ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: INFORMATYKA z. 36 Nr kol. 1414

Bożena MAŁYSIAK
Politechnika Śląska, Instytut Informatyki

MECHANIZMY OCHRONY DANYCH W WIELODOSTĘPNYM SYSTEMIE ZARZĄDZANIA BAZAMI DANYCH ORACLE

Streszczenie. W artykule przedstawiono mechanizmy ochrony danych w wielodostępnym systemie zarządzania bazą danych Oracle. Opisano metody ochrony przed nieautoryzowanym dostępem do bazy danych, przed nieautoryzowanym dostępem do obiektów schematu, a także mechanizmy kontroli używania zasobów oraz nadzorowania akcji użytkowników.

DATA SECURITY MECHANISMS IN MULTIACCESS DATABASE MANAGEMENT SYSTEM ORACLE

Summary. This article provides an overview of the data security mechanisms in multiaccess database management system such as Oracle. It explains strategy for protecting information, which is in database included. It also describes methods of database user authentication, creating profiles and limiting use of system resources.

1. Wprowadzenie

W ostatnich latach w otaczającej nas rzeczywistości można zaobserwować znaczny wzrost zainteresowania problematyką bezpieczeństwa systemów informatycznych. Wzrost ten jest związany z rozwojem obszaru zastosowań informatyki. Pojawiają się problemy związane na przykład z ochroną danych personalnych, bezpiecznym wykorzystywaniem informatycznych systemów bankowych czy też ochroną ważnych informacji gospodarczych.

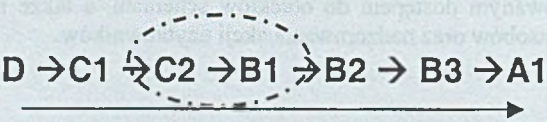
Można sformułować pewne żądania bezpiecznego systemu informatycznego twierdząc, że jeżeli istnieje uzasadniona potrzeba ograniczania dostępu do pewnych lub wszystkich zgromadzonych zasobów informacyjnych, to system powinien zapewnić odpowiednią do

zastosowania kontrolę dostępu do danych przy zachowaniu zarówno warunków integralności danych, jak i dostatecznego poziomu wydajności użytkowej systemu [4].

Próby rozwiązywania problemów związanych z ochroną systemów informatycznych rozpoczęły się około 30 lat temu w Stanach Zjednoczonych, a kilkanaście lat (od 1967 r. do 1982 r.) trwały prace nad sformalizowaniem metod rozwiązywania tych problemów i opublikowaniem ich w postaci dokumentu o nazwie „Orange Book" lub TCSEC[1]. Równie ważnym dokumentem dla problematyki bezpieczeństwa systemów baz danych jest TDI[2] [4].

„Orange Book" jest zbiorem pewnych warunków formalnych, stanowiących podstawę oceny stopnia bezpieczeństwa systemu informatycznego, w dziedzinach polityki bezpieczeństwa (security policy), oznaczania informacji (marking), identyfikacji użytkowników informacji (identification), nadzoru (accountability), mechanizmów zabezpieczających (assurance) i ciągłości ochrony (continous protection).

System informatyczny, w zależności od tego jakie warunki bezpieczeństwa zapewnia, może zostać zaklasyfikowany do jednej z siedmiu standardowych klas bezpieczeństwa (rys.1).



Rys. 1. Podwyższenie klasy ochrony
Fig. 1. Direction of increase class of security

Klasy C2 i B1 mają znaczenie dla produktów komercyjnych, jakimi są na przykład produkty Oracle'a, dlatego zostały wyróżnione.

W Europie także podjęto próby określenia pewnych standardów w dziedzinie oceny bezpieczeństwa systemów informatycznych - powstał dokument o nazwie ITSEC[3], zawierający zbiór kryteriów oceny bezpieczeństwa systemów stosowanych w krajach europejskich. W podejściu europejskim wyróżniono dwa aspekty związane z bezpieczeństwem systemu: funkcjonalność bezpieczeństwa oraz poziom ochrony.

W roku 1991 Oracle poddał ocenie formalnej ze względu na bezpieczeństwo dwa produkty: Oracle7 Server oraz Trusted Oracle7 Server. Oracle7 Server został zaklasyfikowany do klasy C2 względem US TCSEC/TDI (F-C2/E3 względem European ITSEC), a Trusted Oracle7 Server do klasy B1 względem US TCSEC/TDI (F-B1/E3 względem European ITSEC).

---

[1] DoD 52000, 28-STD, Department of Defense Trusted Computer System Evaluation Criteria.

[2] Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, (NSCS-TG-021).

[3] Information Technology Security Evaluation Criteria, Version 1.2, Commision of the EC, 199133.

W kolejnych punktach artykułu przedstawione zostaną mechanizmy bezpieczeństwa systemu informatycznego wykorzystane w produktach Oracle'a, zapewniające im przynależność do odpowiednich klas bezpieczeństwa.

2. Mechanizmy ochrony zapewniające przynależność do klasy C2

Z punktu widzenia ochrony bazy danych można wyróżnić dwie kategorie ochrony: ochronę systemu oraz ochronę danych.

Ochrona systemu powinna zawierać: sprawdzenie nazwy/hasła użytkownika, określenie, czy użytkownik ma prawo połączenia się z bazą (autoryzacja), przydzielanie miejsca na dysku dla obiektów użytkownika, określenie ograniczeń zasobów dla użytkownika, możliwość monitorowania bazy danych oraz określenie, jakie operacje systemowe użytkownik może wykonywać.

Natomiast ochrona danych powinna zawierać: określenie, jacy użytkownicy mają prawo dostępu do określonych schematów obiektów, określenie, jakie typy operacji są dozwolone dla określonych użytkowników, określenie operacji, które będą monitorowane dla każdego schematu obiektów.

