DBSecWorx **White Paper**

Default Insecure
The Oracle Database DEFAULT Profile

Background

It isn't Oracle's fault that the overwhelming majority of its customers deploy their databases with most, if not all, users granted the DEFAULT profile. Oracle has for decades developed and solid the enterprise database with the most solid set of security features available. Breaking into a Microsoft SQL Server database is a trivial exercise compared with breaking into a properly configured Oracle Database. But there is a trap in my wording … I said "properly configured" … and Oracle's customers have deployed many hundreds of thousands of databases, persisting their most valuable information, and not properly configured them.

To learn about Profiles, the Oracle Database Administrators Guide refers readers to the Oracle Database Security Guide. The security guide contains the following statements:

"A profile is a set of limits, defined by attributes, on database resources and password access to the database. The profile can be applied to multiple users, enabling them to share these attributes. You can specify a profile when you create a user. The PROFILE clause of the CREATE USER statement assigns a user a profile. If you do not specify a profile, then Oracle Database assigns the user a default profile."

Almost everything you need to know is about the security aspects of the DEFAULT Profile is contained within that paragraph. The paragraph seems very reassuring but it should scare any DBA with an IQ higher than room temperature like a dark alley on the wrong side of town: Unfortunately, it doesn't.

Here's a rewrite of that paragraph that I think better captures what DBAs and CISOs need to know.

"A profile is a set of limits, defined by limits, on databases resources and password attributes. If your database contains information more valuable than your mother's cookie recipes you should create multiple profiles and based on your specific needs, reassign all users to those profiles, and break the default profile so that it cannot be exploited to attack your systems."

The following, annotated, listing shows the DEFAULT profile active in almost every Oracle database for almost every user.

```
RESOURCE_NAME                    LIMIT
-------------------------------- ----------------
COMPOSITE_LIMIT                  UNLIMITED (5)
CONNECT_TIME                     UNLIMITED (2)
CPU_PER_CALL                     UNLIMITED (5)
CPU_PER_SESSION                  UNLIMITED (5)
FAILED_LOGIN_ATTEMPTS            10        (7)
IDLE_TIME                        UNLIMITED (3)
INACTIVE_ACCOUNT_TIME            UNLIMITED (4)
LOGICAL_READS_PER_CALL           UNLIMITED (5)
LOGICAL_READS_PER_SESSION        UNLIMITED (5)
PASSWORD_GRACE_TIME              7
PASSWORD_LIFE_TIME               180       (6)
PASSWORD_LOCK_TIME               1         (8)
PASSWORD_REUSE_MAX               UNLIMITED (6)
PASSWORD_REUSE_TIME              UNLIMITED (6)
PASSWORD_VERIFY_FUNCTION         NULL      (8)
PRIVATE_SGA                      UNLIMITED (5)
SESSIONS_PER_USER                UNLIMITED (1)
```

Listing 1

And now let's look at that profile with an eye focused clearly on database security:

In blue: A single user, or a collection of users can log into the Oracle Database an unlimited number of time (1), they can stay connected forever (2), whether they are doing any work or not (3). They can never connect and their account will never be locked (4). If they do connect they can use an unlimited amount of cpu and memory resources (5) and a single user can create a denial of service attack but exhausting all available resources.

In green: Each user must change their password every 180 days but they can reuse the same password for all of eternity (6). If the user forgets their password, they will get 10 attempts to remember what it is (7). But if they cannot remember the account will be unlocked one minute later and they can keep on trying (8). If the "user" with the incorrect password is a bank of application servers, they can keep hammering away with the wrong password forever creating a denial of service attack for any user or application server that has the correct password. And the password the user has can be as simple as a single byte and need not container upper and lower case characters, number, special characters, or be of any minimum length (8).

Do I have your attention now?

None of what I have written is in any way negative toward Oracle Corp. Oracle has done everything it can to make its product backward compatible and everything it can to make it easy for a customer to properly secure their product. The problem is that the customers are not making the required configuration changes.

Defining Secure Profiles

To define secure profiles requires analyzing the needs of the database's users. An individual producing ad hoc reports needs a single database connection, a DBA may need two or three simultaneous connections in order to execute code and trace behavior. An application connecting from tens or hundreds of application servers will likely require many hundreds or thousands of simultaneous connections.

For each user class a profile should be designed that provides all of the resources required for that user to perform their job function, but simultaneously put a fence around those profile limits that, if exceeded, put the system at risk.

DEFAULT Profiles Risk Mitigation

After defining database specific profiles all users should be moved to those profiles so that no user, not even SYS, is using Oracle's built-in default.

That done, the next task is to remove the risks from the DEFAULT profile. You likely remember the sentence above from the online Oracle docs.

"*If you do not specify a profile, then Oracle Database assigns the user a default profile.*"

What this means to a security conscious DBA is that if any person or process, perhaps a SQL*Injection attack or a glogin exploit is used to create a new user that user is going to get the DEFAULT profile and be able to attack the system. What we need is a default profile that makes a user not defined by these means incapable of pursuing their attack. How can we do that?

Review the default in Listing 1 above with an eye toward minimizing the danger. How usable would a user connection be if the DEFAULT profile limited the user's CPU_PER_CALL to 1? PRIVATE_SGA to 1? LOGICAL_READS_PER_CALL to 1? LOGICAL_READS_PER_SESSION to 1?

Plus, any user created with the DEFAULT profile will stand out immediately as suspicious.

The DBSecWorx Solution

Our team knows that most DBA and security teams do not have a lot of spare time in their day. Having read this White Paper, they might really want to implement the recommendations contained here but know that there would be required research, code to write, and the need to test their solution first in development, then in QA, and then finally deploy the solution in production and DR.

Contact us today and we can provide you and your team with the details you need to evaluate DBMZ_PROFILE, our compiled Expert System, that automates the entire process.

Contact us today to learn more: damorgan@dbsecworx.com or text +1 612-240-3538