보안 중요성이 크게 요구될 경우에는 다음과 같이 관리자를 구별하는 것이 바람직하다.
  - 객체 생성 및 유지보수 관리자
  - 튜닝 및 성능담당 관리자
  - 사용자 생성 및 권한/역할 부여를 담당하는 보안전문 관리자
  - 일상적인 데이타베이스 유지보수(백업, 데이타베이스 기동)관리자
  - 비상시 데이타베이스 복구전담 관리자
  - 숙련도가 떨어지고, 경험이 더 필요한 초급 관리자

- 응용 소프트웨어 개발자(Application Developer) 응용 소프트웨어 개발자가 있는 데이타베이스를 관리할 경우는 다음 사항을 고려하여야 한다.
  - 개발자의 개발환경이 서비스 데이타베이스에 같이 존재할 경우, 무분별한 자원 사용 및 시험으로 서비스에 크게 영향을 줄 수 있기 때문에 반드시 개발자 데이타베이스 환경은 따로 설치하여야 한다.
  - 부득이 개발자의 개발환경이 서비스 데이타베이스 하에서 만들어질 경우에는 자원사용(Table space) 및 권한의 한계를 분명히 설정하여 관리하여야 한다.

- 응용 소프트웨어 관리자(Application Administrator) 데이타베이스 관련 응용소프트웨어 관리자에게는 다음과 같은 권한이 부여되어야 한다.
  - 응용 소프트웨어와 관련된 역할을 생성하고 그 역할에 속하는 권한을 관리할 수 있어야 한다.
  - 데이타베이스 응용 소프트웨어가 사용하는 객체를 생성/관리할 수 있어야 한다.
  - 응용소프트웨어 코드와 그와 관련된 데이터베이스상의 procedure/package들을 수정할 수 있어야 한다.

3.5 사용자 암호관리차원의 보안

사용자 암호는 다음과 같은 방법으로 노출의 가능성을 최소화할 수 있다

- 사용자 암호입력 오류 시 특정 기간동안 locking를 거는 방법
- 암호의 유효기간 설정
- 동일암호 재사용 금지기간 지정
- 각 계정의 암호에 대하여 복잡성 시험을 하여 재설정 유도(이 복잡성 시험은 다음의 사항을 반영함이 바람직하다)
  - 암호는 최소한 4자리 이상
  - 암호는 사용자계정과 동일하지 않아야 한다
  - 암호는 최소한 하나의 문자, 숫자, 구둣점, 특수문자를 포함하여야 한다.
  - 암호는 이전 암호와 3자리 이상 틀려야 한다.

4. Oracle 데이타베이스 보안성 구현방안

4.1 접근관리(Access Authentication)

데이타베이스 접근관리는 데이타베이스에 동시

접근 가능한 세션(session), 정의 가능한 사용자수 및 데이타베이스 사용 가능 용량을 통제하므로써 안정적인 데이타베이스 작동을 유지하고, 접근하는 사용자를 어떠한 방법으로 정당 사용자 여부를 확인하는 방법에 관한 것이다. 이러한 세부과정에서 고려할 사항을 보면 다음과 같다.

4.1.1 최대 세션수/사용자수 관리

최대 세션수/사용자계정수의 정의는 다음과 같은 초기 환경구성 파일상의 변수를 사용하여 이루어진다.

- License_max_sessions : 최대 동시 설정될 수 있는 세션수
- License_sessions_warning : 최대 세션수를 초과할 때의 경고여부 지정
- License_max_users : 최대 생성 가능한 사용자계정수 또한, 이러한 변수는 데이타베이스 가동 중에도 'Alter System' 구문을 사용하여 변경 가능하다. 최대 세션수나 사용자수를 무제한으로 할 때에는 '0'을 지정한다. 그러나, 이 변수의 값은 성능에 영향을 미칠 수 있으므로 하드웨어 시스템의 용량을 고려하여 설정하여야 한다.

  현재의 최대 세션수나 정의된 사용자수의 검색은 'V$LICENSE' 뷰(View)를 통하여 가능하다.

