ORACLE 9*i* APPLICATION SERVER VERSION 2의 보안 기능

John Heimann, Oracle

머리말

이 백서는 Oracle9*i* Application Server(Oracle9*i*AS) version 2의 보안 기능을 설명하고 있습니다. 우선 웹 보안 요건의 개요를 설명하고 이 요건에 부응하기 위해 Oracle9*i*AS에 구현된 보안 아키텍처를 다루고 있습니다. 이 백서는 Oracle9*i*AS에 도입된 새로운 보안 기능에 특히 중점을 두고 있습니다.

웹 애플리케이션을 위한 보안 기능

웹은 기업이 그 사용자에게 비즈니스 애플리케이션과 데이타에 대한 액세스를 제공하는 데 선호했던 방식인 클라이언트-서버를 대체했습니다. 여기에는 뛰어난 확장성, 비용 절감, 새로운 사용자와 시장에 접근할 수 있는 기회 등 널리 인식된 여러 이유들이 있습니다. 인터넷에 비즈니스 애플리케이션을 배포하는 데 따르는 위험성 역시 널리 인식되어 있습니다. 이러한 위험 요소에는 다음과 같은 것들이 있습니다.

- 사용자의 신분에 대한 제한된 정보
- 사용자의 행동을 제어할 방법이 거의 또는 전혀 없다는 점(사용자가 누구인지 모르거나 기업 구성원이 아닌 경우에는 잘못된 행동에 처벌을 가하기 어렵습니다.)
- 악의적인 사용자들에게 시스템과 데이타가 크게 노출된다는 점
- 보안이 취약한 네트워크에서 정보의 노출 또는 변조 문제
- (worm), 크로스 사이트 스크립팅 등 인터넷의 특정한 개방적 특성을 악용하는 경우

웹 애플리케이션 개발자들은 지금까지 몇 년 동안 이 같은 위험 요소들과 씨름 해야 했습니다. 단일 정보 포탈을 통해 여러 애플리케이션을 배포하는 것, Java를 사용한 웹 애플리케이션 배포 증대, 배포된 애플리케이션의 복잡성과 규모(서비스 대상 사용자 수)의 증대 등 최근의 몇몇 경향들은 웹 애플리케이션 보안을 더욱 복잡하게 하고 있습니다.

ORACLE 9*i* AS의 보안 기능 개요

Oracle9*i*AS version 1은 광범위한 애플리케이션을 지원하는 Oracle9*i*AS Single Sign-On(SSO)을 도입했으며 여러 구성 요소, 특히 Oracle HTTP Server와 Oracle9*i*AS Portal에 보안 서비스를 포함시켰습니다. Oracle9*i*AS version 2에서 Oracle은 모든 Oracle9*i*AS 구성 요소들 뿐만 아니라 Oracle9*i*AS에 배포된 써드 파티 및 맞춤형 애플리케이션들도 지원하는 포괄적인 보안 프레임워크를 도입했습니다. 이 프레임워크는 인증의 경우 Oracle9*i*AS SSO에, 인증 및 사용자 설정(provisioning)의 경우 Oracle Internet Directory에, Java2 Enterprise Edition(J2EE) 애플리케이션 보안의 경우 Oracle Java Authentication 및 Authorization Service(JAAS) 제공자를 기반으로 하고 있습니다.

ORACLE 9*i* AS의 싱글 사인온 기능

Oracle9*i*AS에서 중요한 보안 기능의 하나는 웹 기반 애플리케이션에 SSO(Single Sign-On) 기능을 지원한다는 점입니다. 기업들이 SSO를 고려하는 데에는 몇 가지 이유가 있는데 그 중 하나는 직원, 고객 및 파트너들이 사용할 수 있도록 기업들이 배포하는 웹 기반 eBusiness 애플리케이션의 사용이 증가하고 있다는 점입니다. SSO가 없으면 각 사용자는 자신들이 액세스하는 각 애플리케이션마다 별도의 ID와 암호를 보유해야 합니다. 각 사용자마다 여러 개의 계정과 암호를 보유하는 것은 보안에 좋지 않으며 비용도 많이 듭니다.

여러 계정과 암호는 보안에 불리

대부분의 사용자들은 몇 개의 암호 이상은 기억하지 못합니다. 하나 이상의 로그인 계정을 갖고 있는 사용자들은 기억하기 쉬운 암호를 사용하거나, 여러 계정에 동일한 암호를 사용하거나, 암호 변경을 요구 받았을 때 기존의 암호를 재사용하거나, 암호를 따로 적어둡니다. 이러한 관행들은 모두 암호의 보안성을 떨어뜨립니다. 암호를 적어두거나 쉽게 기억할 수 있는 암호를 사용하는 것은 암호가 누출될 위험성을 높입니다. 암호를 변경할 때 기존 번호를 재사용하거나 여러 시스템에 같은 암호를 사용하면 암호 하나가 노출되었을 때 피해 범위가 확대됩니다. 많은 시스템들이 사용자가 복잡한 암호를 사용하도록 강제하거나 암호를 재사용하지 못하게 하는 암호 관리 처리 방법을 구현하고 있지만 이러한 처리 방법들이 역효과를 일으키는 경우가 많습니다.

사용자들이 이를 무력화하는 방법을 궁리해내므로 보안성을 오히려 더 떨어뜨릴 수 있습니다. 예를 들어, 사용자들에게 무작위 암호를 사용하도록 강제하는 것은 오히려 암호를 적어 두도록 장려하는 것이나 마찬가지일 뿐입니다. 사용자가 어떤 기업에 들어 오거나 나가는 경우, 또는 기업 내에서 역할을 변경하는 경우, 여러 애플리케이션에 액세스할 때 그 사용자가 갖고 있는 권한들이 기업 변화에 의해 지원됩니다. 사용자마다 여러 개의 독립적 계정을 보유하게 되면 그에 연관된 사용자 권한이 기업 변화를 따라가지 못하는 경우가 많습니다. 예를 들어, 어떤 사용자가 기업을 떠나거나 역할을 변경한 이후에도 오랫동안 해당 시스템에 그 사용자의 계정과 액세스 권한이 남아 있을 수 있습니다. 그러면 이 시스템은 불만을 가진 전직 직원의 공격 가능성에 노출되는 것입니다.

여러 개의 암호는 많은 비용을 요구

여러 계정과 암호를 보유하는 것은 많은 비용이 요구됩니다. 규모가 큰 기업들에서는 시스템 관리자의 근무 시간 중 상당 부분이, 사용자가 해당 기업에 들어 올 때 사용자 계정을 처음 만드는 작업, 이들이 떠나거나 역할을 변경하는 경우 계정을 삭제하는 작업, 암호를 잊어버리는 경우 재설정하는 작업 등 계정과 암호에 관련된 작업에 지출되고 있습니다. 사용자마다 여러 계정을 보유하게 되면 그와 관련하여 시스템 관리자가 부담해야 할 요구 사항이 훨씬 더 늘어나게 됩니다.

시스템 관리자가 처리해야 할 문제들 가운데에는 여러 시스템에 사용자 계정을 추가하거나 삭제하기 위해 각각의 시스템에 각각 다른 여러 관리 인터페이스를 이용하여 액세스해야 한다는 문제도 있습니다.

ORACLE9*i*AS SSO의 솔루션

Oracle web SSO 기술은 웹 사용자에게 싱글 사인온 기능을 제공합니다. 이 기능은 Application Server를 통해 여러 웹 기반 애플리케이션에 액세스할 수 있는, Oracle9*i*AS가 제공하는 바와 같은 환경에서 동작하도록 설계되었습니다. SSO를 위한 Oracle의 전략에는 다양한 기술들이 포함되어 있습니다. 점점 확대되는 웹 기반 애플리케이션 분야를 위해 Oracle은 SSO 프레임워크를 개발했으며, 또한 웹 SSO를 제공하기 위해 특별히 설계된 Oracle9*i*AS SSO Server를 개발했습니다.

Oracle의 SSO 방식에는 여러 가지 장점이 있습니다. Oracle의 방식은 표준 프로토콜을 통해 브라우저 클라이언트에서부터 Oracle의 애플리케이션과 도구 등의 웹 기반 애플리케이션에 이르기까지 안전한 SSO를 위한 프레임워크를 제공합니다. Oracle의 SSO는 SSO 프레임워크를 십분 활용하는 파트너 애플리케이션을 지원할 뿐만 아니라 기존 제품 및 써드 파티 제품을 지원할 수 있도록 외부 애플리케이션들도 함께 지원합니다. 파트너 애플리케이션들은 SSO 프레임워크 내부에서 동작하며 SSO 서비스에 의존하여 사용자를 인증합니다. 외부 애플리케이션들은 여전히 각자 고유의 사용자명과 암호를 이용합니다. Oracle9*i*AS의 SSO 접근 방식은 Oracle9*i*AS SSO Server라고 불리는 집중화된 서버와 파트너 애플리케이션들이 만드는 쿠키에 바탕하고 있습니다.

Unocal은 Oracle9*i*AS를 사용하는 Oracle의 주요 고객입니다. Unocal은 자사의 회사 정보 포탈 myUnocal.com에 싱글 사인온 기능을 제공하기 위해 Oracle9*i*AS SSO를 선택했으며 이 표준을 전세계에 구현하는 작업을 진행 중입니다. myUnocal은 비즈니스 애플리케이션 서비스를 전세계의 Unocal 직원들에게 제공하고 있으며 비즈니스 데이타와 서비스를 통합하려는 Unocal의 목표를 지원합니다. Oracle9*i*AS SSO를 통해 myUnocal 직원들은 각 애플리케이션마다 별도의 사용자명과 암호를 기억할 필요 없이 단 한번의 인증으로 자신에게 허용된 애플리케이션들에 액세스할 수 있게 될 것입니다.