2.1. Tworzenie kont użytkowników

Po zainstalowaniu systemu zarządzania bazą danych Oracle wszystkimi uprawnieniami dysponuje wyłącznie administrator bazy danych. Po instalacji w bazie danych znajdują się następujące konta użytkowników:

- **SYS** - właściciel słownika danych - hasło po instalacji *„CHANGE_ON_INSTALL"*, ma uprawnienia do stworzenia bazy danych oraz obiektów podczas rozpoczynania pracy z bazą (tabel, przestrzeni),
- **SYSTEM** - pierwszy administrator bazy, który ma wszystkie prawa do zarządzania bazą danych - hasło po instalacji *„MANAGER"*. Użytkownik ten ma uprawnienia do tworzenia użytkowników bazy danych.

Administrator systemu bazy danych korzysta z mechanizmów pozwalających na utrzymanie bezpieczeństwa bazy danych. Mechanizmy te umożliwiają:

- tworzenie kont użytkowników i przypisywanie im haseł (kontrola dostępu do bazy: autoryzowanie użytkowników do podłączenia się do bazy, przypisywanie im haseł, ograniczenie dostępnej przestrzeni dyskowej dla każdego użytkownika),
- tworzenie schematów – kontrola, do jakich obiektów użytkownik ma dostęp,

- nadawanie użytkownikom uprawnień (przywilejów) i ról – w celu kontroli poleceń oraz akcji na obiektach, jakie użytkownik może wykonywać,
- ograniczenie dostępu do pamięci (storage quotas) – w celu kontroli zajętości przestrzeni dyskowej,
- przydzielanie użytkownikom profili – w celu kontroli dostępu do zasobów przez poszczególnych użytkowników,
- obserwację (śledzenie) - selektywną obserwację aktywności bazy danych w celu wykrycia podejrzanych akcji lub monitorowania użycia bazy.

Administrator, tworząc użytkowników, określa sposób, w jaki będą oni identyfikowani przez system. Możliwa jest identyfikacja bezpośrednia przez system bazy danych Oracle7 lub za pośrednictwem systemu operacyjnego komputera.

Z każdym użytkownikiem związany jest schemat o tej samej nazwie (generowany przy tworzeniu konta użytkownika). Wszystkie obiekty użytkownika są strukturami logicznymi. Po podłączeniu się do bazy użytkownik ma dostęp do wszystkich obiektów zawartych w jego schemacie. Dostęp do bazy i jej obiektów jest kontrolowany poprzez przywileje nadane każdemu schematowi.

W systemie Oracle do tworzenia konta użytkownika służy polecenie **CREATE USER:**

```
CREATE USER user
IDENTIFIED BY password
EXTERNALLY
DEFAULT TABLESPACE tablespace
TEMPORARY TABLESPACE tablespace
PROFILE profile
QUOTA  integer ON tablespace (np.: dla kilku przestrzeni)
       UNLIMITED,
```

gdzie:

- EXTERNALLY – oznacza, że użytkownik będzie weryfikowany z poziomu systemu operacyjnego komputera,
- DEFAULT TABLESPACE, – domyślna przestrzeń tabel, określa lokalizację tworzonych przez użytkownika obiektów, domyślnie przydzielana użytkownikom jest przestrzeń SYSTEM,
- TEMPORARY TABLESPACE - tymczasowa przestrzeń tabel, określa miejsce na dysku wykorzystywane przez bazę przy używaniu niektórych operacji języka SQL (np.: sortowanie, grupowanie, indeksowanie),
- PROFILE - ograniczenie dostępu do zasobów systemowych - parametr ten określa:
  - ilość czasu CPU przeznaczonego na obsługę zadań jednego użytkownika,
  - logiczną liczbę odczytów z dysku na potrzeby jednego użytkownika,

- liczbę konkurencyjnych sesji otwartych przez jednego użytkownika na kilku komputerach,
- ilość czasu bez wykonywania operacji na bazie (idle_time),

– QUOTA - ograniczenie zasobów w przestrzeni tabel - określenie limitu miejsca w przestrzeni tabel, umożliwia administratorowi ograniczenie możliwości przepełnienia przestrzeni tabel:

- 0 - oznacza, że obiekty pozostają, ale nie mogą alokować nowej przestrzeni, przestrzeń tabel jest niedostępna dla użytkownika,
- UNLIMITED - oznacza, że użytkownik nie ma żadnych ograniczeń na ilość miejsca w przestrzeni tabel.

Przykład

```
CREATE USER student
IDENTIFIED BY stu23dent67
DEFAULT TABLESPACE stud_grup
TEMPORARY TABLESPACE stud_temp
PROFILE studenci
QUOTA 15M ON stud_grup
```

Jeśli użytkownik nie ma nadanego prawa ALTER USER, może sam zmienić sobie tylko hasło. Pozostałe parametry może zmieniać tylko użytkownik posiadający uprawnienie ALTER USER.

2.2. Identyfikacja użytkownika za pośrednictwem systemu operacyjnego

Aby mogła nastąpić identyfikacja użytkownika za pośrednictwem systemu operacyjnego, należy:

- ustawić parametr OS_AUTHENT_PREFIX,
- stworzyć konto użytkownika w bazie danych Oracle'a, który będzie weryfikowany przez system operacyjny (jego nazwa powinna być taka jak nazwa konta użytkownika w systemie operacyjnym, tyle że poprzedzona zdefiniowanym powyżej prefiksem), poleceniem:

```
CREATE USER user
IDENTIFIED EXTERNALLY;
```

- podłączyć się do komputera jako zdefiniowany wcześniej użytkownik,
- uruchomić program SQLPLUS komendą *sqlplus /*.