4.1.2 사용자 인증방법

정당한 사용자인지를 확인하는 방법으로 어떠한 구조적인 방법을 채택할 지는 데이타베이스의 배치, 규모 및 보안 민감성에 근거하여 결정하여야 한다. 특히, Oracle은 최근 여러 보안기술 중 업계 표준화된 세부기술을 복합적으로 통합한 여러 보안기능을 Package 형태로 제공하고 있다.

- 데이타베이스 인증

  사용자의 정당성을 데이타베이스 관리시스템이 관리하고 통제하는 구조이며, 이 구조는 다음과 같은 장점이 있다.
  - 사용자 관리가 데이타베이스 시스템 내에서 모두 이루어지므로 외부의존이 없고 영향을 받지 않는다.
  - 데이타베이스 관리시스템의 VERSION-UP 시 보안 기능의 향상이 자동적으로 이루어진다.
  - 자료의 민감성이 떨어지고 소규모인 데이타베이스 운용환경에 적합하다.

- 외부 인증

  사용자계정은 데이타베이스 시스템이 관리하되, 그 계정의 암호관리나 사용자 인증은 운용체제 또는 외부 다른 장(예 : Kerberos)가 대신하는 것이 외부인증 구조로 볼 수 있다. 이 방법은 다음과 같은 장점이 있다.
  - 'Smart Card', 'Fingerprints' 또는 'Kerberos'와 같은 인증기법들을 세부 기술로 적용이 가능하다.
  - 'Single Signon' 인증기법을 사용할 수 있다. 즉, 데이타베이스를 여러개 사용할 경우, 1개의 암호만을 기억하면 된다.

- 중앙관리형 인증

  데이타베이스 시스템이 여러개 분산되어 있을 경우 전체 데이타베이스 사용자의 계정 및 암호를 한 곳의 데이타베이스 시스템 또는 별도의 장비에서 관리하는 구조이다.

  이 때, 사용자는 Global 사용자로 정의될 수 있고, SSL(Secure Socket Layer)를 사용하여 인증과정의 안정성을 크게 높일 수 있다. 이 방법의 장점은 다음과 같다.
  - 관리가 용이하고 강력한 인증기법(예 : SSL) 적용할 수 있다.
  - 'Single SignOn'을 용이하게 할 수 있다
  - 데이타베이스의 규모가 클 경우에는 바람직하다.

- 다계층 인증
  클라이언트와 데이타베이스 서버사이에 응용소프트웨어 서버를 'Middle Tier'로 설치하여 클라이언트가 응용소프트웨어 서버를 통하여 데이타베이스 접근인증을 받도록 하는 구조이다.

4.1.3 사용자계정 생성관리

데이타베이스 사용자 생성작업은 그로 인하여 정의된 사용자가 데이타베이스에 접근하여 자원을 사용할 수 있고, 자원사용과정에서 전체적인 데이타베이스 운용품질 저하 및 서비스 중단을 초래하는 상황이 발생할 수 있으므로 매우 중요하다. 따라서, 사용자계정 생성권한 ('Create User')은 데이타베이스 보안관리자가 직접 관리하여야 한다. 사용자 생성시 고려 사항은 다음과 같다.

- 주기적으로 사용자 생성권한이 관리자이외의 사람에게 부여되는지를 확인
- 사용자 생성시에는 암호, 'Default tablespace', 'Temporary tablespace', 'Tablespace quota' 및 'Profile'을 같이 정의하여야 한다. 이때, 중요한 것은 사용자가 Object를 생성할 때 사용하는 'Default tablespace'와 'Sorting' 작업 등에 소요되는 'Temporary tablespace'를 어디에 어느 정도 용량으로 할당하느냐의 문제이다. 시스템의 전반적인 상황과 응용소프트웨어의 성격을 고려하지 않은 지정/할당은 서비스 중단으로 바로 이어질 수 있다.

4.2 사용자 권한/역할 관리(Privilege & Role Management)

사용자의 데이타베이스에 접근이 허용되면, 데이타베이스 내에서 여러 작업들을 수행할 수 있는데, 그러한 권한은 크게 시스템권한, 오브젝트 권한으로 다음과 같이 구별된다.

시스템 권한은 사용자로 하여금 데이타베이스와 관련된 특정 오퍼레이션을 하도록 허용해준다. 이러한 시스템 권한은 100개 이상이 있고, 작업 자체의 결과가 데이타베이스에 미치는 영향이 크므로 부여시 신중을 기하여야 한다. 이중, 특히 'any'가 포함된 시스템 권한은 특히 부여시 신중하여야 한다.