구성 요소

ORACLE9iAS SSO SERVER

Oracle SSO 기술의 핵심은 Oracle9*i*AS SSO Server입니다. 이 기술은 원래 Oracle9*i*ASv1에 Oracle9*i*AS Portal의 구성 요소로서 처음 도입되었으나 Oracle9*i*AS에서는 SSO Server가 인프라 구성 요소이며 Portal을 설치할 필요가 없습니다.

Oracle9*i*AS SSO Server는 사용자를 인증하고 이들의 ID를 파트너 애플리케이션으로 안전하게 전달합니다. 미리 설정된 기간(보통 하루) 중 해당 시스템에 처음 액세스하는 경우 사용자에게 사용자명과 암호 입력을 요구하며, 사용자가 입력한 암호를 확인합니다.
Oracle9*i*AS SSO Server는 웹 서버가 브라우저 클라이언트에 저장한 일정 형식의 정보인 쿠키를 사용합니다.

쿠키를 통해 웹 서버는 클라이언트 사용자에 관한 정보를 저장하고 불러 올 수 있으며, 사용자를 구별할 수 없는 웹 환경에서 사용자를 구별할 수 있는 정보를 효과적으로 관리할 수 있습니다.
비록 사용자가 그 기능을 꺼놓을 수 있지만 쿠키 기능은 현재의 모든 브라우저에서 지원되고 있습니다. 쿠키 기능을 꺼놓은 경우에는 Oracle9*i*AS SSO Server가 SSO를 지원하지 않습니다. 쿠키는 클라이언트의 하드 디스크에 저장되기 때문에 브라우저를 종료하더라도 계속 유지됩니다. 브라우저를 종료할 때 쿠키를 삭제하는 경우에는 쿠키 기능이 유지되지 않습니다. 사용자가 Oracle9*i*AS SSO Server에 인증되면 이 서버는 SSO 쿠키를 사용자의 브라우저에 설치합니다. 그 후 브라우저가 Oracle9*i*AS SSO Server에 액세스하면 유효한 SSO 쿠키 정보의 존재 유무로 사용자의 인증을 확인하는 것입니다.

파트너 애플리케이션

파트너 애플리케이션이란 SSO 프레임워크 내부에서 동작하는 애플리케이션을 말합니다. 특히 이들은 사용자 인증에 대한 책임을 Oracle9iAS SSO Server에 위임하기 위해 설계(또는 수정)됩니다. 이들은 Oracle9*i*AS SSO Server가 제시한 사용자 ID를 받아들입니다.
Oracle9*i*AS SSO의 인증 서비스를 활용하기 때문에 파트너 애플리케이션은 자체 인증 모듈을 구현할 필요가 없습니다. 파트너 애플리케이션의 경우 암호를 관리할 필요가 없으므로 사용자 관리 작업이 단순화됩니다. 따라서 어떤 애플리케이션을 파트너 애플리케이션으로 배포하면 배포 비용과 관리 비용을 줄일 수 있습니다.

MOD_OSSO

Mod_osso는 Oracle9*i*AS에 도입된 새로운 기능입니다. 이 기능은 Oracle HTTP Server가 SSO 파트너 애플리케이션이 될 수 있도록 해주는 확장 기능입니다. 서블릿처럼 HTTP Server 아래에서 실행되는 애플리케이션들은 사용자의 인증된 ID를 mod_osso로부터 Apache 헤더 형태로 받게 됩니다. 따라서 Mod_osso는 애플리케이션들이 특정한 파트너 애플리케이션 로직을 내장할 필요 없이 Oracle9*i*AS SSO 프레임워크에 참여할 수 있도록 합니다. 이는 Oracle9*i*AS에서 실행되는 애플리케이션들이 Oracle9*i*AS SSO에 참여하는 데 권장하는 방식입니다.

외부 애플리케이션

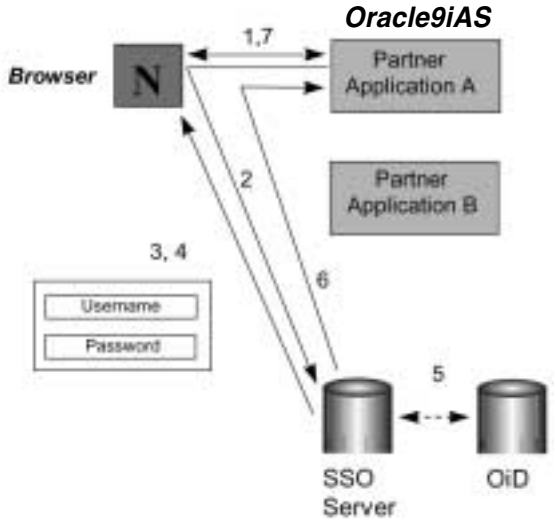
외부 애플리케이션들은 사용자명과 암호 입력에 각각 서로 다른 웹 형식(web form)을 사용하므로 외부 애플리케이션을 지원하려면 Oracle9*i*AS SSO Server를 특정 애플리케이션에 맞게 사용자 정의해야 합니다. 사용자 또는 시스템 관리자는 사용자명과 암호를 설치 또는 변경할 때 특수한 등록 페이지에 이를 입력하는 등 몇몇 특정한 조치를 수행해야 할 수도 있습니다. 이 같은 등록 페이지를 설정하려면 Oracle9*i*AS SSO Server 내부에 해당 애플리케이션에 맞는 기능을 개발해 넣어야 할 수도 있습니다.
mod_osso 또는 파트너 애플리케이션으로 애플리케이션을 배포하는 것이 더 좋은 방법이지만 기존 애플리케이션을 개선하는 것이 현실적으로 어려운 경우가 있습니다. Oracle9*i*AS SSO Server가 외부 애플리케이션을 지원하는 것은 이 때문입니다. Oracle9*i*AS SSO Server에서 파트너 애플리케이션과 외부 애플리케이션을 모두 지원하면

시스템 통합업체들이 최대의 유연성을 확보하게 되는데 이는 하나의 SSO 프레임워크에서 최신 애플리케이션 뿐만 아니라 기존 웹 애플리케이션까지 지원할 수 있게 되기 때문입니다.

기능 개요

최초 인증

어떤 사용자가 파트너 애플리케이션에 처음으로 액세스를 시도하면 파트너 애플리케이션이 이 사용자를 Oracle9*i*AS SSO Server로 재전환해 줍니다. Oracle9*i*AS SSO Server는 이 사용자가 유효한 SSO 쿠키 설정을 갖고 있는지 확인하는데, 유효한 쿠키 설정이 없는 경우에는 사용자명과 암호를 입력하여 인증 받을 것을 요구합니다. 사용자가 사용자명과 암호를 입력하면 Oracle9*i*AS SSO Server는 암호를 확인하고 사용자의 브라우저에 SSO 쿠키를 설정합니다. 이 쿠키는 이후 이어지는 Oracle9*i*AS SSO Server와의 HTTP 상호 작용에서 Oracle9*i*AS SSO Server에 해당 클라이언트를 인증하는 데 사용됩니다.
SSO 쿠키는 Oracle9*i*AS SSO Server에 의해 암호화되므로 써드 파티는 이를 읽거나 설정할 수 없습니다. 쿠키는 관리자가 설정한 특정 기간(일반적으로 8시간) 이후, 또는 사용자가 자신의 브라우저를 종료한 경우에 만료됩니다. SSO 쿠키가 파트너 쿠키와 구별되는 점은 브라우저를 종료하면 계속 유지되지 않고 삭제된다는 점입니다.



1. 사용자가 파트너 애플리케이션 A에 액세스하면 파트너 애플리케이션이 사용자의 인증 여부(애플리케이션 A 쿠키 유무)를 확인합니다.
2. 애플리케이션 A가 사용자를 SSO Server로 재전환합니다.
3. SSO Server가 사용자명과 암호 페이지를 표시합니다.
4. SSO Server가 암호를 확인하고 SSO Server로의 인증을 위해 SSO 쿠키를 설정합니다.
5. SSO 증명서가 외부 서버에 저장될 수 있습니다.(향후)
6. SSO Server는 사용자를 해당 애플리케이션에 인증 해주는 암호화된 토큰과 함께 사용자를 파트너 애플리케이션으로 전환합니다.
7. 파트너 애플리케이션 A가 애플리케이션 A 쿠키를 설정합니다.

그림 1:Oracle9*i* AS SSO Server와 파트너 애플리케이션으로의 인증

Oracle9*i*AS SSO Server, 브라우저 클라이언트, 애플리케이션 사이의 상호 작용이 모두 표준 HTTP를 통해 이루어집니다. 쿠키 지원 외에는 클라이언트에 다른 특별한 요구 사항은 없습니다. 써드 파티가 사용자명과 암호, SSO 쿠키 등을 가로채는 사태를 방지하려면 Oracle9*i*AS SSO Server와 클라이언트 사이에 SSL 기능

을 켜 놓는 것이 좋습니다. 써드 파티가 이 정보들을 가로채게 되면 이를 이용하여 Oracle9*i*AS SSO Server를 속일 수 있습니다.