W systemie operacyjnym musi być założone konto użytkownika i musi ono należeć do grupy DBA (np. w systemie Unix definicja kont użytkowników znajduje się w katalogu etc/group):

```
dba::20000: kadrowy
oper::20010:kadrowy
```

Po wykonaniu wymienionych wcześniej operacji i wpisaniu komendy *sqlplus /* nazwa użytkownika systemu operacyjnego, z dodanym do niej prefiksem (np. OPS$), porównywana jest z nazwami użytkowników zdefiniowanymi w bazie danych, jeśli znajdzie się w bazie konto użytkownika o takiej samej nazwie jak nazwa konta w systemie operacyjnym, następuje połączenie z bazą.

Przykład

Niech OS_AUTHENT_PREFIX = OPS$, nazwa użytkownika w SO „kadrowy", w bazie Oracle'a konto o nazwie „OPS$kadrowy", po wykonaniu polecenia *„sqlplus /"* następuje dołożenie prefiksu do nazwy użytkownika zdefiniowanego w systemie operacyjnym, porównanie go z nazwą użytkownika w bazie i połączenie.

2.3. Nadawanie uprawnień użytkownikom

Zadaniem każdego administratora jest kontrola dostępu do bazy danych i jej obiektów poprzez nadawanie użytkownikom odpowiednich uprawnień do wykonywania określonych operacji, ograniczanie dostępu i możliwości zmiany danych, ograniczanie możliwości wykonywania funkcji systemowych i zmian struktur bazy danych, nadawanie uprawnień: pojedynczym użytkownikom i rolom oraz wszystkim użytkownikom (PUBLIC).

Uprawnienia użytkownika można nadawać na dwa sposoby: bezpośrednio oraz poprzez role (grupy uprawnień).

2.3.1. Przywileje systemowe i obiektowe

Przywileje bazodanowe można zdefiniować jako prawa do wykonywania określonych operacji na bazie przez uprawnionych przywilejem użytkowników.

Przywileje dzielimy na:

- systemowe (system privileges) - określają prawa użytkownika do wykonywania dopuszczalnych operacji,
- obiektowe (object privileges) - wyznaczają obiekty bazy danych, na których uprawnione operacje mogą być wykonywane.

Produkty Oracle'a udostępniają administratorom systemów baz danych około 70 uprawnień systemowych oraz od 1 do 7, w zależności od klasy obiektu (tablica, procedura, perspektywa) uprawnień obiektowych.

Przykłady przywilejów systemowych (perspektywa System_privilege_map)

- *UPDATE ANY TABLE* - umożliwia użytkownikowi zmianę zawartości wierszy każdej tablicy w dowolnym schemacie bazy,
- *CREATE SESSION* - umożliwia użytkownikowi bazy na otwarcie sesji, czyli połączenie z bazą,

– *CREATE TABLE* - umożliwia użytkownikowi bazy utworzenie tablicy w obrębie własnego schematu.

Przykład przywilejów obiektowych

```
SELECT TABLE pracownik
```

Nadawanie przywilejów systemowych lub ról użytkownikom lub rolom wykonuje się za pomocą polecenia GRANT, a odbieranie za pomocą polecenia REVOKE. Można nadać przywilej lub rolę jednocześnie wszystkim użytkownikom, wtedy należy zamiast nazwy użytkownika wpisać słowo kluczowe *PUBLIC*. Opcja WITH ADMIN OPTION umożliwia przekazywanie nadanego przywileju innym użytkownikom:

```
GRANT system_priv or role TO user or role
                                PUBLIC
WITH ADMIN OPTION;

REVOKE system_priv or role FROM user or role;
                                  PUBLIC
```

Przykład

Nadanie uprawnień systemowych:

GRANT *CREATE SESSION, CREATE TABLE, CREATE VIEW* TO *STUDENT* oznacza, że użytkownik identyfikowany jako STUDENT posiada uprawnienia do nawiązania połączenia z bazą danych, tworzenia własnych tabel i perspektyw, ale nie ma innych uprawnień (do tworzenia innych obiektów), jak np. użytkownicy, role, procedury czy sekwencje.

W analogiczny sposób przekazywane są przywileje obiektowe, przy czym może tego dokonywać każdy użytkownik w stosunku do swoich własnych obiektów lub administrator, w odniesieniu do wszystkich obiektów bazy danych.

Nadawanie przywilejów obiektowych użytkownikom lub rolom wykonuje się za pomocą polecenia GRANT, a odbieranie za pomocą polecenia REVOKE. Można nadać przywilej jednocześnie wszystkim użytkownikom, wtedy należy zamiast nazwy użytkownika wpisać słowo kluczowe *PUBLIC*. Opcja WITH GRANT OPTION umożliwia przekazywanie nadanego przywileju innym użytkownikom:

```
GRANT object_priv (column1, column2)
ON    object or schema or role
TO user or role or PUBLIC
WITH GRANT OPTION;

REVOKE object_priv
ON object or schema
FROM user or role or PUBLIC
CASCADE CONSTRAINTS;
```

Przykład

Nadanie uprawnień obiektowych:

GRANT SELECT, UPDATE ON ASYSTENT.DOCHODY TO STUDENT przez użytkownika ASYSTENT lub administratora oznacza, że użytkownik identyfikowany jako

STUDENT posiada prawa dostępu do tablicy DOCHODY w zakresie wyszukiwania i aktualizacji danych (ale bez możliwości wstawiania nowych lub usuwania).

2.3.2. Role – tworzenie i nadawanie

Administrator, zamiast nadawania każdemu użytkownikowi z osobna różnych uprawnień do wykonywania operacji w bazie danych, może określić grupy użytkowników wykonujących w bazie podobne operacje i wymagających podobnych uprawnień, związanych ze stanowiskiem pracy lub funkcjami. Na podstawie wyodrębnionych użytkowników może stworzyć grupy niezbędnych uprawnień zwane rolami. Role mogą zawierać zarówno przywileje obiektowe, jak i systemowe. Nie należą do żadnego użytkownika i schematu, mogą być nadane dowolnemu użytkownikowi lub roli z wyjątkiem siebie, mogą być włączane lub wyłączane dla każdego użytkownika, mogą wymagać autoryzacji (podania hasła) do włączania, mogą również być nadawane z poziomu systemu operacyjnego.