오브젝트 권한이란 사용자가 특정 오브젝트에 접근하여 작업할 수 있는 권리를 말한다.

사용자 권한은 여러 개로 그룹화하여 역할(Role)이란 이름으로 묶어 사용자에게 부여하고 경우에 따라 회수하는 작업들을 역할관리라 한다. 앞에서의 언급처럼 구체적인 권한을 사용자에게 부여할 것인지, 역할(Role)로 묶어 관리할 것인지는 데이타베이스 규모와 서비스 특성에 의해 결정해야한다.

4.3 사용감사(Auditing)

데이타베이스에 대한 접근통제, 사용자 인증 및 권한관리 등은 사전 조치적인 보안정책으로 분류된다면, 데이타베이스 사용감사는 데이타베이스에 대한 보안이 제대로 이루어지는가를 확인하고 그 결과에 따른 방책을 마련하기 위한 사후적인 과정이다. 이러한 사용감사를 수행할 때 고려해야 할 사항은 다음과 같다.

- 데이타베이스 사용감사는 운용체제의 감사 기능/기록을 이용하는 방법과 데이타베이스 사용감사 기능/자료를 이용하는 방법이 있으나, 감사의 효율성 등을 고려할 때, 데이타베이스 감사기능/기록을 이용하는 것이 바람직하다.
- 데이타베이스에 대한 사용감사는 데이타베이스 성능과 서비스 품질에 영향을 미칠 수 있으므로 무작위식 감사범위 설정/수행은 자제되어야 한다.
- 데이타베이스 감사는 그 목적이 뚜렷하여야 하고, 그 목적에 수행하고자 하는 감사 절차나 접근방법이 타당한지를 사전 평가하여야 한다. 그에 따라 감사범위와 대상을 축소해야 한다.

- 데이타베이스 감사 조건 설정시 감사 옵션(Option)은 처음에는 포괄적인 조건을 지정하여 데이타를 수집하고 그 수집된 정보에 의하여 구체적인 옵션으로 수정하는 방식이 바람직하다.
- 감사 정보 및 기록은 관련자 이외에는 접근할 수 없도록 최대한 보호가 되어야 하고 감사종료 후의 기록은 시스템 운영측면에서 삭제하여야 한다.

4.4 데이타베이스 운용 안정성 확보

데이타베이스 보안은 데이타베이스의 안정적인 운용에 그 초점이 있고 그러한 측면에서 데이타베이스의 자료 안정성 및 복구시간 단축도 보안측면에서 같이 점검되어야 한다. 또한, 대부분의 서비스 시스템이 데이타베이스를 근간으로 설계되므로 그 중요성이 매우 크다고 볼 수 있다.

데이타베이스의 자료 안정성 추구방안은 자료가 훼손되었을 때를 대비하는 것이므로 이를 위해 고려할 사항은 다음과 같다.

- 모든 데이타베이스가 탑재되는 시스템의 하드디스크는 물리적으로 이중화되어야 한다.
- 데이타베이스가 탑재된 시스템의 장애 가능성을 고려하여 데이타베이스 탑재 시스템의 물리적 이중화를 고려하고 두 시스템간 데이타베이스 일치성은 Replication이나 OPS(Oracle Parallel Server)의 도입을 검토한다. 두 자료 일치 Mechanism의 장·단점은 서비스의 특성에 의해 결정될 문제이지 어느 일방이 좋다고는 말할 수 없다.

  데이타베이스의 복구시간 단축은 서비스 특성에 따라 판단할 사항이다.
- 온라인 서비스의 데이타베이스 시스템 이중화, 주기적인 Off-Line Full Backup 및 Archive 기능을 모두 적용한다.
- Batch 서비스적인 데이타베이스 주기적인 Off-Line Full Backup 및 Archive 기능만을 적용한다.

5. 보안성 구현 검증

5.1 검증항목과 방법

데이터베이스 보안 일반원칙을 기준으로 Oracle 시스템의 보안성을 검증하는 방법을 기술한다.