파트너 애플리케이션으로의 인증

사용자 인증이 이루어지고 SSO 쿠키가 설정되면, Oracle9*i*AS SSO Server는 사용자를 파트너 애플리케이션으로 돌려 보내고, 사용자의 ID를 가지고 있는 암호화된 토큰을 파트너 애플리케이션 URL에 포함시킵니다. 토큰은 Oracle9*i*AS SSO Server와 파트너 애플리케이션만이 공유하는 키로 암호화됩니다. 이 키는 해당 토큰이 가짜가 아니며 Oracle9*i*AS SSO Server에 의해 생성되었음을 파트너 애플리케이션에 확인시켜 줍니다.
파트너 애플리케이션이 URL 토큰을 받아서 해독하면, 파트너 애플리케이션은 애플리케이션에 대한 인증 받은 사용자의 액세스에 권한 부여를 할 것인지 여부를 결정합니다. 액세스에 권한 부여를 하기 위해, 파트너 애플리케이션은 사용자의 브라우저에 있는 파트너 애플리케이션 쿠키를 설정합니다. 파트너 애플리케이션 쿠키를 통해 애플리케이션은 인증을 위해 사용자를 Oracle9*i*AS SSO Server로 재지정할 필요 없이 클라이언트 사용자에 대한 액세스를 확인하고 권한 부여를 할 수 있습니다. 파트너 애플리케이션 쿠키는 SSO 쿠키와 마찬가지로 특정 기간 이후에는 만료되지만, SSO 쿠키와는 달리 지속적일 수도 비지속적일 수도 있습니다. 즉, 파트너 쿠키는 브라우저 종료 후에 존속할 수도 있고 존속하지 않을 수도 있습니다. 파트너 애플리케이션 쿠키의 만료 기간은 해당 애플리케이션에서 결정되며, SSO 쿠키 만료 시간과는 다를 수 있습니다.
SSO 쿠키의 경우처럼, SSL 암호화는 브라우저와 파트너 애플리케이션간의 쿠키 교환을 보호하는 데 사용되는 것이 바람직합니다.

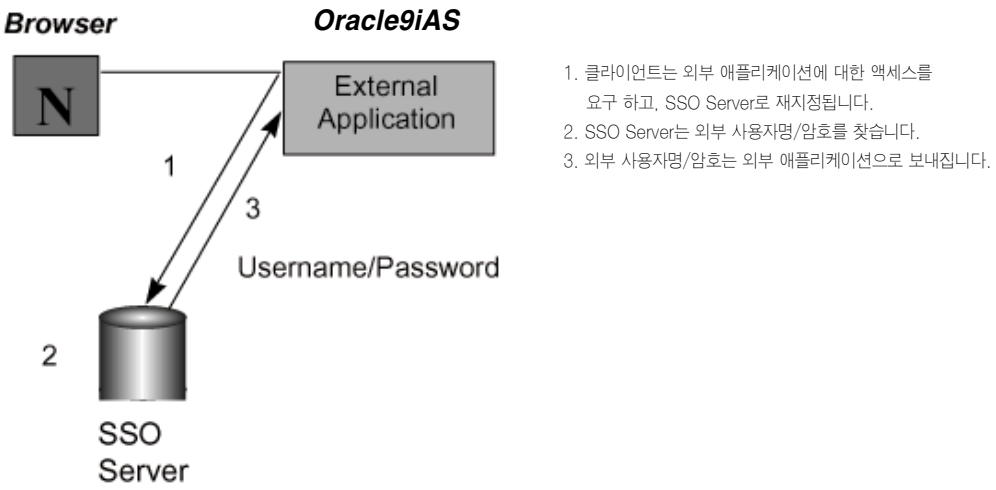


그림 1: **외부 애플리케이션에 대한 인증**

외부 애플리케이션에 대한 인증

외부 애플리케이션은 Oracle9*i*AS SSO Server로부터 직접 인증 ID를 받을 수 없습니다. Oracle9*i*AS SSO Server는 웹 폼을 통해 인증을 지원하는 외부 애플리케이션에 SSO를 제공합니다. Oracle9*i*AS SSO Server는 안전한 암호 저장 처리 방법을 통해 외부 애플리케이션에 SSO를 제공합니다. 암호 저장은 Oracle9*i*AS SSO Server내의 테이블에 특정 애플리케이션 사용자명과 암호를 유지 관리합니다. 이 테이블에 대한 액세스는 Oracle9*i*AS SSO Server가 제한하고, 암호는 암호화를 통해 한층 더 보호됩니다. Oracle9*i*AS SSO Server가 인증한 사용자가 외부 애플리케이션에 대해 액세스할 때, Oracle9*i*AS SSO Server는 특정 애플리케이션에 대한 사용자의 사용자명과 암호를 암호 저장에서 검색하여, 적절한 웹 폼으로 생성한 뒤, 애플리케이션에 제출합니다. 이러한 과정은 사용자에게 투명하게 행해집니다.

Oracle9*i*AS SSO Server와 외부 애플리케이션간의 SSL 암호화는 네트워크상에서 애플리케이션 암호가 노출되는 것을 막는 데 사용됩니다.

LDAP 통합

LDAP(Lightweight Directory Access Protocol)를 지원하는 디렉토리는 사용자에 대한 기업 규모의 정보에 대한 단일 소스로서 점점 사용이 늘고 있습니다. 이 디렉토리들은 기업에서 여러 애플리케이션이나 서버를 사용하는 사용자들을 준비(생성 및 구성) 및 관리하기 편리한 처리 방법을 제공합니다. 이는 LDAP가 널리 지원되는 인터넷 표준 프로토콜이기 때문이고, LDAP 디렉토리가 브라우저에 대한 편리한 단일 소스로서 사용되기 때문입니다.

사용자에 대한 사용자명/암호 정보는 기업 전체에서 액세스할 수 있습니다. OID(Oracle Internet Directory)는 특별히 이러한 유형의 애플리케이션에 적합한데, OID가 안전하고, 확장성이 있으며, 수행적이고, 매우 가용성이 좋은 디렉토리 서비스를 제공하기 때문입니다. (OID에 관한 자세한 정보는 "OID를 통한 디렉토리 허용 보안" 섹션을 참조하십시오.)

Oracle Oracle9*i*AS SSO Server는 OID를 사용하여 SSO 사용자명과 암호를 확인합니다. 초기 인증 부분으로 사용자가 SSO 사용자명과 암호를 제출하면, Oracle9*i*AS SSO Server는 사용자명 및 암호를 OID에 있는 사용자명 및 암호와 비교합니다. 비교 결과가 일치할 경우, SSO 사용자명과 암호는 확인된 것으로 간주됩니다. Oracle9*i*ASv1에서는, OID에서 사용자 ID를 유지 관리하는 것이 옵션이지만, Oracle9*i*AS에서는 기본값입니다. OID에서 사용자명과 암호를 관리하는 것은 구성 요소에 대한 사용자 정보를 집중된 단일 표준 LDAP 저장소에서 관리한다는 전반적인 Oracle9*i* 플랫폼(데이타베이스와 애플리케이션 서버) 전략과 일치하는 것입니다.

Oracle9*i*AS에서, OID는 Password Verifier API를 통해 폭 넓은 인증을 제공합니다. Password Verifier API를 통해 OID는 사용자 인증 데이타(〈username〉/〈password verifier〉폼)를 받아서 사용자 정의나 써드 파티 인증 처리 방법을 사용하여 유효성을 검사합니다. Oracle9*i*AS SSO는 인증 데이타의 유효성을 검사하는 데 있어 OID에 의존하므로, OID Password Verifier는 Oracle9*i*AS SSO에 폭 넓은 인증을 제공하여 포괄적인 인증 기술을 지원할 수 있게 합니다

써드 파티 통합

Oracle9*i*AS SSO는 써드 파티 인증 통합과 싱글 사인온(single sign-on)에 대한 API를 제공합니다. 이 기능은 Oracle9*i*ASv1.0.2.2에서 소개되었습니다. API를 통해 SSO는 신뢰 받은 외부 인증 처리 방법으로부터 사용자 ID를 얻어 구성될 수 있고, Siteminder(r) from Netegrity, Inc.와 같은 써드 파티 제품이 제공하는 SSO 프레임워크로 Oracle9*i*AS를 통합할 수 있습니다.

PKI 지원

PKI 인증은 많은 애플리케이션에서 암호를 대체하기 시작했습니다. 웹 기반 애플리케이션에서, PKI 인증은 SSL(Secure Sockets Layer) 세션 확립 부분으로 일반적으로 X.509 인증서 교환을 통해 수행됩니다. PKI는 그 자체로 SSO 제공에 사용되는데, 인증서를 가진 사용자는 암호를 입력하지 않고 여러 애플리케이션에 대해 인증 받을 수 있기 때문입니다.

Oracle9*i*AS에서, 사용자는 PKI를 통해 Oracle9*i*AS SSO Server에 대해 인증 받을 수 있습니다. 이는 Oracle9*i*AS SSO Server가 지원하는 웹 기반 애플리케이션과 그 밖의 PKI 허용 애플리케이션에 SSO를 제공할 것입니다. SSO 사용자명과 암호를 제공하는 대신, 사용자는 SSL을 통해 클라이언트 및 서버 X.509 인증서 교환으로 Oracle9*i*AS HTTP Server에 대해 인증 받게 됩니다. Oracle9*i*AS SSO Server는 HTTP Server로부터 사용자의 SSL 유효 인증서를 얻어 OID(Oracle Internet Directory)에서 이 인증서를 찾습니다. 사용자를 찾게 되면, OID는 사용자의 SSO ID를 Oracle9*i*AS SSO Server로 반환합니다. 파트너 및 외부 애플리케이션에 대한 인증은 앞에서 설명한 쿠키 기반 방법을 사용하여, Oracle9*i*AS SSO Server가 수행합니다. 이 방법의 이점은 Oracle9*i*AS SSO Server가 PKI를 허용할 때, Oracle9*i*AS SSO Server 프레임워크에서 작업하는 애플리케이션이 자동으로 PKI를 허용하게 된다는 것입니다. Oracle9*i*AS SSO Server와 OID는 이름 매핑에 대한 책임을 맡게 됩니다. 더욱이, 쿠키 획득 및 확인은 SSL 교환을 수행하는 것보다 프로세싱 집중도가 훨씬 덜하기 때문에, SSO 프레임워크에 대한 초기 인증에는 PKI를 사용하고, 파트너 애플리케이션에 대한 인증에는 쿠키를 사용하면 PKI만 사용하여 인증하는 방법보다 성능이 틀림없이 더 좋습니다. 많은 단기 세션이 파악하는 웹 애플리케이션에 있어서, 이것은 서버 성능과 처리 능력에 상당한 개선 효과를 줍니다.