Zalety używania ról:

- ograniczenie liczby wymienianych uprawnień przez:
  - nadawanie i odbieranie wielu przywilejów jednym poleceniem,
  - nadawanie roli nowemu użytkownikowi bez pamiętania wszystkich niezbędnych przywilejów,
  - uproszczenie zarządzania przywilejami w systemach z wieloma użytkownikami, wieloma tabelami,
- dynamiczna obsługa uprawnień:
  - zmienianie przywilejów roli, gdy zmieniają się obowiązki,
  - zmienianie przywilejów wielu użytkownikom zmieniając jedną rolę,
- wybiórcze udostępnianie uprawnień:
  - włączanie i wyłączanie ról, by włączyć i wyłączyć przywileje tymczasowo,
  - aplikacja może przeglądać słownik danych i włączać w razie potrzeby lub wyłączać role,
- dodatkowe korzyści:
  - zarządzanie przywilejami bez kaskadowych odbierań przywilejów,
  - możliwość użycia mechanizmów zabezpieczeń systemu operacyjnego przy zabezpieczaniu bazy danych,
- wydajność:
  - mniej indywidualnych przywilejów do sprawdzania i kodowania w aplikacji.

Tworzenie roli wykonuje się za pomocą polecenia CREATE ROLE, a modyfikację poleceniem ALTER ROLE. Stworzona rola może być identyfikowana przez hasło (*IDENTIFIED BY password*), może być również nadawana z poziomu systemu operacyjnego (*EXTERNALLY*):

```
CREATE ROLE role
NOT IDENTIFIED or IDENTIFIED BY password
EXTERNALLY;

ALTER ROLE role
NOT IDENTIFIED
IDENTIFIED BY password
EXTERNALLY;
```

Po stworzeniu roli należy nadać jej odpowiednie uprawnienia (poleceniem GRANT).

Do nadawania ról użytkownikom lub innym rolom służy polecenie GRANT, do odbierania polecenie REVOKE, a do usuwania polecenie DROP:

```
GRANT role TO user;
REVOKE role FROM user;
DROP ROLE role;
```

2.3.3. Skutki nadawania i odbierania przywilejów i ról

Zarówno przy nadawaniu, jak i odbieraniu użytkownikom przywilejów i ról administrator powinien zachować dużą ostrożność.

Nadanie na przykład uprawnienia użytkownikowi do przeglądania jakiejś tabeli (SELECT), nawet bez użycia członu WITH GRANT OPTION (z możliwością nadawania tego uprawnienia innym użytkownikom), nie powstrzyma go przed skopiowaniem danych z tej tabeli do tabeli, której on jest właścicielem, do której przywilejami może dowolnie dysponować (np.: może nadawać je innym użytkownikom).

Odbieranie przywilejów obiektowych może mieć efekt kaskady, gdyż przywileje obiektowe nadane z opcją WITH GRANT OPTION (tzn. z możliwością przekazywania ich pozostałym użytkownikom) są odbierane także innym użytkownikom, gdy są one odbierane użytkownikowi, który je przekazał.

Przykład

Użytkownik ADM nadał przywilej WITH GRANT OPTION użytkownikowi PRAC, a ten nadał go użytkownikowi STUD. Użytkownik ADM odebrał przywilej użytkownikowi PRAC, w efekcie ani użytkownik PRAC, ani użytkownik STUD nie posiadają tego przywileju. PRAC nie może odebrać przywileju ADM, a STUD nie może odebrać przywileju ani ADM, ani PRAC.

Odbieranie przywilejów systemowych i ról nie powoduje efektu kaskady, gdyż przywileje nadane WITH ADMIN OPTION nie są hierarchiczne, odebranie przywileju WITH ADMIN OPTION nie jest wtedy kaskadowe.

2.3.4. Obsługa ról na poziomie systemu operacyjnego

Podobnie jak użytkownicy, role mogą być weryfikowane oraz przydzielane za pośrednictwem systemu operacyjnego komputera. Role kontrolowane przez system operacyjny są nadawane podczas połączenia. Zabezpieczenia na poziomie systemu operacyjnego zaimplementowano w różny sposób w różnych systemach operacyjnych, na przykład w Unixie odpowiednie role odpowiadają odpowiednim grupom użytkowników.

Aby zarządzać rolami z poziomu systemu operacyjnego, należy:

- utworzyć odpowiednie role w bazie Oracle'a,
- nadać utworzonym rolom odpowiednie przywileje,
- ustawić parametr inicjalizacyjny OS_ROLES na TRUE (albo bezpośrednio w systemie operacyjnym, albo w pliku inicjalizacyjnym),
- nadać użytkownikowi, któremu mają być przydzielone role z poziomu operacyjnego, prawa w systemie operacyjnym do korzystania z ról,

Format przywileju SO:

ORA_SID_ROLE[_D][A], gdzie:

- D - rola jest domyślna dla użytkownika,
- A - rola została nadana z opcją WITH ADMIN OPTION,
- SID – identyfikator bazy danych,
- ROLE – nazwa roli.

Przykład

Konto użytkownika w systemie operacyjnym musi mieć zdefiniowane role w swoim profilu. Zakładając, że w systemie operacyjnym i w bazie danych Oracle'a zdefiniowane jest konto użytkownika - *kadrowy*, weryfikowane przez system operacyjny (patrz punkt 2.2), następujący zapis w pliku ../etc/group:

```
ORA_NowaBaza_ROLA1 :: 20020 : kadrowy
ORA_NowaBaza_ROLA2_A :: 20030 : kadrowy
ORA_NowaBaza_ROLA3_D :: 20040 : kadrowy
ORA_NowaBaza_ROLA4_DA :: 20050 : kadrowy
```

oznacza, że po połączeniu się z bazą (poleceniem *sqlplus /*) użytkownikowi nadane są wszystkie cztery role. Rola3 i rola4 są domyślne, rola2 i rola4 są dostępne z opcją ADMIN OPTION.