검증방법은 보안취약점과 환경설정에 관하여 적용 대상별로 분류하여 시스템 구성, 접근통제, 권한관리, 사용감사 등 각 기능 분야별로 보안성 구현이 어느 수준인지를 체크하는 것이다.

일반적으로 해킹침투는 시스템의 가장 취약한 부분을 통해 이루어지는 것을 감안하면 취약가능 기능을 모두 적용하고 점검해야 하지만 이를 본 논문에서 모두 다루기는 무리이므로 우선적으로 적용되어야 할 필수부분만을 대상으로 선정한다.

5.2 보안성 구현 검증사례

이상의 검증항목 점검방법을 기준으로 실제로 보안성 구현여부 검증 사례를 제시한다. 본 사례에서 제시된 내용은 검증항목 전체내용 중 위험한 privilege 부여 여부, 사용자 접근통제, sys와 시스템 계정의 암호노출 여부 확인 등 소재에 대한 것이다.

- 위험한 privilege의 부여 여부 점검

  먼저 위험한 privilege가 부여된 지를 사용자 계정별로 검색/점검하는 내용이다.

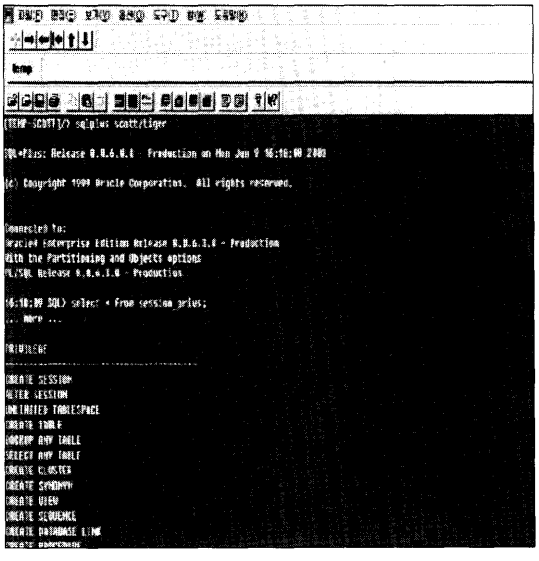
  이 기능은 시스템 전반에 영향을 미치는 시스템 권한(system privilege)이 일반 사용자계정에 부여되었는지를 검증하는 것이다. DBA는 사용자계정별로 login하여 'create user'와 같은 보안관리자(또는 시스템 관리자)만이 가져야하는 시스템 권한 및 'any'가 포함된 privilege가 일반 사용자계정에게 부여되었는지 점검한다.

