

30 October 2008

Different ways to guess Oracle database SID

Digital Security Research Group (DSecRG)

Alexander Polyakov

a.polyakov@dsec.ru

<http://dsecrg.ru>

Content

Introduction	3
A brief info about SID and SERVICE_NAME	4
Getting SID and SERVICE_NAME	4
Connecting to database with SERVICE_NAME	5
Connecting to database with SID	5
Problems with getting SID and SERVICE_NAME in new versions of Oracle database	5
Guessing SID	7
Trying default SID's	7
Test for typical SID's	8
Guessing a SID using Dictionary	8
Brute force	9
Searching SID and SERVICE_NAME in third-party applications	11
Oracle Enterprise Manager Control	12
Oracle Application Server	13
Oracle XDB	15
SAP	16
SAP Web Application Server Default administration page	16
SAP Web Application Server non-existent page	17
SAP RFC	17
SAP SID Bruteforcing	18
Getting SID using vulnerable Web Application	19
Getting database <i>SID</i> using additional rights on target system	20
Getting SID using some rights on target server	20
Getting SID using operating system account on server	20
Getting SID using FTP account on server	20
Getting SID using MsSQL account on server	21
Getting SID using list of services	22
Getting database SID using registry key HKLM\SOFTWARE\ORACLE	23
Getting SID using directory listing	24
Getting SID using additional rights in Company's network	25
Getting SID from another database's	26
Getting SID from another servers in target information system	26
Sniffing database SID from network	27
Conclusion	29
Links	30

Introduction

Nowadays there is a lot of public information about Oracle security and a different vulnerabilities that hacker can use to get access to database. Standard attack scenario can be presented as a sequence of following actions:

- Listener Attacks (Buffer Overflow, Log-file)
- Guessing Password and username
- Guessing *SID*
- Database privilege escalation (PL/SQL Injections, Buffer Overflows, Cursor snarfing, Password hash bruteforce e.t.c.)
- Access to OS (*Extproc, Java, UTL_FILE, DBMS_LOB*)
- Install rootkit/backdoor
- Cleaning log/audit files (*SYS.AUD\$* etc)

Many of these steps are good explained in public resources. Default user accounts are a big known problem and there is much information about it. As for vulnerabilities there are only 10 percent of DBA's regularly installing Critical Patch Updates (According to Sentrigo's reports). Access to OS files and shell can be done using many different techniques such as *Extproc, Java, DBMS_JOB, UTL_FILE, DBMS_LOB* and others. As for rootkits and cleaning audit data, in this area hackers are one step behind DBA's.

In this information about Oracle security there is one area that is not so good explained as others. I am talking about getting Oracle *SID*. Without knowing Oracle database *SID* attacker cannot get access to database even if he know username and password. With Oracle 10g process of getting database *SID* is not so trivial as before. That's why I decided to research this area and write this document as a result of my researching.

In this whitepaper I collect all well-known ways to get database *SID* and add some new techniques.

A brief info about SID and SERVICE_NAME

Every instance of database identified by *SID* (*System IDentifier*). *SID* contains from alphanumeric symbols and stored in system environment variable *ORACLE_SID*. *SID* is using by network utilities to get remote access to database.

There is also a variable called *SERVICE_NAME* which is very similar to *SID* but not the same. *SERVICE_NAME* it is a new variable (defined in Oracle 8i). *SERVICE_NAMES* specifies one or more names for the database service to which this instance connects. You can specify multiple service names in order to distinguish among different uses of the same database.

If we can get database *SID* or *SERVICE_NAME* than we can try all other steps to access to database. For example if we know *SID* we can try to brute force database accounts.

Getting SID and SERVICE_NAME

Standard way to get database *SID* is *lsnrctl* utility with option “*services*”:

```
LSNRCTL> services
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

In program output we can see database *SID* (named *Instance*) and *SERVICE_NAME* (named *Service*). In this example we can see that database *SID* is “*ORCL*”.

Connecting to database with *SERVICE_NAME*

If we know *SERVICE_NAME* we can simply connect to database using *sqlplus* utility:

```
C:\ >sqlplus system/manager@192.168.40.33/orcl
SQL*Plus: Release 10.1.0.5.0 - Production on Tue Aug 26 17:18:23 2008

Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL>
```

Connecting to database with *SID*

To connect to database using *SID* we must firstly add a connection descriptor to configuration file *tnsnames.ora*.