2.4. Profile i limity zasobów

Każdemu użytkownikowi nadawany jest jakiś profil, który określa ograniczenia do korzystania z różnych zasobów systemu dostępnych dla danego użytkownika. Jeśli przy tworzeniu użytkownika nie zostanie przypisany mu żaden profil domyślnie zostanie

przydzielony mu profil *Default*. Dotyczy on wszystkich zasobów nie ujętych w ograniczeniach innych profili. Początkowo profil domyślny nie ma ograniczeń na zasoby. Może być jednak zmieniany, tak aby żaden użytkownik nie miał nieograniczonego dostępu do zasobów.

Profile nadawane są użytkownikom w celu ograniczenia ilości zasobów systemowych dostępnych dla użytkownika, uniemożliwienia użytkownikom operacji zbyt mocno obciążających zasoby, uproszczenia zarządzania zasobami, przypisania indywidualnych lub domyślnych limitów użytkownikom, specyfikacji ograniczenia zasobów dla grupy użytkowników, zarządzania zasobami w dużych, złożonych systemach, ograniczenia użycia zasobów na poziomie sesji lub odwołania (call), odłączenia po odpowiednio długim czasie oczekiwania w sesji oraz wylogowania użytkowników jeśli nie pracują (idle).

Wprowadzanie ograniczeń może być na poziomie [3]: sesji, wywołania lub na obydwu poziomach. Zasoby kontrolowane na poziomie sesji to:

CPU_PER_SESSION, SESSIONS_PER_USER, CONNECT_TIME, IDLE_TIME, LOGICAL_READS_PER_SESSION, PRIVATE_SGA

Zasoby kontrolowane na poziomie wywołania to:

CPU_PER_CALL LOGICAL_READS_PER_CALL

Do tworzenia profili służy polecenie CREATE PROFILE, a do modyfikacji ALTER PROFILE o takiej samej składni:

```
CREATE PROFILE profil LIMIT
            SESSIONS_PER_USER          }
            CPU_PER_SESSION            }
            CPU_PER_CALL               }   Integer
            CONNECT_TIME               }   UNLIMITED
            IDLE_TIME                  }   DEFAULT
            LOGICAL_READS_PER_CALL     }
            LOGICAL_READS_PER_SESSION  }
            COMPOSITE_LIMIT            }
            PRIVATE_SGA integer (rozmiar) w KB lub MB
                        UNLIMITED
                        DEFAULT;
```

Przykłady tworzenia i modyfikacji profili

```
CREATE PROFILE pracownik LIMIT
            SESSIONS_PER_USER 4
            CPU_PER_CALL UNLIMITED
            IDLE_TIME 40;

ALTER PROFILE pracownik LIMIT
            SESSIONS_PER_USER 3
            CPU_PER_SESSION 20000
            IDLE_TIME 20
            LOGICAL_READS_PER_CALL 1500;
```

Przydzielanie profili następuje w momencie tworzenia lub zmiany definicji użytkownika (opisane w punkcie 2.1).

Przykład

```
CREATE USER asystent IDENTIFIED BY 0lasyst89
DEFAULT  TABLESPACE asyst1
TEMPORARY TABLESPACE temp
PROFILE pracownik;

ALTER USER doktor
PROFILE pracownik;
```

Skutki przydzielania profilu nie są widoczne w bieżącej sesji. By móc przydzielić profil, należy mieć przywilej ALTER USER. Profile mogą być przydzielane tylko użytkownikom podczas tworzenia lub modyfikacji definicji ich kont.

Można nakładać ograniczenia na jeden konkretny zasób lub stosować ograniczenie złożone, które ogranicza użycie wielu zasobów.

Jest ono sumą ważoną ograniczeń: CPU_PER_SESSION, CONNECT_TIME, PRIVATE_SGA, LOGICAL_READS_PER_SES-SION, określoną w jednostkach usługowych i może być użyte z ograniczeniem indywidualnym.

Dla ograniczeń złożonych można określać koszt zasobu poleceniem: ALTER RESOURCE COST:

```
ALTER RESOURCE COST  CPU_PER_SESSION integer
                     CONNECT_TIME integer
                     LOGICAL_READS_PER_CALL integer
                     LOGICAL_READS_PER_SESSION  integer
                     PRIVATE_SGA  integer
```

integer - waga zasobu

Dla zmiany kosztu zasobu potrzebny jest przywilej ALTER RESOURCE COST.

Można również obliczyć całkowity koszt zasobów dla sesji według wzoru:

```
Total Resource Cost =  CPU_used * weight +
                       Connect_time_used * weight +
                       Logical_reads_used_in_session * weight +
                       Private_SGA_used * weight
```

Przykład ustawienia kosztów zasobów i obliczenia całkowitego kosztu zasobów

```
ALTER RESOURCE COST CPU_PER_SESSION 10
                    LOGICAL_READS_PER_SESSION 25
                    PRIVATE_SGA 5;

Total Resource Cost = CPU_used * 10 +
                      Logical_reads_used_in_session * 25 +
                      Private_SGA_used * 5
```

Przykład: `CREATE PROFILE stud LIMIT COMPOSITE_LIMIT 5000000`

Kontrola użycia zasobów jest włączona, gdy parametr RESOURCE_LIMIT ma wartość TRUE. Wartość tego parametru można ustawić na dwa sposoby:

- z poziomu systemu operacyjnego komputera – przed ustawieniem parametru należy bazę danych zamknąć, po ustawieniu ponownie wystartować; stosowany tylko w sytuacjach, gdy baza może zostać zamknięta,

– za pomocą komendy ALTER SYSTEM, na otwartej bazie; takie ustawienie parametru utrzymuje się do następnej zmiany lub do zamknięcia bazy. Sposób ten stosowany jest w sytuacjach, gdy baza nie może być zamknięta lub zmiana ma być tymczasowa.