결국, PKI 허용 Oracle9*i*AS SSO Server는 PKI를 사용하여 사용자가 Oracle Applications에 대해 인증 받을 수 있게 하는 Oracle 전략입니다. Oracle Applications는 Oracle9*i*AS Portal에 있는 Application Portlet을 통해 SSO 프레임워크에 참여하므로, Oracle9*i*AS SSO에서의 PKI에 대한 지원은 Oracle Applications에 대한 PKI 인증을 허용합니다.

그 밖의 보안 개선점

Oracle9*i*AS에서, SSO에는 SSO 솔루션의 유연성과 보안을 향상시킨 많은 개선점이 들어 있습니다. 이러한 개선점으로는 싱글 사인오프(sign-off), 패러노이드(paranoid) 애플리케이션 지원, 글로벌 비활성 탐지가 있습니다.

싱글 사인오프(single sign-off)

싱글 사인오프를 통해 사용자는 SSO 세션을 종료할 뿐만 아니라, 파트너 애플리케이션 세션도 종료할 수 있습니다. 파트너 애플리케이션 시간 초과 기간이 SSO 세션보다 길어질 수 있기 때문에 이 기능은 중요합니다. 따라서 파트너 애플리케이션 쿠키가 SSO 쿠키보다 유효 기간이 더 길면 사용자가 유효한 SSO 세션을 가지는 것이 아니라(SSO 세션 쿠키가 시간 초과되기 때문), 계속해서 유효한 파트너 애플리케이션 세션을 가지게 될 수도 있습니다. 싱글 사인오프를 통해 사용자는 한번의 조치(예: 그 날 집으로 갈 때)로 SSO 세션과 모든 파트너 애플리케이션에서 로그 아웃할 수 있습니다.

패러노이드(paranoid) 애플리케이션 지원

패러노이드 애플리케이션 지원을 통해 파트너 애플리케이션은 SSO 서버가 SSO 쿠키가 여전히 유효한지의 여부에 대해 사용자를 재인증하도록 할 수 있습니다. 이 기능이 소개되기 전에는, 파트너 애플리케이션 세션이 시간 초과되고 SSO 세션은 여전히 유효할 경우, 파트너 애플리케이션은 사용자를 SSO 서버로 재지정해야 했지만, SSO 서버는 사용자를 재인증할 필요 없이 간단히 사용자의 ID만 반환하면 됩니다. 패러노이드 애플리케이션 기능을 통해 특별히 민감한("패러노이드") 애플리케이션은 SSO 서버가 요구하는 것보다 더욱 빈번하게 재인증을 요구할 수 있습니다. 또한 이벤트 방식 인증을 허용하여, 애플리케이션에서 몇 가지 특별히 민감한 행동을 수행할 때 사용자가 재인증 받게 할 수 있습니다.

글로벌 비활성 탐지

파트너 애플리케이션은 앞에서 설명한 패러노이드 애플리케이션 기능을 사용하여 비활성을 기준으로 세션을 시간 초과할 수 있습니다. 또한 9*i*AS에서 글로벌 비활성 탐지를 구성하여, 특정 기간 동안 파트너 애플리케이션을 사용하지 못하면 SSO 세션은 시간 초과될 수 있습니다. 이 기능은 사용자가 주어진 기간 동안 비활성적이면 사용자 재인증을 요구하는 보안 정책을 구현하는 데 사용됩니다.

3-Tier 통합

Oracle9*i*AS SSO Server는 웹 서버에 대한 웹 클라이언트 액세스에 대해 SSO를 제공합니다. 웹 서버는 백 엔드 계층 데이타베이스에 대해 액세스를 제공하는 3-Tier 아키텍처에서 중간 계층으로서 배치되는 경우가 점점 많아지고 있습니다. 데이타베이스에 대한 액세스를 요구하는 웹 애플리케이션 사용자는 데이타베이스에 저장되어 있는 데이타에 대한 액세스에 대해 데이타베이스 사용자명과 암호를 제공할 필요는 없습니다. Oracle9*i*AS SSO Server가 비웹 기반 애플리케이션을 지원하지 않더라도, Oracle9*i* 데이타베이스는 3-Tier 아키텍처를 통해 데이타베이스에 대한 안전한 액세스를 지원하도록 특별히 설계된 기능을 가지고 있습니다.

SSO 요약

Oracle9*i*AS SSO는 여러 웹 애플리케이션을 배치할 때 매우 유용한 인증 프레임워크를 제공합니다. SSO는 사용자 경험을 개선하고, 애플리케이션이 자체 인증 처리 방법을 필요로 하지 않으므로 애플리케이션 개발 비용을

줄일 수 있으며, 시스템 관리 비용을 절감하고, 여러 암호와 관련된 문제를 해결했으므로 시스템 보안을 개선합니다.

OID를 통한 디렉토리 활용 보안

Oracle은 기업에서 사용자 및 서비스에 대한 엔터프라이즈 정보를 관리하기 위해 Oracle 제품에 대한 공동 처리 방법으로 LDAP를 표준화했습니다. 이를 지원하기 위해 Oracle은 Oracle9*i*의 입증된 데이타베이스 기술을 기반으로 확장성, 안정성, 보안성 높은 LDAPv3 호환 디렉토리인 OID(Oracle Internet Directory)를 개발했습니다. OID는 Oracle 제품에 LDAP 디렉토리 서비스를 제공하고, Oracle 제품은 OID에 대해 해당 LDAP 구현을 보장합니다.

디렉토리 권한 부여

Oracle 제품은 OID를 사용하여 엔터프라이즈 보안 정보를 관리하며, 특히 이러한 정보는 Oracle 엔터프라이즈 구성 요소 사이에 공유됩니다. 이러한 정보에는 사용자 ID, SSO 암호와 같은 인증 데이타, 롤 또는 그룹 멤버십과 같은 권한 부여 데이타 등이 포함됩니다. Oracle9*i*AS에서 OID는 Oracle9*i*AS 인프라의 핵심 구성 요소이며, 사용자, 인증 및 권한 부여 정보에 대한 공동 저장소입니다. 이것은 Oracle9*i*ASv1에 존재하던 구성 요소별 저장소를 대체합니다. OID는 사용자 권한을 나타내기 위한 공동의 LDAPv3 표준 프레임워크를 제공합니다. OID의 사용은 Oracle9*i*AS 구성 요소에 대한 공동 권한 관리 처리 방법을 제공할 뿐만 아니라, Oracle9*i*AS와 기타 LDAP 호환 엔터프라이즈 구성 요소 사이의 통합 권한 관리를 위한 수단을 제공합니다.
OID에서 권한을 관리하기 위해 Oracle9*i*AS에는 시스템 관리자 및 경우에 따라 사용자 자신이 OID에서 사용자 정보를 관리할 수 있는 애플리케이션인 DAS(Delegated Administrative Services)도 도입되었습니다. DAS에는 웹 기반 GUI 애플리케이션과 OID 데이타의 관리를 위한 API가 모두 포함되어 있습니다.

확장 가능한 인증

OID는 사용자 암호 관리를 위한 집중된 보안 저장소를 제공합니다. Oracle9*i*AS에서 OID에는 암호 확인 API를 통한 확장 가능한 인증에 대한 지원이 새로 포함됩니다. 이 API는 다양한 사용자 정의 및 써드 파티 인증 처리 방법에 대한 지원 기능을 제공하며, Oracle9*i*AS SSO 섹션에서 이미 설명했습니다.

써드 파티 디렉토리 지원

여러 애플리케이션에 걸쳐 단일 디렉토리 기반의 보안 프레임워크를 제공하기 위해 OID를 다른 써드 파티 디렉토리 제품과 통합해야 할 필요가 있습니다. 이에 따라, Oracle9*i*AS에는 DIP(Directory Integration Platform)가 도입되었습니다. DIP는 OID와 써드 파티 디렉토리 사이에 커넥터를 구현하기 위한 프레임워크를 제공하고, OID와 기타 디렉토리간 참조 및 동기화를 지원합니다.

ORACLE HTTP SERVER 보안

Oracle HTTP Server는 Oracle9*i*AS의 Oracle 웹 서버 구성 요소이며, 공개 소스인 Apache 웹 서버를 기반으로 합니다. 가장 널리 채택되고 있는 웹 서버 제품 중 하나인 Apache 서버는 다양한 기존 애플리케이션을 지원하며, 유연하고 쉬운 보안 모델을 제공합니다. Apache는 보안 애플리케이션 구현에 있어 입증된 플랫폼입니다. Apache에 친숙한 고객은 Oracle HTTP Server를 사용하여 보안 애플리케이션을 쉽게 구축 및 배치할 수 있습니다.

HTTP SERVER 보안 서비스 개요

Oracle HTTP Server는 다양한 표준 및 Oracle 고유의 향상 기능(Apache 커뮤니티에서 언급되는 대로 "mod"라고도 함)으로 Apache를 확장합니다. 이를 통해 웹 브라우저 사용자들은 표준 웹 프로토콜을 사용하여 Oracle9iAS에 액세스할 수 있게 됩니다. 또한, 기본적인 HTTP 리스너 기능(HTTP 및 보안 HTTP 또는 HTTPS)을 제공하고, 다양한 인터페이스를 통해 정적 웹 페이지 및 동적 컨텐트 모두에 대한 액세스 기능을 제공합니다.