〈표 1〉 DB 보안 주요 검증항목과 검증방법 사례

항목		점검항목	점검 기준/방법	
대항목	소항목			
시스템 구성		시스템 파일 권한	Oracle 시스템 파일의 UID/GUID 확인	• Oracle root 디렉토리 이하의 각종 시스템 화일의 UID/GUID가 초기 설치시와 동일한지를 확인. 초기 설치값은 /etc/passwd 파일상의 oracle(UID)와 /etc/group 파일상의 dba(GUID) 값을로 설정되어짐 초기값이 변경되어 있지 않아야 정상이다.
			Oracle 시스템 파일의 Permission 확인	• Oracle root 디렉토리 이하의 각종 시스템 파일의 permission이 other에게 'read', 'write' 권한이 부여되어 있는지 확인. 권한이 부여되지 않아야 정상이다.
접근 통제		암호 관리	Sys와 system 계정의 암호노출 여부 확인	• 쉘 상태에서 'sqlplus sys/manger' 또는 'sqlplus system' manager 명령어를 쳤을 때, sqlplus로 log-in 되는지 확인. Log-in이 되면 초기 설치시 암호값(manager)이 변경이 안된 상태이므로 바로 변경하여야 한다.
권한 관리	시스템 권한	시스템 권한	시스템관리자가 아닌 일반 사용자에게 ANY privilege가 부여되어 있는지 확인	• 일반 사용자에게는 'drop any role'과 같이 'any'가 포함된 privilege가 부여되어서는 안된다. 따라서 다음과 같은 방법으로 점검한다. Sqlplus>select * from sys.dba_sys_privs ;
			System privilege의 role이 'admin' 옵션이 부여되어 일반사용자에게 할당되었는지 확인	• system privilege의 role이 'admin' 옵션이 부여되어 사용자에게 부여되면, 그 사용자는 그 role을 타인에게 재부여할 수 있다. 그러므로 일반 사용자에게 'admin' 옵션으로 system privilege의 role을 부여하는 것은 피하여야함 따라서 다음과 같은 방법으로 점검한다. Sqlplus > select * from sys.dba_role_privs ;
			'create user'의 system privilege가 시스템 관리자 이외의 사람에게 부여되어 있는지 확인	create user 권한은 반드시 보안관리자 또는 시스템관리자만이 가져야 하는 것이므로, 다음과 같은 방법으로 확인한다. Sqlplus > select * from sys.dba_sys_privs ;
			위험한 privilege가 부여 된지를 사용자 계정별로 검색/점검	사용자계정별로 위험한 privilege가 부여되어 있는지의 여부를 각 계정별로 login 하여 다음과 같은 방법으로 점검한다. Sqlplus > select * from session_privs ;
사용 감사	감사 옵션	시스템 권한	허가되지 않은 Auditing 옵션들이 설정되어 있는지를 확인/점검	Auditing은 시스템에 부하를 줄 수 있으므로 허가되지 않은 Auditing 옵션들은 작동되어 있는지를 다음과 같은 방법으로 확인한다. Sqlplus > select * from sys.dba_stmt_audit_opts ; Sqlplus > select * from sys.dba_priv_audit_opts ;
접근 통제	시스템 부하	시스템에 부하를 줄 수 있을 정도의 사용자 접근통제	시스템 성능을 크게 상회하는 사용자 접근을 막기 위하여 시스템 parameter 파일에 다음과 같은 변수들이 설정되어 있는지를 확인하고, 정의가 되어있지 않으면 적정한 값을 설정하여 사용한다. LICENSE_MAX_SESSIONS LICENSE_SESSIONS_WARNING LICENSE_MAX_USERS 이러한 값의 현재 값을 확인하는 방법은 다음과 같다. Sqlplus > select sessions_max s_max, sessions_warning s_warning, sessions_current s_current, sessions_highwater s_high, users_max from v$license ; Sqlplus > select count(*) from dba_users ;	

〈표 1〉 DB 보안 주요 검증항목과 검증방법 사례(계속)

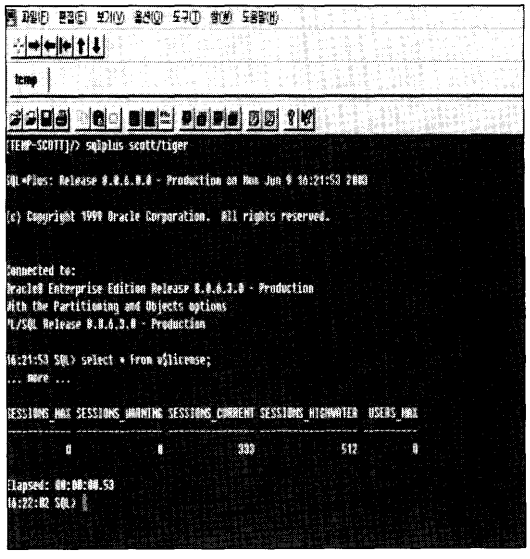
항 목		점 검 항 목	점 검 기 준 / 방 법
대항목	소항목		
접 근 통 제	자 원 사 용 통 제	모든 사용자계정의 profile 점검	사용자계정을 설정할 때는 자원사용과 관련된 profile을 지정하는데, default profile이 아닌 사용자계정의 profile은 일일이 점검하여 과도한 자원사용을 허용하지 않았는지를 확인/점검한다. Sqlplus > select username, profile, account_status From dba_users ;
		현재의 설정되어 있는 session 중에서 과도한 메모리를 사용하고 있는지를 확인/점검	각 session의 SGA 메모리 사용도를 확인하여 사전조치를 하는 것이 안정적인 시스템 운용가 특정 응용소프트웨어의 문제점을 해결할 수 있는 방법이다. 확인방법은 다음과 같다. Sqlplus > select username, value || "bytes" "Current session memory" From v$sesson sess, v$sesstat stat, v$statname name where sess.sid = stat.sid and stat.statistic # = name.statistic # and name.name = session memory ;
	암 호 관 리	각 사용자계정의 profile을 확인하여 주요 암호관련 변수의 한계치가 적정한지를 확인	사용자계정을 생성할 때 지정하는 암호와 관련된 다음과 같은 변수들의 한계값이 적정한지를 일일이 확인/점검한다. . FAILED_LOGIN_ATTEMPTS . ACCOUNT_LOCK_TIME . PASSWORD_LIFE_TIME . PASSWORD_REUSE_TIME . PASSWORD_REUSE_MAX