```
ORCL_192.168.40.33 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.40.33) (PORT = 1521)))
    (CONNECT_DATA =
      (SID = ORCL)
      (SERVER = DEDICATED))
  )
```

After we define a connection string "*orcl_192.168.40.33*" and we can use it in *sqlplus*:

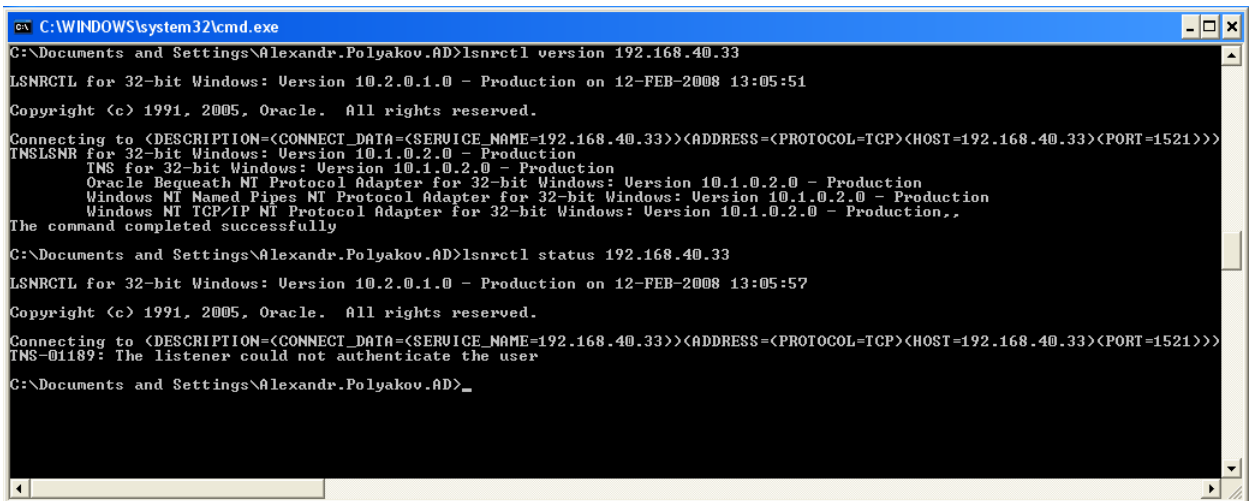
```
C:\ >sqlplus system/manager@orcl_192.168.40.33
SQL*Plus: Release 10.1.0.5.0 - Production on Tue Aug 26 17:18:23 2008

Copyright (c) 1982, 2005, Oracle. All rights reserved.
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options
SQL>
```

Problems with getting *SID* and *SERVICE_NAME* in new versions of Oracle database

In new versions of Oracle database starting from 10g R1 and higher, there is security option called *LOCAL_OS_AUTHENTICATION* used by default. This option deny using Listener

commands such as “services” and “status” from remote hosts. That’s why we cannot get *SID* and *SERVICE_NAME* using method described in “Getting SID and SERVICE_NAME”.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Alexandr.Polyakov.AD>lsnrctl version 192.168.40.33
LSNRCTL for 32-bit Windows: Version 10.2.0.1.0 - Production on 12-FEB-2008 13:05:51
Copyright (c) 1991, 2005, Oracle. All rights reserved.
Connecting to (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=192.168.40.33))<ADDRESS=(PROTOCOL=TCP)(HOST=192.168.40.33)(PORT=1521)>>)
TNSLSNR for 32-bit Windows: Version 10.1.0.2.0 - Production
TNS for 32-bit Windows: Version 10.1.0.2.0 - Production
Oracle Bequeath NT Protocol Adapter for 32-bit Windows: Version 10.1.0.2.0 - Production
Windows NT Named Pipes NT Protocol Adapter for 32-bit Windows: Version 10.1.0.2.0 - Production
Windows NT TCP/IP NT Protocol Adapter for 32-bit Windows: Version 10.1.0.2.0 - Production,,
The command completed successfully
C:\Documents and Settings\Alexandr.Polyakov.AD>lsnrctl status 192.168.40.33
LSNRCTL for 32-bit Windows: Version 10.2.0.1.0 - Production on 12-FEB-2008 13:05:57
Copyright (c) 1991, 2005, Oracle. All rights reserved.
Connecting to (DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=192.168.40.33))<ADDRESS=(PROTOCOL=TCP)(HOST=192.168.40.33)(PORT=1521)>>)
TNS-01189: The listener could not authenticate the user
C:\Documents and Settings\Alexandr.Polyakov.AD>_
```