Przykład

```
ALTER SYSTEM
      SET RESOURCE_LIMIT = TRUE
                           FALSE
```

2.5. Obserwacja bazy danych

Obserwacja bazy danych jest warunkiem przynależności systemu informatycznego do klasy bezpieczeństwa C2. Obserwacja bazy może pomóc w ujawnieniu daty i czasu nielegalnego dostępu do danych. Istnieje wiele typów podejrzanych operacji, które można wykryć na podstawie obserwacji, np. próby podłączenia się jako inny użytkownik, wpisujący różne kombinacje hasła w krótkim czasie, czy nielegalne zmiany w tablicach zawierających poufne informacje lub nielegalne zakładanie lub usuwanie obiektów. Obserwacja może być również wykorzystana w celu zebrania statystyk na temat użytkowania bazy, np. które tabele są najczęściej używane, jaka była największa liczba podłączonych użytkowników do bazy danych [2].

Wyróżnia się 3 typy obserwacji:

– obserwacja zdań – monitoruje użycie konkretnego zdania SQL,

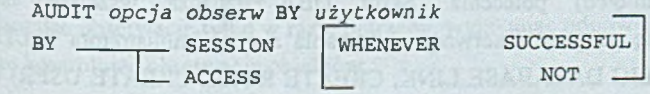
– obserwacja przywilejów – monitoruje wykorzystanie przywilejów systemowych, jest selektywną obserwacją zdań wykonywanych przez użytkownika uprawnionego danym przywilejem; można prowadzić taką obserwację dla wszystkich użytkowników bądź dla wybranej grupy użytkowników (obserwacja przywilejów jest bardziej szczegółowa, ponieważ rejestruje określony typ zdań SQL, a nie grupę zdań),

– obserwacja obiektów – monitoruje wykonanie konkretnych zdań na konkretnych obiektach; obserwacja rejestruje polecenia DML (również zapytania) dla dowolnego obiektu, a także zdania GRANT i REVOKE dotyczące danego obiektu.

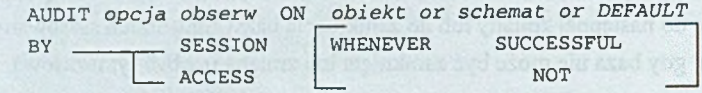
W skład obserwacji wchodzi śledzenie podejrzanych operacji, monitorowanie akcji mających miejsce w bazie i zbieranie danych o akcjach mających miejsce w bazie danych.

Do ustawiania obserwacji służy polecenie AUDIT.

Polecenie AUDIT dla obserwacji zdań i przywilejów:



Polecenie AUDIT dla obserwacji obiektów:



Gdzie:

- *opcja obserw* - opcja obserwacji zdań, przywilejów lub obiektów,
- *BY użytkownik* - obserwacja tylko dla tego użytkownika (lub grupy użytkowników); domyślnie prowadzona jest obserwacja dla wszystkich użytkowników,
- *BY SESSION* - wstawiany jest tylko jeden zapis dla obiektu bazy danych dla danej sesji, niezależnie od tego, ile zdań tego samego typu jest wykonywanych,
- *BY ACCESS* - wstawiany jest zapis za każdym razem, gdy obserwowane zdanie SQL jest wykonywane. Jeżeli ustawiona jest obserwacja zdań DDL, Oracle będzie prowadził obserwację BY ACCESS, niezależnie od domyślnego ustawienia. Dla zdań nie - DDL domyślnie jest BY SESSION,
- *WHENEVER* - zapis dla zdań wykonanych:
  - *SUCCESSFUL* z powodzeniem,
  - *NOT* bez powodzenia.

Opcje obserwacji odnoszą się do sesji pojawiających się po włączeniu obserwacji. Obserwacja nie obejmuje sesji, w której została włączona. Do wydania komendy AUDIT lub NOAUDIT potrzebny jest przywilej SYSTEM AUDIT.

Przykłady

Obserwacja zdań SQL:

```
AUDIT dba BY ACCESS WHENEVER NOT SUCCESSFUL;
AUDIT system grant, system audit BY stud1, stud2;
```

Obserwacja przywilejów:

```
AUDIT select any table BY stud1;
AUDIT  alter any table, alter any procedure
BY stud1 BY ACCESS WHENEVER NOT SUCCESSFUL;
```

Obserwacja obiektów:

```
AUDIT execute ON dochody BY SESSION;
AUDIT delete ON dochody BY ACCESS;
AUDIT grant ON  zęspoły BY SESSION WHENEVER SUCCESSFUL;
```

Jeżeli pominie się klauzulę WHENEVER, Oracle będzie prowadził obserwację operacji udanych i nieudanych. Podczas obserwacji nieudanych operacji nie są rejestrowane niepoprawne (np. składniowo) polecenia. Skrót DBA oznacza wszelkie akcje administracyjne. Używa się go przy obserwacji wykonania akcji administratora (ALTER SYSTEM, CREATE PUBLIC DATABASE LINK, CREATE ROLE, CREATE USER) lub przy ustalaniu, jak często powyższe operacje są nieudane z powodu braku miejsca na dysku

lub błędu systemu. Opcja SYSTEM GRANT włącza obserwację nadawania/odbierania przywilejów systemowych dla wszystkich lub dla wybranej grupy użytkowników. Opcja SYSTEM AUDIT włącza obserwację włączania/wyłączania obserwacji poleceń dla wszystkich lub dla wybranej grupy użytkowników. Opcje WHENEVER SUCCESSFUL / NOT SUCCESSFUL / BY ACCESS obserwacji CREATE / ALTER są używane do wykrycia modyfikacji atrybutów konta użytkownika.