(그림 1) 위험한 privilege 검증

● 시스템에 부하를 줄 수 있을 정도의 사용자 접근통제.

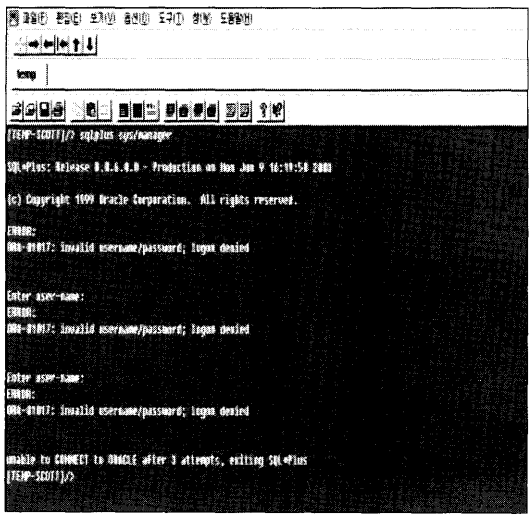
이 기능은 현재 운용 중인 시스템에 설정되어 있는, 시스템 부하에 영향을 미치는 param



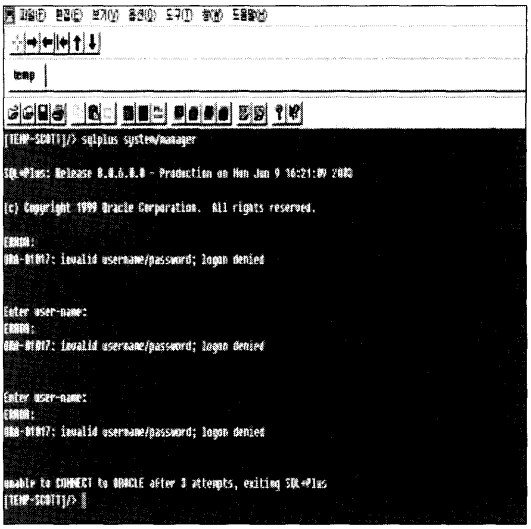
(그림 2) 시스템 부하 접근통제 검증

eter값을 점검하는 용도로 사용된다. 오라클 데이타베이스에 최대동시 설정될 수 있는 세션수(session_max)와 오라클 인스턴스에 대한 최대동시 사용 세션수(sessions_highwater), 현재 접속되어 있는 세션수(sessions_current) 등 시스템 성능을 크게 상회하는 사용자 접근을 막기 위하여 시스템 parameter 파일에 설정되어 있는 값을 확인하고 현재 설정된 값의 적정여부와 만약 값이 정의되어 있지 않으면 적정한 값을 설정하여 사용해야 한다.

- SYS와 SYSTEM 계정의 암호 노출 여부 확인
  오라클 데이터베이스 생성시 default로 생성되는 DBA User SYS와 SYSTEM 에 설정되어 있는 password가 초기 설치시 암호값(manager)을 유지하고 있는지, 또는 변경되어 있는지를 확인한다. SYS 와 SYSTEM으로 접속한 사용자는 다양한 방법으로 데이터베이스를 수정할 수 있는 강력한 권한을 가지고 있으므로, 반드시 초기 암호값을 변경해야 하며 변경 후 암호는 외부에 노출되어서는 안된다.
  아래 화면은 현재 운용 중인 오라클 RDMS에 SYS와 SYSTEM User로 login 시도시 암호값 'manager'로의 접속여부를 보여준다.