Failed to launch "lsnrctl status" command in Oracle 10g

Also if administrator install a password on Listener version 9.2.0.6 or later we cannot execute commands such as “services” and “status” without knowing a listener password.

As we can see in new versions of Oracle database we must find new ways to get database *SID*. All known ways to get database *SID* we can divide on 3 groups:

- Guessing *SID*;
- Finding *SID* in third party applications;
- Getting *SID* using additional rights.

Guessing SID

The first thing if we cannot get *SID* using *lsnrctl* commands is try to guess the *SID*. There is 4 ways to guess database *SID*:

- Trying default *SID*'s;
- Trying widespread *SID*'s;
- Guess *SID* using a dictionary;
- Bruteforcing *SID*.

Trying default *SID*'s

It is a well-known that many administrators left default database *SID*. For example default *SID* when installing Oracle database is "ORCL". Default *SID* when installing Oracle 10G Express edition is "XE".

Oracle Database 10g Installation - Installation Method

Select Installation Method

Basic Installation
Perform full Oracle Database 10g installation with standard configuration options requiring minimal input. This option uses file system for storage, and a single password for all database accounts.

Oracle Home Location: Browse...

Installation Type:

Create Starter Database (additional 720MB)

Global Database Name:

Database Password: Confirm Password:

This password is used for the SYS, SYSTEM, SYSMAN, and DBSNMP accounts.

Advanced Installation
Allows advanced selections such as different passwords for the SYS, SYSTEM, SYSMAN, and DBSNMP accounts, database character set, product languages, automated backups, custom installation, and alternative storage options such as Automatic Storage Management.

Help Back Next Install Cancel

ORACLE

Screenshot from installation Oracle database 10g with standard *SID* "ORCL"

A List of all well-known default *SID*'s you can find in <http://www.red-database-security.com/scripts/sid.txt>. If you want to test default *SID* values automatically you can try utilities that will be described later.

Test for typical *SID*'s

Next step is to try some typical *SID* values such as public data about company. For example a company or department name can be used in *SID*. If we have a company name "Super Big Company" we can test *SID* like "SBC" or "SBCDB".

In our statistics about 10 percent of databases use such typical *SID*'s

Also some companies use *DNS/NETBIOS-name* of database server in *SID* value maybe with some modifications.

In our statistics 5 percent of databases use *SID* like *DNS/NETBIOS-name* and 8 percent of databases use *SID* like modified *DNS/NETBIOS-name*

Guessing a *SID* using Dictionary

If database *SID* is not from the default list and don't use typical values we can try to guess *SID* using a dictionary. There are some utilities that can be used to automate this process. Most popular utilities are presented in table named "*SID* guessing utilities".

Table "*SID* guessing utilities"

Utility	Author	Link
CsidGuess.py from Inguma	Joxean Koret	http://sourceforge.net/projects/inguma
ora-getsid, ora-brutesid from OAK	David Litchfield	http://www.vulnerabilityassessment.co.uk/oak.htm
oscanner	Patrik Karlsson	http://www.cqure.net/tools/oscanner_bin_1_0_6.zip
sidguess	Red database Security	http://www.red-database-security.com/software/sidguess.zip
sidguesser	Patrik Karlsson	http://inguma.sourceforge.net/index.php

When we use these utilities the main parameter is speed. To measure a speed of these utilities I decided to make two tests.

- First test is to measure time of working with list of standard *SID*'s (~600 values).
- Second test is to measure a time of guessing *SID* "ORCL" using bruteforce technique. Results of tests you can see in table "Speed testing".