Zapis obserwacji przechowuje informacje wygenerowane podczas obserwacji poleceń, przywilejów lub obiektów. Zapis obserwacji jest przechowywany w tabeli słownika danych SYS.AUD$. Podstawowe informacje zawarte w tabeli SYS.AUD$ to: nazwa użytkownika, wpisującego polecenie, kod akcji (liczba) oznaczający typ wykonanego polecenia, obiekt(y), do którego odnosiło się polecenie oraz data i czas wykonania polecenia.

Mimo że komendy AUDIT i NOAUDIT mogą być użyte w dowolnym momencie, to czy i gdzie będzie tworzony zapis obserwacji, zależy od ustawienia parametru pliku startowego - AUDIT_TRAIL [3].

AUDIT_TRAIL = value, gdzie *value: DB, OS, NONE*

- *DB* - kieruje zapis obserwacji do bazy danych,
- *OS* - kieruje zapis obserwacji do pliku systemu operacyjnego przy każdym dostępie, opcja BY SESSION działa wtedy jak BY ACCESS (jeżeli jest to dopuszczalne na danej platformie sprzętowej),
- *NONE* - wyłącza zapis obserwacji (wartość domyślna).

Zmiana parametru AUDIT_TRAIL daje efekt dopiero po starcie instancji ze zmienionym plikiem startowym. Opcje obserwacji można ustawiać za pomocą komend AUDIT i NOAUDIT, a włączanie i wyłączanie obserwacji odbywa się globalnie na poziomie bazy danych, tzn. jeśli parametr AUDIT_TRAIL nie jest ustawiony, to nie generuje się zapis obserwacji. Zapis obserwacji powstaje podczas fazy wykonania zdania SQL. Obserwacja jest niezależna od transakcji użytkownika; jeśli transakcja zostanie wycofana, zapis obserwacji pozostanie. Obserwacja nie rejestruje zmienianych wartości. Aby to zrobić, należy użyć wyzwalaczy zapisujących usuwaną/zmienianą wartość do specjalnej tabeli.

Monitorowanie rozmiaru i sposobu wzrostu objętości pliku zapisu obserwacji zabezpiecza dysk i chroni bazę przed pogorszeniem wydajności. Jeśli zapis obserwacji nie może być stworzony z powodu braku miejsca, obserwowane zdania będą zwracać błąd.

Wzrost zapisu obserwacji zależy od ilości włączonych opcji oraz od częstotliwości wykonywania obserwowanych zdań. Jak zatem kontrolować wzrost zapisu obserwacji? Włączając obserwacje tylko w razie potrzeby, wybierając odpowiednie opcje obserwacji oraz ostro kontrolując obserwacje obiektów.

Tylko właściciel obiektu lub użytkownik posiadający przywilej AUDIT ANY może obserwować obiekt. Aby zagwarantować, że jedynym użytkownikiem, który może włączać obserwacje obiektów, jest administrator kontroli dostępu, trzeba zapewnić dwa warunki:

- tylko administrator kontroli dostępu ma przywilej AUDIT ANY, jest on właścicielem wszystkich obiektów,
- żaden aktualny użytkownik nie ma swoich obiektów, wszystkie obiekty zawarte w schematach nie mają aktualnych właścicieli (tzn. właściciele schematów nie mają przywileju CREATE SESSION).

Podsumowując, można stwierdzić, że omówiony sposób funkcjonowania systemu bazy danych Oracle7 jest zgodny z formalną definicją modelu Dyskrecjonalnej (uznaniowej) Kontroli Dostępu DAC (Discretionary Access Control) modelu ochrony systemu informatycznego wg wymagań TCSEC/TDI. Uznaniowość polega na tym, że odpowiednie uprawnienia muszą być związane z użytkownikiem, by użytkownik miał dostęp do obiektu (administrator, właściciel), a uprzywilejowany użytkownik może przekazywać innym użytkownikom uprawnienia zgodnie z własnym uznaniem.

Model DAC kontroli dostępu do obiektów bazy danych wraz z dodatkowymi właściwościami użytkowymi bazy danych, takimi jak: zabezpieczanie integralności informacji, zapewnienie kontrolowanego dostępu do wyróżnionych z punktu widzenia ochrony danych podzbiorów zasobów systemu oraz śledzenie akcji obiektów aktywnych w stosunku do pasywnych w celu tworzenia i analizy kroniki dostępu, w całości wyczerpują wymagania stawiane przed systemami klasy C2.

3. Przypadki, w których ochrona bezpieczeństwa systemu informatycznego zapewniana mechanizmami właściwymi dla klasy C2 nie jest wystarczająca

Jest wiele przypadków, gdy ochrona bezpieczeństwa systemu informatycznego zapewniana mechanizmami właściwymi dla klasy C2 okazuje się niewystarczająca.

Jednym z nich jest sytuacja, gdy realizowana w systemie polityka bezpieczeństwa nie dopuszcza możliwości przekazywania uprawnień związanych z dostępem do określonych informacji. Jest to związane z faktem (opisanym w punkcie 2.3.3), że nad raz przekazanymi uprawnieniami tracimy kontrolę.

Kolejny przypadek może wystąpić w sytuacji, gdy w systemie przetwarza się informacje wymagające zróżnicowanego poziomu ochrony (ogólnie dostępne, poufne, tajne). Mechanizmy ochronne klasy C2 pozwalają tylko na fizyczny podział bazy danych na

„jednorodne" kawałki. Mogą wystąpić tu jednak pewne niedogodności, np. kłopoty z administrowaniem wieloma systemami czy utrudniony dostęp do danych zgromadzonych w systemie. W sytuacji gdy rozwiązania te nie mogą być zastosowane, pozostaje używanie bardziej zaawansowanych narzędzi.

