ORACLE RDMBS - Unauthenticated Session Exhaustion, Denial of Service Attack

Background

Oracle is a widely deployed DBMS. Clients use a protocol called TNS to communicate to the Oracle server. Protocol messages are used for session setup, authentication and data transfer. The standard authentication mechanism is a two-step challenge-response protocol that requires a client to supply a valid pair of user name and password.

Scope

Imperva's Application Defense Center is conducting an extensive research of the TNS protocol and its implementation. As part of the research the team has identified vulnerability in Oracle's session management mechanism that allows an unauthenticated attacker to exhaust all available sessions and deny database access from legitimate users.

Findings

An unauthenticated attacker can use a single TCP connection to the database server to exhaust all available database sessions. Thus legitimate users will not be able to connect to the database server until the attacker's TCP connection is closed.

Details

The Oracle DBMS default authentication mechanism is a two-step challenge response protocol. In the first step the client sends a message containing the account name (AKA username, schema name). The server responds with a challenge and the client then sends a second message with the account name and a response to the challenge.

It turns out that the server allocates a session after the first request is made. More over, if the client does not respond to the challenge but rather resends the first message, the server allocates a new session. Thus an attacker can open a TCP connection to database server, send the introductory TNS message (those do not require authentication but rather negotiate the data representation for the rest of the session) and then repeatedly send the first authentication message (function code 0x52 or 0x76, both equally apply) through that same connection, the attacker will exhaust all the available sessions in the database server.

Notice that in Oracle versions prior to 10g the attacker must know the name of an existing database account in order to execute this attack. However, in Oracle version 10g (both releases) that attacker does not even need to know the name of an existing account.

An additional issue to notice regarding this attack is that it leaves no trace in the database audit trail, because the audit record for a connection operation (either success or failure) is not written until the operation is completed.

Exploit

Repeatedly send function calls 0x76 and 0x52 through an unauthenticated connection to the database.

Tested Versions

Vulnerable

Oracle 8i (8.1.7.x.x)

Oracle 9i (9.2.0.7)

Oracle 10g Release 1 (10.1.0.4.2)

Oracle 10g Release 2 (10.2.0.1.0)

Not Vulnerable

Vendor's Status

Fixed in 11g and 10gR2

Workaround

None