Table "Speed testing"

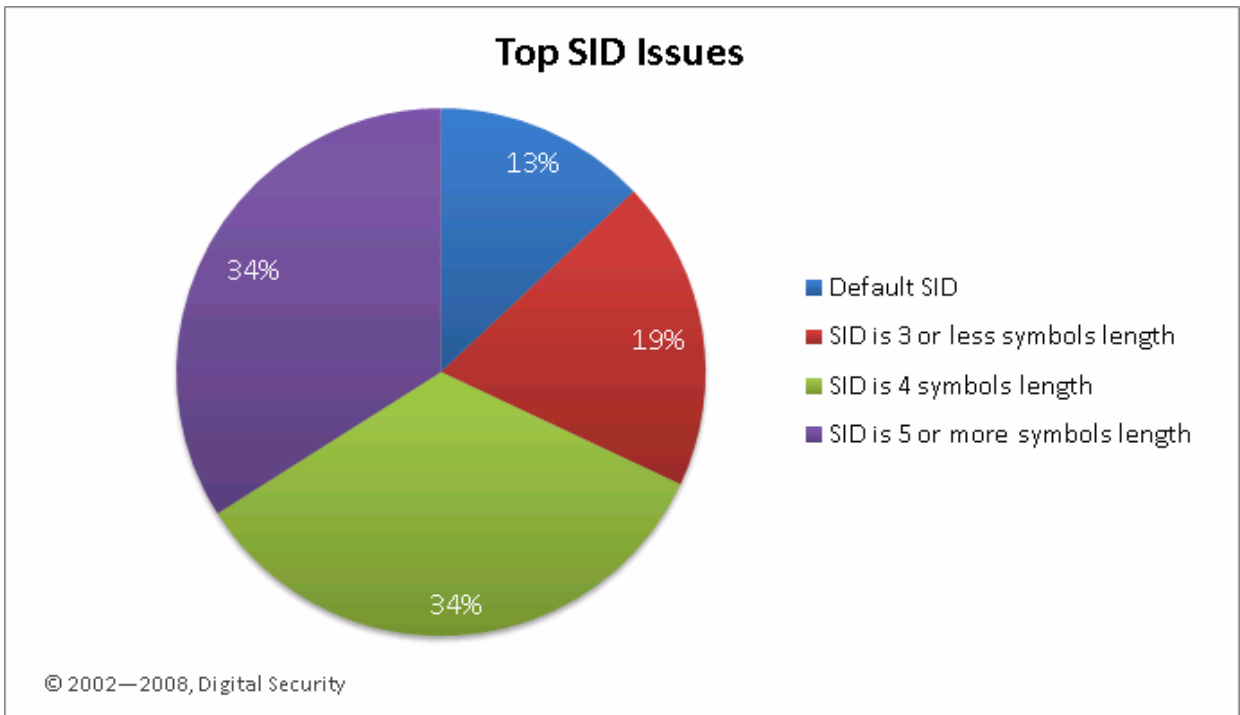
Utility	Bruteforce speed	Time to try all default <i>SID</i> values	Time for guessing <i>SID</i> "ORCL" using Bruteforce
Ora-brutesid	90 <i>SID</i> /sec	Not implemented	114 minutes
Ora-getsid	88 <i>SID</i> /sec	7 sec.	Not implemented
Oscanner	80 <i>SID</i> /sec	8 sec.	Not implemented
Sidguesser	71 <i>SID</i> /sec	10 sec.	Not implemented
Sidguess	11 <i>SID</i> /sec	58 sec.	Utility cannot finish working

Utility *Ora-getsid* have the best speed for guessing *SID* using dictionary as we can see in table.

Brute force

If we test all our dictionaries and *SID* is not found we have a last chance try to bruteforce *SID*. The best utility for bruteforcing *SID* is *ora-brutesid* (see table "Speed testing"). Using *ora-brutesid* we can test all 4-characters *SID*'s in 3 hours. If we have 5-character *SID* we will spend on it about 3 days. But it is still real. As a bonus – we can execute some parallel *ora-brutesid* processes to guess *SID* on different databases. It will save us time.

In our statistics of penetration testing big companies: 13 percent of databases use default *SID* values, 19 percent of databases *SID*'s are 3 characters length or less and 34 percent of databases *SID*'s are 4 characters length.