Oracle HTTP Server 보안 서비스에는 기본적인 경쟁/응답 작업, 클라이언트 제공 X.509 인증서, IP 또는 호스트 이름 주소 등을 통한 사용자의 ID를 기반으로 파일 및 서비스에 대한 액세스를 제한하거나 허용하는 기능이 포함됩니다.

Oracle HTTP Server 보안의 또 다른 중요 기능은 클라이언트와 서버간에 교환되는 데이타의 보호입니다. 이 기능은 SSL 프로토콜을 통해 제공되고, 사용자와 HTTP 서버 모두의 데이타 무결성 및 강력한 인증도 제공합니다.
또한, Oracle HTTP Server는 침입 시도를 탐지 및 해결하기 위해 필요한 로깅 및 기타 기능을 제공하며, 다른 Oracle9*i*AS 구성 요소 및 Oracle8i 데이타베이스와 같은 제품과의 통합도 가능합니다. 이를 통해, Oracle HTTP Server는 웹 애플리케이션 구축을 위한 종합적인 보안 서비스를 제공합니다.

액세스 제어

URL 요청이 Oracle HTTP Server에 도달하면, 유명한 대부분의 웹 서버/리스너에서 공통적인 mod/플러그인 아키텍처를 통해 여러 단계로 처리됩니다. 액세스 제어는 요청 프로세스 단계 초기에 적용됩니다.
Oracle HTTP Server 액세스 제어는 서버 관리자가 서버의 특정 파일, 디렉토리 또는 URL에 대한 액세스를 제한할 수 있는 Apache 액세스 제어 처리 방법을 기반으로 합니다. 서버에서 제한되는 각 객체에 대해 관리자는 지시어를 사용하여 해당 객체에 대한 액세스가 요청자와 연결된 하나 이상의 속성 값을 기반으로 거부되거나 허용되도록 지정할 수 있습니다. 관리자는 deny, allow, order 등과 같은 지시어를 구성하여 호스트 이름, IP 주소, 브라우저 종류와 같은 사용자 속성을 기반으로 추가 프로세스 작업을 제한할 수 있습니다. 제한은 〈files〉, 〈directory〉, 〈location〉 구성 지시어를 사용하여 각각 특정 파일, 디렉토리, URL 형식에 적용

될 수 있습니다.

예를 들어, 다음의 경우 192.168.1.* 범위의 모든 IP 주소 또는 us.oracle.com이 포함된 호스트 이름에서 오는 모든 요청은 /internalonly/ 디렉토리의 파일에 대한 액세스가 허용됩니다.

```
<Directory /internalonly/>
  order deny,allow
  deny from all
  allow from 192.168.1.* us.oracle.com
</Directory>
```

호스트 이름은 속이기가 쉽기 때문에 인터넷 액세스에 대해 호스트 이름을 기준으로 액세스를 허용 또는 제한하는 것은 보안을 제공하는 현명한 방법이 아닙니다. 속이기가 약간 더 어렵기는 하지만 IP 주소의 경우도 마찬가지입니다. 따라서, 인트라넷 사용에 대해 IP/호스트 이름을 통한 액세스 제어는 인터넷 IP 및 호스트 이름 제한이 가능하지 않은 많은 상황에서 유용하게 사용할 수 있습니다.

Oracle HTTP Server가 공개 소스인 Apache Server를 기반으로 하지만, 보안을 향상시키는 몇 가지 액세스 제어 기능이 더 포함되어 있습니다. 예를 들어, Apache Server는 접미어 .htaccess가 붙은 파일을 통해 디렉토리/폴더별 액세스 제한 기능을 제공합니다. .htaccess 프로세싱의 경우 보안과 성능 저하 문제가 동반하여 나타나기 때문에 Oracle HTTP Server에서는 이러한 파일의 프로세싱이 기본적으로 비활성화됩니다.

사용자 인증 및 권한 부여

많은 애플리케이션의 경우 사용자 ID를 기준으로 웹 서버의 리소스에 대한 액세스를 제한하는 것이 바람직합니다. Oracle HTTP Server는 X.509 인증서, 사용자명/암호(basic 인증) 및 기타 방법을 사용하여 SSL(Single Sockets Layer)을 통한 클라이언트 인증을 포함한 사용자 인증을 위한 여러 가지 처리 방법을 제공합니다. 서버 관리자는 특정 URL에 대한 액세스가 특정 처리 방법을 통해 인증되어야 하는 특정 사용자에게 제한되도록 Apache 지시어로 지정할 수 있습니다.

사용자 인증이 설정되면 URL 요청의 추가 프로세싱을 제한 또는 허용하는 규칙을 추가적으로 적용할 수 있습니다. 사용자 ID를 기준으로 하는 액세스 제어 지시어는 IP 주소 또는 호스트 이름을 기준으로 하는 지시어(위에서 설명)와 함께 사용할 수 있으므로 추가 프로세싱을 허용하기 위해서는 사용자 요청이 두 지시어를 모두 만족해야 합니다. 예를 들어, 특정 이름의 사용자 액세스 및 기업 인트라넷 내부 액세스만 허용하는 특정 URL에 대한 액세스를 제한할 수 있습니다. 이를 위해서는 사용자 인증, 특정 이름의 사용자를 제외한 액세스 거부 및 특정 범위 내의 IP 주소(인트라넷 내부의 사용자 확인) 허용을 통해 해당 URL에 대한 지시어를 구성해야 합니다.

즉, Oracle HTTP Server에서 구현되는 Apache 지시어 액세스 제어 처리 방법은 객체에 대한 사용자 액세스 관리에 있어 고도의 유연성을 제공합니다.

MOD_OSSO

Mod_osso는 Oracle9*i*AS에서 Oracle HTTP Server의 새로운 기능으로 HTTP Server가 SSO 지원 파트너 애플리케이션이 될 수 있도록 합니다. Mod_osso는 이 문서의 Oracle9*i*AS SSO 섹션에 자세히 설명되어 있습니다.

SSL(SECURE SOCKETS LAYER)

SSL은 Oracle9*i*AS HTTP Server와 클라이언트 브라우저 사이에서 포인트-투-포인트 보안 기능을 제공합니다. SSL에 의해 제공되는 보안 관련 서비스에는 인증, 권한 부여, 비밀 유지, 데이타 무결성 등이 포함됩니다. 다음은 이러한 서비스에 대한 설명입니다.

SSL 비밀 유지

SSL에 의해 제공되는 주요 서비스는 비밀 유지로 메시지를 암호화하여 다른 사람이 읽거나 해독하지 못하도록 하는 것입니다. SSL은 표준 암호화 처리 방법을 사용하여 데이타를 암호화하고 통신 장치간에 키를 배포합니다. 선택되는 특정 암호화, 무결성 보호, 키 배포 알고리즘은 사용되는 암호화 키 길이와 함께 암호화 집합(ciphersuite)을 정의합니다. Oracle9*i*AS SSL 구현은 광범위한 표준 암호화 집합을 지원합니다. 특히, Oracle9*i*AS는 인증 및 키 배포를 위해 X.509 인증서를 사용하는 암호화 집합(PKI 인증이라고도 함)을 지원합니다.

Oracle HTTP Server는 SSL 세션이 캐시되도록 하여 두 IP 주소간의 여러 메시지 교환이 한 세션 아래서 이루어질 수 있습니다. 세션 캐싱은 성능적 측면에서 매우 중요합니다. SSL 세션 연결은 CPU 사용량이 매우 높아 사용 가능한 CPU 리소스의 90%까지 점유하는 것으로 알려져 있습니다. SSL 세션 캐싱은 SSLSessionCache 지시어를 통해 지정되고, 이 지시어의 매개변수는 SSL 세션 정보가 관리되는 파일 또는 공유 메모리 세그먼트를 지정합니다.

SSL 클라이언트 인증

SSL은 PKI(public key infrastructure) 구현의 일부로 X.509 인증서를 사용하여 클라이언트 인증을 제공하기 위해 사용될 수도 있습니다. Oracle HTTP Server는
클라이언트의 X.509 인증서에 있는 정보를 기준으로 파일 및 서비스에 대한 액세스를 제한하도록 구성될 수 있습니다. 액세스 여부를 결정하는 데 사용될 수 있는 정보에는 클라이언트 인증서의 DN(distinguished name), DN 내의 프로파일 정보, 인증서 신뢰 포인트(즉, 사용자의 인증서를 발급한 인증 기관) 등이 포함됩니다. SSL은 알고 있는 신뢰 포인트 또는 알고 있는 신뢰 포인트가 서명한 신뢰 포인트 등을 허용하도록 구성될 수 있습니다. 인증은 일부 또는 전체 DN의 목록, 또는 이러한 이름의 "일드 카드" 버전을 기준으로 이루어질 수 있습니다.

SSL 인증이 이루어지면 인증서에서 사용 가능한 정보는 위에서 설명한 대로 〈directory〉, 〈files〉, 〈location〉

과 같은 지시어로 사용될 수 있습니다. SSL 인증은 기본 인증 및/또는 호스트 기반 액세스 제어와 함께 사용될 수 있습니다. 이와 같은 방식으로 파일 및 서비스에 대해 여러 조합의 SSL 및 기본 인증 사용자 액세스를 허용하거나 제한할 수 있으며, 이러한 제한 방식을 호스트 기반의 액세스 제어와 함께 사용할 수도 있습니다.
Oracle9*i*AS의 새로운 기능에는 Oracle9*i*AS SSO에 대한 SSL 클라이언트 인증 지원이 포함됩니다. 자세한 정보는 Oracle9*i*AS SSO에서의 PKI 지원 섹션에 설명되어 있습니다.