3.1. Metody ochrony o zróżnicowanym poziomie poufności

Systemy zawierające takie metody nazywane są systemami MLS (Multilevel Security); wykorzystują one technikę etykietowania informacji (information labeling). Etykieta jest trwale przypisaną do każdej przechowywanej w systemie informacji cechą, która określa wymagany dla informacji poziom ochrony [5]. Podstawowa struktura etykiety ochronnej może zawierać dwa składniki:

POUFNE: Projekt X

element hierarchicznej klasyfikacji — opcjonalna kategoria klasyfikacyjna

Treść etykiety zapisywana jest automatycznie w dodatkowej kolumnie o nazwie ROWLABEL każdej tablicy bazy danych. Ponieważ każdy obiekt relacyjnej bazy danych (tabela, kolumna, użytkownik) jest zapisem w jakiejś tabeli (np. systemowej), może być niezależnie etykietowany.

Technika etykietowania informacji wykorzystywana jest przez model MAC (Mandatory Access Control) ochrony bezpieczeństwa systemu informatycznego. Polega on na negocjowaniu dopuszczalności operacji w bazie danych na podstawie porównania uprawnień (mandatu) użytkownika wyrażonego przez jego etykietę z etykietą obiektu, na którym miała być wykonana operacja.

Ponieważ model DAC ochrony w całości realizowany jest przez mechanizmy zarządzania bazą danych (np. Oracle7), można te mechanizmy budować w środowisku dowolnego systemu operacyjnego. Dowolność taka nie jest możliwa w przypadku modelu MAC, gdyż polityka bezpieczeństwa MAC jest wynikiem współdziałania systemu zarządzania bazą danych, takiego jak np. Trusted Oracle 7, i systemu operacyjnego komputera.

Tylko właściwy system operacyjny (klasy MLS) komputera pozwala na definiowanie etykiet ochronnych oraz przekazuje systemowi bazy danych zakres uprawnień (clearance) użytkownika bazy danych. Procedura negocjowania dopuszczalności operacji odbywa się między systemem zarządzania bazą danych i systemem operacyjnym. Wykorzystywane są tu dość złożone reguły dominacji etykiet: etykieta ochronna dominuje nad inną, jeżeli jej część klasyfikacyjna stoi wyżej w hierarchii, a opcjonalne kategorie klasyfikacyjne tworzą nadzbiór kategorii z etykiety zdominowanej.

Zgodnie z zasadami polityki ochronnej MAC realizowanej przez systemy operacyjne klasy MLS użytkownik może pisać wyłącznie na poziomie ochrony równym poziomowi własnej etykiety i czytać na poziomie własnej etykiety i wszystkich etykiet przez nią zdominowanych.

System zarządzania bazą danych Trusted Oracle7 posiada możliwość wyboru jednego z dwóch trybów pracy: OS MAC Mode (opisany wcześniej), w którym oprogramowanie Oracle'a restrykcyjnie przestrzega reguł polityki bezpieczeństwa systemu operacyjnego, oraz trybu RDBMS MAC Mode - w którym uprzywilejowany proces Oracle może w sposób kontrolowany omijać ograniczenia wyznaczane przez system operacyjny [5].

W systemie Trusted Oracle7 mechanizmy kontroli wykraczające poza ograniczenia OS MAC zbudowane są wokół trzech dodatkowych uprawnień systemowych:

- WRITEUP - prawo pisania na poziomie powyżej własnej etykiety,
- WRITEDOWN - prawo pisania poniżej poziomu ochrony własnej etykiety,
- READUP - prawo czytania powyżej poziomu własnej etykiety.

Istnieje możliwość wyboru pracy w jednym z dwóch trybów Trusted Oracle7. Wybór trybu OS MAC pozwala zachować integralną politykę bezpieczeństwa w ramach całego systemu informatycznego, a w przypadku dysponowania odpowiednim środowiskiem komputerowym z klasą ochrony powyżej B1, wykorzystywać płynące z tego faktu korzyści. Natomiast wybór trybu RDBMS MAC w szerszym stopniu uwzględnia specyfikę administrowania bazą danych (np. w zakresie przeklasyfikowywania informacji).

LITERATURA

1. Rodgers U.: ORACLE przewodnik projektanta baz danych. WNT, Warszawa 1995.
2. ORACLE7 Server Concepts Manual. Oracle Corporation, 1992.
3. ORACLE7 Server Administrator's Guide. Oracle Corporation.
4. Dec L.: Bezpieczeństwo danych w systemach Oracle7 (część I). Wiadomości Oracle.
5. Dec L.: Bezpieczeństwo danych w systemach Oracle7 (część II). Wiadomości Oracle.
6. Beynon-Davies P.: Systemy baz danych. WNT, Warszawa 1998.
7. Ullman J. D.: Database and knowledge–base systems. Computer Science Press, 1988.

Recenzent: Dr hab. inż. Stanisław Wołek, Prof.Pol.Rzeszowskiej

Wpłynęło do Redakcji 30 marca 1999 r.

Abstract

The primary purpose of database security is protecting information, which is included in database. Oracle's strategy for protecting data is to provide comprehensive discretionary access control. Discretionary access control regulates all user access to named objects. User access to objects is controlled through privileges. Privileges are granted to users at the discretion of other users. A privilege is a right to execute a particular type of SQL statement or to access another user's object. There are two distinct categories of privileges: system and object privileges. Oracle provides for easy and controlled privilege management through the use of roles. Roles are named groups of related privileges that are granted to user or other roles. Each Oracle database has a list of valid database users. To prevent unauthorised use of a database username, Oracle provides user validation via two different methods: authentication by the operating system and authentication by the associated Oracle database. Each user is associated with a default and a temporary tablespace and also each user can be assigned a tablespace quota for any tablespace of the associated database.

The resource limit feature of Oracle is very useful in large, multiuser systems. The security administrator can prevent the uncontrolled consumption of valuable system resources such as CPU time. Resource limits are managed with user profiles. A profile is a named set of resource limits that can be assigned to a user.