(그림 3) SYSTEM 계정 암호노출 검증



(그림 4) SYSTEM 계정 노출 검증

6. 결 론

데이타베이스에 대한 비밀성과 무결성에 대한 위협이 다양하기 때문에 데이타베이스 보호에 대한 접근 또한 체계적이고 종합적이어야 한다.

데이타베이스는 운용체제 하에서 작동되는 응용 소프트웨어로서 보안체제 또한 네트워크 환경과 운영체제 보안과 밀접한 관련이 있다. 따라서, 데이타베이스 보안체제 수립은 전체 시스템과 네트워크 구성이 함께 고려되어야 한다.

이같은 접근방법은 네트워크 차원, 시스템 차원, 데이타 차원, 사용자 보안차원, 사용자 암호관리차원, 사용 감사차원 등 6가지 분야로 분류할 수 있다.

Oracle 데이타베이스 보안성 구현은 데이타베이스 접근관리, 데이타베이스 사용자 권한/역할 관리, 데이타베이스 사용감사, 데이타베이스 운용 안정성 확보방안이 기술적으로 강구되어야 한다.

보안성 구현항목과 방법은 시스템화일 권한, 암호관리, 시스템 권한관리, 사용감사, 접근통제 분야에서 세부적인 기준이 적용된다.

이상의 보안성 구현 결과에 대한 검증은 시스템

구성분야에서 Oracle 시스템 화일의 UID/GUID 확인, Oracle 시스템 화일의 permission 확인, 시스템 권한관리 분야에서 위험한 Privilege 부여 여부 점검, 접근통제분야에서 시스템 부하관리를 위해 시스템에 부하를 줄 수 있을 정도의 사용자 접근통제 점검, 접근통제분야에서 암호관리를 위한 sys와 시스템 계정의 암호노출 여부 확인 등 소재에 관한 것들을 대상으로 시행해 볼 수 있다.

참 고 문 헌

[1] Andrews D. J. and MacEwen G., "A Review of Tools and Merhods for System Assurance", Andyne Computing Ltd, 1990.

[2] Atzeni P. and De Anthonellis V., "Relational Database Theory", Benjamin Cummings, 1993.

[3] Bell D. E., Lattice, Polics, and inplementations, In Proc. 13th Nantional Computer Security Conf., October 1990.

[4] Bonium D. A., "Logging and accountability in database management systems", In Database Security : Status and Prospects(Landwehr C. E., ed), Elservier North-Holland, IFIP, 1988.

[5] Bussolati U. and Martella G., "Data security management in distributed databases", Information systems, Vol.7, No.3, 1982.

[6] Courtney R. H., "Factors affecting the availability of security measures in data processing system components", In Pro. 13th National Computer Security Conf., October 1990.

[7] Data C. J., An Introduction to database system 5thedn., addison-wesley, 1990.

[8] Denning D. E., "a Lattice model secure informantion flow", Comm, ACM, Vol.19, No. 5, 1970.

[9] Denning D. E., "Certification of programs for secure information flow", commacm, Vol.20, No.7, 1977.

[10] Denning D. E., "Cryptography and Data Sdcurity", Addison-Wesley, 1982.

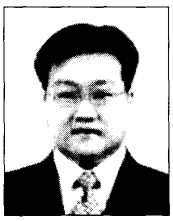
[11] Denning D. E., Scheore J., "Inference controls for statistical databases, IEEE Conputer, Vol.16, No.2 1983.

노 시 춘

1992년 고려대학교 경영대학원 경영정보학과석사
2003년 경기대학교 대학원 정보보호기술공학과 박사과정
1980년~현재 KT IT본부 충청전산국 국장

박 상 민

1970년 한양대학교 산업공학
1983년 한양대학교 산업공학 석사
1990년 한양대학교 산업공학 박사
현재 인천대학교 공과대학 산업공학과 교수



조 성 백

1994년 한국과학기술원 산업공학과(공학사)
1998년 런던대학교 정보보호학과(공학석사)
2003년 런던대학교 정보보호학과(공학박사)
2003년~현재 경기대학교 정보보호기술공학과 연구교수

김 귀 남

미국 캔자스대학 수학과(응용수학사)
미국 콜로라도주립대학 통계학과(통계학석사)
미국 콜로라도주립대학 기계·산업공학과(기계·산업공학과박사)
현재 경기대학교 정보보호기술 공학과 주임교수