SSL 환경 변수

Oracle9*i*AS는 환경 변수라고 하는 SSL 세션에 대한 보안 관련 정보가 CGI 스크립트, 서블릿, Perl 스크립트 등과 같은 웹 서버 애플리케이션에 전달되도록 합니다. 애플리케이션은 이러한 환경 변수를 사용하여 사용자에 대한 정보 또는 사용자 대신 이루어진 SSL 세션의 유형을 기준으로 사용자 요청에 대한 추가 액세스 제어 또는 인증을 수행할 수 있습니다.
환경 변수에는 다음과 같은 정보가 포함됩니다.

- HTTPS 요청으로 도달된 URL
- SSL 세션에서 사용된 암호화 키의 크기
- SSL 세션에서 사용된 암호화 집합
- 클라이언트 인증서의 DN

SSL 로깅

Oracle HTTP Server는 SSL 관련 정보의 로깅도 제공합니다. 이러한 정보는 침입이 시도되었는지 여부 및 이러한 침입의 성공 여부를 결정하는 데 사용될 수 있습니다. 또한, 침입 공격의 대상 결정 또는 기타 목적으로도 사용될 수 있습니다.

ORACLE DATABASE에 대한 보안 액세스

Oracle9*i*AS는 Oracle 데이타베이스 백 엔드 저장소를 사용하여 3-Tier시스템을 쉽게 구축할 수 있도록 합니다. Oracle9*i*AS는 데이타베이스 액세스 및 데이타베이스에 대한 애플리케이션 호출을 위한 다양한 처리 방법을 제공합니다. 가장 일반적으로 사용되는 처리 방법은 팻(fat) 클라이언트 JDBC 및 Oracle9*i*AS가 Oracle 데이타베이스 프로그래밍 언어인 PL/SQL로 작성된 데이타베이스 애플리케이션을 호출할 수 있도록 하는 Oracle HTTP Server의 플러그인인 mod_plsql입니다. JDBC(팻 클라이언트에서 실행될 때) 및 mod_plsql은 Oracle 클라이언트-서버 네트워킹 프로토콜을 사용하여 Oracle 데이타베이스에 액세스하기 때문에 팻 클라이언트 JDBC 또는 mod_plsql을 사용하는 개발자들은 Oracle Advanced Security(Oracle9*i* 옵션)를 활용하여 Oracle9*i*AS와 Oracle9*i* 데이타베이스간에 교환되는 데이타를 보호할 수 있습니다.
Oracle Advanced Security는 Oracle 데이타베이스 클라이언트와 서버에 암호화, 무결성 보호, 고급 인증

서비스 등을 제공하며, SSL과 같은 업계 표준 암호화 프로토콜 및 RSA의 RC4, DES, 3DES 등을 포함한 표준 암호화 알고리즘을 지원합니다.

Oracle은 많은 방화벽 솔루션 업체와 협력하여 Oracle Advanced Security에 의해 암호화된 데이타가 모든 일반적인 상용 방화벽 제품에서 지원되도록 하고 있습니다. Oracle Advanced Security는 Oracle9*i*AS와 Oracle9*i*간에 교환되는 데이타가 회사의 내부 정보에 대한 액세스 권한을 가지고 있는 내부 침입자에 대해서도 안전하게 유지되도록 합니다. Oracle9*i*는 프록시 인증이라는 기능도 제공합니다. 이 기능은 3-Tier 애플리케이션 설계에서 발생할 수 있는 성능 문제를 해결하기 위해 고안되었습니다. 이 기능을 통해 Oracle9*i*AS가 사용자 컨텍스트를 전환할 때마다 로그아웃과 로그인을 반복할 필요 없이 Oracle9*i*AS는 Oracle9*i* 데이타베이스에 액세스하여 특정 Oracle9*i* 사용자 권한을 얻을 수 있습니다. 더욱이, 이 기능은 인증된 사용자 대신 데이타베이스에 액세스할 때 중간 계층 애플리케이션 서버에 부여되는 제한적인(완전하지는 않더라도) 신뢰와 관련된 보안 문제를 해결합니다. 과거에는 애플리케이션 개발자가 중간 계층 수퍼 유저 권한(예: SYS 또는 root)을 부여하여 사용자 대신 데이타베이스에 액세스할 수 있도록 하거나 중간 계층에 데이타베이스 사용자 암호를 저장해 두어야 했습니다. 이 두 가지 방식은 모두 안전하지 못합니다.

프록시 인증을 통해 Oracle9*i*AS는 Oracle9*i* 서버에서 하나의 인증된 세션(예: 팻 클라이언트 JDBC 또는 mod_plsql을 사용하여)을 만들고, 세션 내의 각 사용자에 대해 별도의 인증 자격 증명을 제출할 필요 없이 여러 Oracle9*i* 사용자를 대신하여 작업을 수행할 수 있습니다. 애플리케이션 서버는 어떤 사용자에 대해 대신 작업이 수행되는지 지정해야 하고, 해당 사용자 대신 작업을 수행할 수 있는 권한을 Oracle9*i*가 부여해야 합니다. 또한, 액세스 여부를 결정하거나 이벤트에 대한 감사 기록을 작성할 때 Oracle9*i*는 Oracle9*i*AS의 인증된 ID 및 Oracle9*i*AS가 프록시 인증을 수행하는 대상 사용자의 ID를 모두 사용할 수 있습니다. 프록시 인증은 데이타베이스에 대한 수퍼 유저 권한을 부여하거나 Oracle9*i*AS에 여러 데이타베이스 사용자 암호를 저장할 필요 없이 Oracle9*i*가 중간 계층 Oracle9*i*AS에 제한적인 신뢰를 위임할 수 있도록 합니다.

ORACLE9*i*AS의 JAVA 보안

Java, 특히 Java2 Enterprise Edition은 현재 많은 새로운 웹 애플리케이션의 개발 환경으로 채택되고 있습니다. Java2 Enterprise Edition은 Java2 보안 모델 및 JAAS(Java Authentication and Authorization Service)라고 하는 보안 프레임워크를 정의합니다.

Oracle9iAS는 완벽한 J2EE 호환 JAAS 제공자를 통해 이 프레임워크를 구현합니다. JAAS 제공자는 애플리케이션 개발자들이 사용 가능한 사용자 인증, 권한 부여 및 위임 서비스를 만들고, 이러한 서비스를 J2EE 애플리케이션 환경에 통합시킬 수 있도록 합니다.

JAVA2 보안 모델

Java2 보안 모델은 Sun Microsystems, Inc.의 Java 사업부에 의해 정의되었습니다. 기능을 기반으로 하며, 개발자들이 보호 도메인 및 해당 도메인과 연결되는 보안 정책을 지정할 수 있도록 합니다. 보안 정책은

이러한 도메인 내에서 실행되는 Java 클래스와 연관된 사용 권한을 지정합니다. 사용 권한은 특정 객체에 부여되는 액세스 유형(예: 디렉토리 /salaries/에 대한 읽기 액세스)을 정의합니다.

JAAS 제공자

Oracle9*i*AS JAAS 제공자는 Java2 보안 모델을 구현하여 애플리케이션 개발자가 JAAS에 의해 제공되는 표준 인증 서비스에서 인증된 사용자(기본) ID를 얻어 기본이 객체 액세스를 위해 가지는 권한을 관리하도록 합니다. 기본이 호출한 방식의 권한을 관리하기 위한 권한 위임도 지원합니다.

AAS 인증

Oracle9*i*AS JAAS 제공자는 유연한 인증 프레임워크를 지원합니다. SSL 및 SSO를 기반으로 인증을 위한 특정 처리 방법을 지원할 뿐 아니라, 개발자들이 표준 JAAS 로그인 모듈 API를 통해 사용자 정의 인증 모듈을 통합할 수 있도록 합니다.

SSL 인증

SSL 인증은 JAAS에 대한 클라이언트 X.509v3 인증서를 가지고 있는 사용자가 이러한 인증서를 사용하여 J2EE 애플리케이션에 액세스할 수 있도록 합니다. SSL 인증은 Oracle HTTP Server의 mod_ossl을 사용하여 SSL 교환을 통해 확인된 클라이언트 X.509 인증서에서 인증된 사용자 ID를 얻습니다. 그러면 이 ID는 JAAS를 통해 Java 애플리케이션에 제공될 수 있습니다.

SSO 인증

SSO 인증은 Java 애플리케이션이 사용자 인증을 위해 Oracle9*i*AS SSO를 사용할 수 있도록 합니다. 이 경우, 인증된 사용자 ID는 mod_osso에서 얻어 JAAS를 통해 Java 애플리케이션에서 사용할 수 있게 됩니다.

사용자 정의 인증

Oracle9*i*AS JAAS 제공자는 개발자들이 사용자 정의 인증 방식을 JAAS에 통합할 수 있도록 표준 JAAS 로그인 모듈 API를 지원합니다.

JAAS 권한 부여

권한 부여를 위해 완전한 롤 기반 액세스 제어 모델을 제공하는 것 이외에, Oracle9*i*AS JAAS 구현은 개발자들에게 권한 부여를 관리할 때 아키텍처 측면의 유연성을 제공합니다. 여기에는 LDAP를 사용한 집중된 관리와 XML 기반 API를 사용하여 파일 시스템을 통한 관리가 포함됩니다. 이러한 처리 방법은 표준 JAAS principals.xml보다 보안성 높은 대체 방식을 제공합니다.

LDAP-기반 권한 부여

Oracle9*i*AS JAAS 사용자 정보 및 권한 부여는 Oracle9*i*AS의 확장 가능하고 안전한 LDAP 디렉토리인 OID에

저장될 수 있습니다. LDAP에서의 권한 부여 관리는 사용자 커뮤니티가 방대하고 확장성과 집중 관리가 필수적일 때 특히 유용합니다.

XML-기반 권한 부여

Oracle9*i*AS JAAS 제공자는 인코딩 처리 방법으로 XML을 사용하여 권한 부여 API의 빠른 경량 구현도 지원합니다. 이 API를 통해 Java 개발자들은 OID가 아닌 운영 체제 파일에서 안전하게 사용자 및 규칙 정보를 가져올 수 있게 됩니다. 이 방식은 방대한 사용자 커뮤니티로의 확장이 필요하지 않을 때 Oracle9iAS의 경량 구현에 유용합니다. 사용자 암호가 암호화되지 않은 형식으로 저장되는 principals.xml과 달리, Oracle9*i*AS JAAS XML 권한 부여를 사용할 경우에는 암호가 암호화됩니다. 더욱이, principals.xml과 달리 Oracle 구현은 규칙 기반 액세스 제어 및 Java2 사용 권한 모델을 완벽하게 지원합니다. Oracle9*i*AS는 principals.xml 기반 사용자 관리에서 Oracle9iAS JAAS XML 구현으로 이전할 수 있도록 해주는 이전 도구를 제공합니다.

JAAS 위임

Oracle9*i*AS JAAS 제공자는 권한 위임을 지원하여 Java 애플리케이션이 지정된 사용자의 권한으로 실행되게 합니다. RunAsClient와 RunAsID 모두 지원됩니다. RunAsClient는 Java 애플리케이션(예: 엔터프라이즈 빈(bean), 서블릿, JSP)이 현재 클라이언트 사용자와 연결된 사용 권한으로 실행되도록 구성될 수 있음을 의미합니다. RunAsID는 빈, 서블릿, JSP 등이 지정된 사용자와 연결된 사용 권한으로 실행(예: "DBAdmin"으로 실행)되도록 구성될 수 있음을 의미합니다. 이를 통해, 개발자들은 사용자들에게 특정 기능을 수행하는 데 필요한 권한만 부여함으로써 사용자들이 잘 구성된 비즈니스 규칙(예: 엔터프라이즈 빈)의 컨텍스트에서 필요한 권한만 행사할 수 있게 되므로 해당 애플리케이션에서 최소의 권한 부여 원칙을 적용할 수 있습니다.

데이타베이스에 대한 JAVA 애플리케이션 액세스 보안

HTTPS를 통해 웹 클라이언트와 Java 애플리케이션 사이에 교환되는 정보를 보호(Oracle9*i*AS HTTP Server 보안 섹션에서 설명)할 뿐 아니라, Oracle9*i*AS는 Oracle Advanced Security 프로토콜 및 Oracle9*i* 프록시 인증을 사용하여 Java 애플리케이션과 백 엔드 데이타베이스 사이에 교환되는 정보를 보호할 수도 있습니다. 이러한 기능은 이미 Oracle HTTP Server 보안 섹션에서 설명했습니다.

ORACLE9*i*AS PORTAL 보안

Oracle9iAS Portal은 "엔터프라이즈 포탈" 분야에서 Oracle 제품의 주요 구성 요소입니다. 새롭게 형성되는 이 분야의 웹 제품은 기업 인트라넷에서 비즈니스 관련 정보에 대한 게이트웨이를 제공합니다. 원래 엔터프라이즈 포탈 시장을 대상으로 하고 있지만 Oracle9*i*AS Portal은 더욱 큰 규모의 인터넷 커뮤니티에 대한 액세스를 제공하도록 확장이 가능합니다.

Oracle9*i*AS Portal을 통해 Oracle9*i*AS 고객이 자사의 웹 컨텐트 및 애플리케이션을 구성하고 이러한 것들을

논리적이고 일관성 있는 웹 포탈 형식으로 사용자에게 제공할 수 있습니다. 또한, 사용자 및 Oracle9*i*AS Portal 컨텐트에 대한 액세스 권한을 구성하고 관리할 수 있는 도구도 제공합니다.
기존 마켓 공간의 통합 및 확장 수단으로서 엔터프라이즈 포탈은 세 가지 강력한 구성 요소를 활용할 수 있습니다.

- Oracle 데이타베이스 고유의 강력한 정보 관리 기술
- ERP(enterprise resource planning), CRM(customer relationship management), BI(businessintelligence) 솔루션 등을 포함하는 중요 비즈니스 데이타를 관리하기 위한 다양한 애플리케이션
- 이 기술을 활용하여 인트라넷의 기타 데이타스토어와 애플리케이션을 통합하는 프레임워크 (Oracle9*i*AS Portal)

Oracle9*i*AS Portal에 내장된 기능은 여러 Oracle 제품 및 애플리케이션에 걸쳐 공동 프레임워크를 제공합니다. 포탈 기능이 있는 Oracle 제품을 구입한 고객은 비즈니스 요구 및 우선 순위에 따라 점진적으로 다른 용도로 포탈 통합 기능을 쉽게 확장할 수 있습니다.
이 섹션에서는 Oracle9iAS Portal의 보안 기능 및 아키텍처에 대한 개요를 제공합니다. 사용자 및 그룹에 대한 개념 및 표현, 그리고 데이타베이스 스키마와 사용자의 관계를 설명합니다. 또한, 인증 문제, 세션 관리 및 권한 부여, 그리고 이러한 기능을 구현하는 아키텍처의 다양한 구성 요소를 설명합니다.

ORACLE9*i*AS PORTAL 보안 개요

Oracle9*i*AS Portal은 여러 애플리케이션을 하나의 종합적인 포탈 환경에 통합하기 위한 안전한 플랫폼을 제공하고, 이 환경을 효율적으로 관리하기 위한 관리 도구와 인터페이스를 제공합니다.
Oracle9*i*AS Portal은 포탈에서 컨텐트 및 애플리케이션에 대한 사용자 액세스를 위해 사용자 권한 및 그룹을 기준으로 종합적이고 확장 가능한 권한 부여 모델을 제공합니다. 또한, 애플리케이션을 포탈 인증 프레임워크에 연결하여 포탈, 파트너 또는 외부 애플리케이션으로 배치될 수 있도록 유연한 통합 모델을 제공합니다.
Oracle9*i*AS Portal은 이벤트 로깅 서비스를 통해 보안 관련 이벤트의 감사도 지원합니다.

포탈 사용자

포탈에 수백만 명의 사용자가 액세스할 수 있는 인터넷 컴퓨팅 모델에서는 사용자를 최대한 가볍게 표현하는 것이 중요합니다. 다수의 사용자와 방대한 데이타를 안전하고, 확장성 높으며, 장애에 대처할 수 있도록 관리하기 위해
Oracle9*i*AS Portal은 Oracle 데이타베이스의 보안 및 데이타 관리 기술을 활용합니다.
Oracle9*i*AS Portal은 자신의 사용자 계정을 정의하는데, 이러한 각 계정은 Oracle 데이타베이스에서 고유하게 연결된 데이타베이스 스키마를 가지지 않기 때문에 "경량"이라고 합니다. 반면, 각 Oracle9*i*AS Portal 사용자

계정은 Oracle9*i*AS SSO 사용자 계정과 고유하게 일치합니다. Oracle9*i*AS Portal은 Oracle9*i*AS SSO 지원 애플리케이션의 사용자들이 관리되는 처리 방법을 제공합니다.

설치된 사용자

Oracle9*i*AS Portal이 설치되면 기본 사용자 계정이 만들어집니다. 여기에는 포탈 관리자 계정과 public 계정이 포함됩니다. public 계정은 특정 포탈 컨텐트에 대해 공개적으로 액세스가 가능합니다. 즉, 자신의 Oracle9*i*AS Portal 사용자 계정을 가지고 있는 않은 사용자 또는 계정을 가지고 있지만 아직 로그인하지 않은 사용자도 컨텐트에 액세스할 수 있습니다. 기본적으로 두 개의 포탈 관리 계정이 만들어집니다. 이 중에서 portal은 연결된 Oracle 데이타베이스의 DBA(database administrator)인 Oracle9*i*AS Portal 관리자를 위해 만들어집니다. 또 다른 하나인 portal_admin은 DBA 권한이 필요하지 않은 Oracle9*i*AS Portal 관리자를 위한 계정입니다.

사용자 생성

Oracle9*i*AS에서 Portal은 사용자 관리를 위해 OID를 사용하게 되었습니다. 사용자 관리는 새로운 OID DAS 프레임워크를 사용하여 수행됩니다.

포탈 그룹

Oracle9*i*AS Portal은 그룹을 지원하며, 그룹은 크게 두 가지 목적으로 사용될 수 있습니다. 그룹은 한 번에 많은 사용자들에게 권한을 부여할 수 있는 편리한 방법을 제공합니다. 또한, 포탈 시스템의 일부 속성은 그룹과 연결될 수 있는데, 어떤 사용자가 환경 설정에서 기본 그룹이 지정되어 있을 경우 이러한 속성이 해당 사용자의 세션에 적용될 수 있습니다. 이러한 속성의 예에는 그룹의 기본 홈 페이지 또는 기본 스타일이 있습니다.
그룹은 사용자는 물론 다른 그룹도 포함할 수 있습니다. 따라서, 계층적인 그룹 구성이 가능합니다. 다른 그룹의 하위 그룹에 속하는 사용자는 그룹 멤버십에 따라 상위 그룹의 멤버입니다. 따라서, 상위 그룹에 부여된 권한은 하위 그룹의 사용자도 가지게 됩니다.

그룹 생성

그룹을 만들기 위해서는 사용자가 그룹 생성 권한을 가지고 있어야 합니다. 기본적으로 이 권한은 인증된 모든 사용자에게 부여되며, 특정 사용자로 한정시킬 수도 있습니다. 사용자가 그룹을 만들 경우에는 그룹명, 그룹에 대한 간략한 설명, 그룹의 범위가 특정 컨텐트 영역으로 제한되는지 여부, 그룹의 존재를 다른 사용자들로부터 숨길 것인지 여부 등을 정의합니다. 그런 다음 그룹의 멤버십을 지정하고, Oracle9*i*AS Portal 관리자가 특정 권한을 부여할 수 있는 권한을 부여 받은 경우 해당 권한을 부여할 수 있습니다.
그룹은 비밀로 지정될 수도 있습니다. 이 경우, 그룹의 소유자로 지정된 사용자들만 모든 그룹 목록에서 해당 그룹을 볼 수 있게 됩니다. 그룹 멤버를 포함한 다른 모든 사용자는 해당 그룹을 볼 수 없습니다. 하지만, 이 비밀 속성은 어떤 방식으로도 그룹 멤버십을 통한 사용자 권한 부여와 같은 그룹의 작동 방식에 영향을 미치지 못합니다.

포탈 인증

Oracle9*i*AS Portal은 사용자 인증을 위해 Oracle9*i*AS SSO 파트너 애플리케이션으로 구현됩니다. 인증되지 않은 사용자는 public 사용자 계정을 통해 Oracle9*i*AS Portal의 특정 컨텐트에 액세스하지 못할 수도 있습니다.

포탈 권한 부여

권한 부여는 사용자의 ID를 기준으로 Oracle9*i*AS Portal의 여러 영역에 대한 액세스를 제어하는 프로세스입니다. 사용자가 식별되면 인증 과정을 거친 후 Oracle9*i*AS Portal이 ID를 기준으로 해당 사용자에 대한 적절한 권한 부여를 결정합니다. Oracle9*i*AS Portal은 포탈의 각 객체에 대해 액세스 제어 목록을 정의하는 데 사용되는 확장 가능한 권한 집합을 제공합니다. 다음 섹션에서는 이 모델에 대해 설명합니다.

포탈 권한

Oracle9*i*AS Portal에는 포탈의 특정 객체(웹 페이지 또는 폴더)에 대한 특정 액세스 유형에 대해 사용자에게 할당할 수 있는 다양한 권한이 포함되어 있습니다. 권한은 주어진 객체 유형 안에서 순위가 결정되어 있으며, 상위의 권한은 하위의 모든 권한을 포함합니다. 따라서, 사용자에게 특정 페이지와 관련된 권한이 직접 부여된 경우 해당 사용자는 해당 페이지에 대한 모든 하위 권한도 가지게 됩니다. 이 계층적 또는 누적 방식을 통해 애플리케이션은 주어진 사용자가 특정한 작업을 수행할 수 있는 최소한의 사용 권한을 가지고 있는지 여부를 결정할 수 있게 됩니다.
권한에는 글로벌 권한 및 인스턴스별 권한의 두 가지 유형이 있으며, 포탈 환경에서 객체에 대한 사용자 액세스를 관리하기 위한 편리한 처리 방법을 제공합니다.

글로벌 권한

글로벌 권한은 지정된 유형의 모든 객체에 걸쳐 적용되는 권한입니다. 예를 들어, ANY_PAGE/EDIT 권한이 부여될 경우 특정 페이지에 대해 명시적인 인스턴스별 권한이 주어졌는지 여부에 상관 없이 시스템의 모든 페이지를 편집할 수 있습니다.

인스턴스별 권한

인스턴스별 권한은 사용자 또는 그룹에 부여되어 특정 페이지 또는 특정 폴더나 항목과 같은 객체의 특정 인스턴스에 대한 권한을 정의합니다.

애플리케이션 통합

Oracle9*i*AS Portal의 주요 기능 중 하나는 다양한 애플리케이션을 하나의 프레임워크에 통합하여 사용자들에게 해당 비즈니스 애플리케이션에 대한 효율적인 액세스 경험을 제공하는 것입니다. 이를 위해 Oracle9*i*AS Portal은 Oracle9*i*AS SSO에 의존하며, 포탈 보안과 애플리케이션 보안 사이에 다양한 수준의 통합을 지원합니다.

가장 높은 수준의 통합은 애플리케이션이 Oracle9*i*AS Portal 자체에서 구현될 때이며, Oracle9*i*AS Portal에서 직접 사용자 ID를 얻습니다. 이러한 애플리케이션을 포탈 애플리케이션이라고 합니다. Oracle9*i*AS Portal 자체는 Oracle9*i*AS SSO에 대한 파트너 애플리케이션으로서 SSO 서버에서 사용자의 ID를 얻을 수 있습니다. 사용자가 Oracle9*i*AS SSO를 통해 인증되면 포탈 애플리케이션은 Oracle9*i*AS Portal에서 직접 사용자의 ID를 얻습니다.

Oracle9*i*AS Portal을 통해 액세스 가능한 기타 애플리케이션은 Oracle9*i*AS SSO 섹션에서 설명한 대로 Oracle9*i*AS SSO 파트너 애플리케이션이나 외부 애플리케이션이 될 수 있습니다. 파트너 애플리케이션은 포탈 자체에서 구현될 필요는 없지만, Oracle9*i*AS Portal도 사용하는 Oracle9*i*AS SSO 인증 프레임워크에 포함되어야 합니다. 외부 애플리케이션은 자신의 인증 처리 방법을 관리하며, Oracle9*i*AS SSO에 의해 제공되는 인증 프레임워크에 직접 포함되지 않습니다.

HTTPS 지원

더욱 높은 수준의 보안을 위해 일부 설치의 경우 SSL을 사용할 필요가 있습니다. Oracle9*i*AS SSO Server와 Oracle9*i*AS Portal은 모두 HTTPS 모드로 실행될 수 있습니다. 또는, 성능적 측면을 고려하여 Oracle9*i*AS SSO Server는 HTTPS 모드에서 실행되고, Oracle9*i*AS Portal은 HTTP 모드에서 실행되도록 구성하는 것이 더 바람직할 수도 있습니다.

감사

시스템에 대한 무단 사용을 나타낼 수 있는 모니터링 활동 이외에, 보안 관련 이벤트에 대한 감사는 권한이 부여된 사용자가 시스템 사용 정책을 준수하고 "정직한 사용자들을 정직하게 유지"하는 가장 효과적인 방법입니다." Oracle9*i*AS Portal은 특정 보안 이벤트를 로깅하고 포탈 애플리케이션이 정의한 임의의 이벤트를 로깅하도록 호출되는 로깅 서비스를 제공합니다. Oracle9*i*AS Portal 애플리케이션의 보고 기능을 통해 로그된 데이타를 볼 수 있습니다. 이벤트 로깅은 보안 관련 이벤트를 감사하고, 시스템 보안을 뚫거나 보안 정책에 위배되는 방법으로 시스템을 사용하려는 시도를 탐지하기 위한 목적으로 사용됩니다.

결론

보안은 웹 애플리케이션을 배치할 때 매우 중요한 고려 사항입니다. Oracle9*i*AS는 Apache 기반의 Oracle HTTP Server, Oracle의 J2EE 프레임워크 및 Oracle9*i*AS Portal을 사용하여 웹 애플리케이션을 구축하기 위한 완벽한 프레임워크를 제공합니다. Oracle9*i*AS 보안은 Apache가 제공하는 기본적인 테스트된 구성이용이한 웹 보안 서비스부터 종합적인 웹 싱글 사인온, 디렉토리 기반 권한 부여 및 사용자 관리, Java2 보안 서비스 등을 추가하여 포탈 보안 및 애플리케이션 통합 처리 방법으로 이러한 기능을 더욱 확장합니다. Oracle9*i*AS는 Oracle Advanced Security를 사용하여 Oracle 데이타베이스 시스템에 대한 안전한 액세스도 지원합니다. 이러한 기능은 안전한 웹 애플리케이션을 구축 및 배치할 때 애플리케이션 서버로 Oracle9*i*AS를 선택하게 되는 주된 요인이 됩니다.

ORACLE®

한국오라클(주)
서울특별시 강남구 삼성동 144-17
삼화빌딩
대표전화 : 2194-8000
FAX : 2194-8001

한국오라클교육센타
서울특별시 영등포구 여의도동 23-10
SK증권빌딩 11층(사무실)
19 · 20층(강의실)
대표전화 : 3779-4000
FAX : 3779-4100 1

대전사무소
대전광역시 서구 둔산동 929번지
대전둔산사학연금회관 18층
대표전화 : (042)483-4131 2
FAX : (042)483-4133

대구사무소
대구광역시 동구 신천동 111번지
영남타워빌딩 9층
대표전화 : (053)741-4513 4
FAX : (053)741-4515

부산사무소
부산광역시 동구 초량동 1211 7
정암빌딩 8층
대표전화 : (051)465-9996
FAX : (051)465-9958

울산사무소
울산광역시 남구 달동 1319-15번지
정우빌딩 3층
대표전화 : (052)267-4262
FAX : (052)267-4267

광주사무소
광주광역시 서구 양동 60-37
금호생명빌딩 8층
대표전화 : (062)350-0131
FAX : (062)350-0130

고객에게 완전하고 효과적인
정보관리 솔루션을 제공하기 위하여
오라클사는 전 세계 145개국에서
제품, 기술지원, 교육 및
컨설팅 서비스를
제공하고 있습니다.

http://www.oracle.com
http://www.oracle.com/kr

제품구입문의

수신자부담 전화번호 : 00368-440-0051 수신자부담 팩스번호 : 00368-440-0062 E-Mail문의 : oracleisd_kr@oracle.com