So it will take us about 3 hours to guess database *SID* with 66% success probability
(13%+19%+34% = 66% ~ 2/3)

Searching SID and SERVICE_NAME in third-party applications

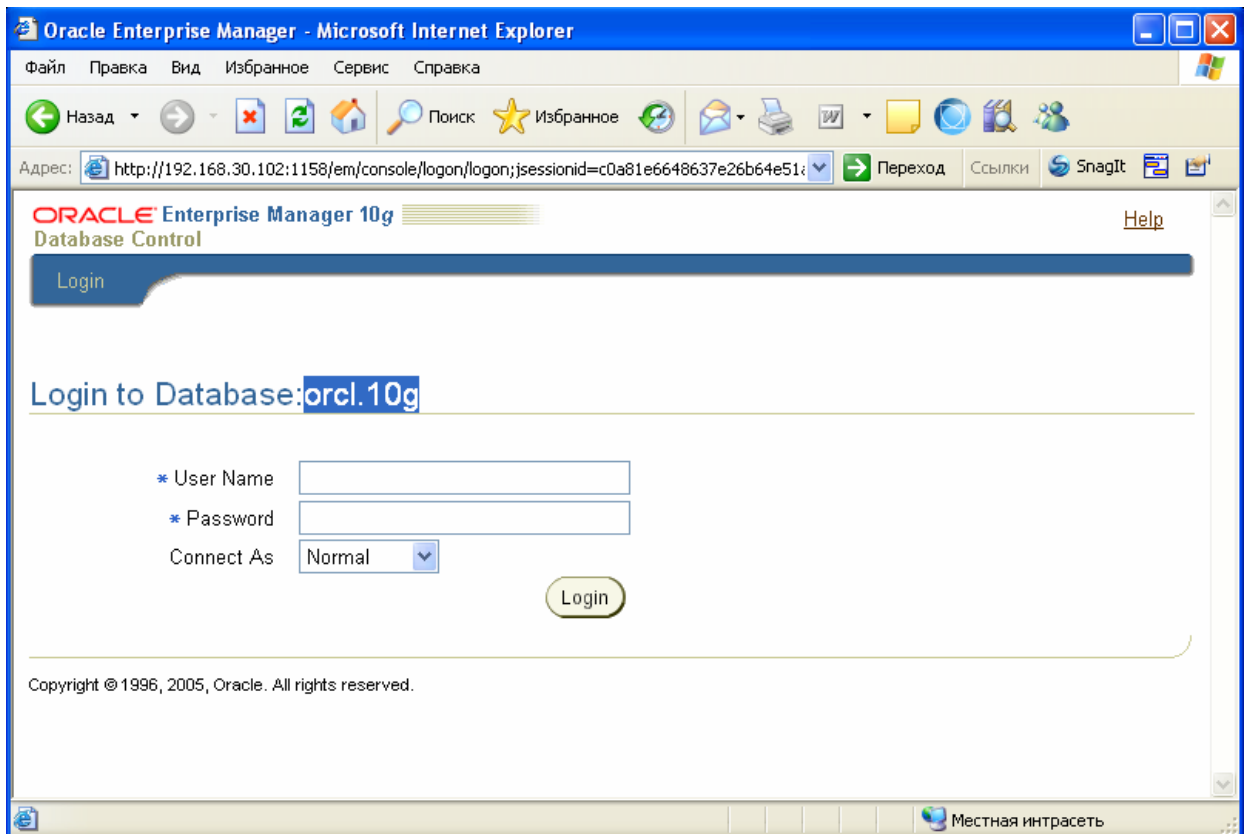
Guessing *SID* using dictionary or brutefore is not always successful. Also bruteforcing is very noisy. It generates many traffic and alerts in listener log and can be simply detected. That's why I decided to find another ways to get database *SID* that will be explained in this chapter.

In many big companies Oracle database are used with different applications for example *Oracle Application Server* or *Oracle SOA Suite* and also with third-party products such as *SAP R/3*. If we have access to applications that integrated with Oracle database we can try to find *SID* or *SERVICE_NAME* even if remote listener commands are denied and bruteforcing is not successful. Here are the most popular applications that can be used to find database *SID* without any additional privileges:

- Oracle Enterprise Manager Control;
- Oracle Application Server;
- Oracle XDB;
- SAP Web Application Server;
- Vulnerable Web-applications.

Oracle Enterprise Manager Control

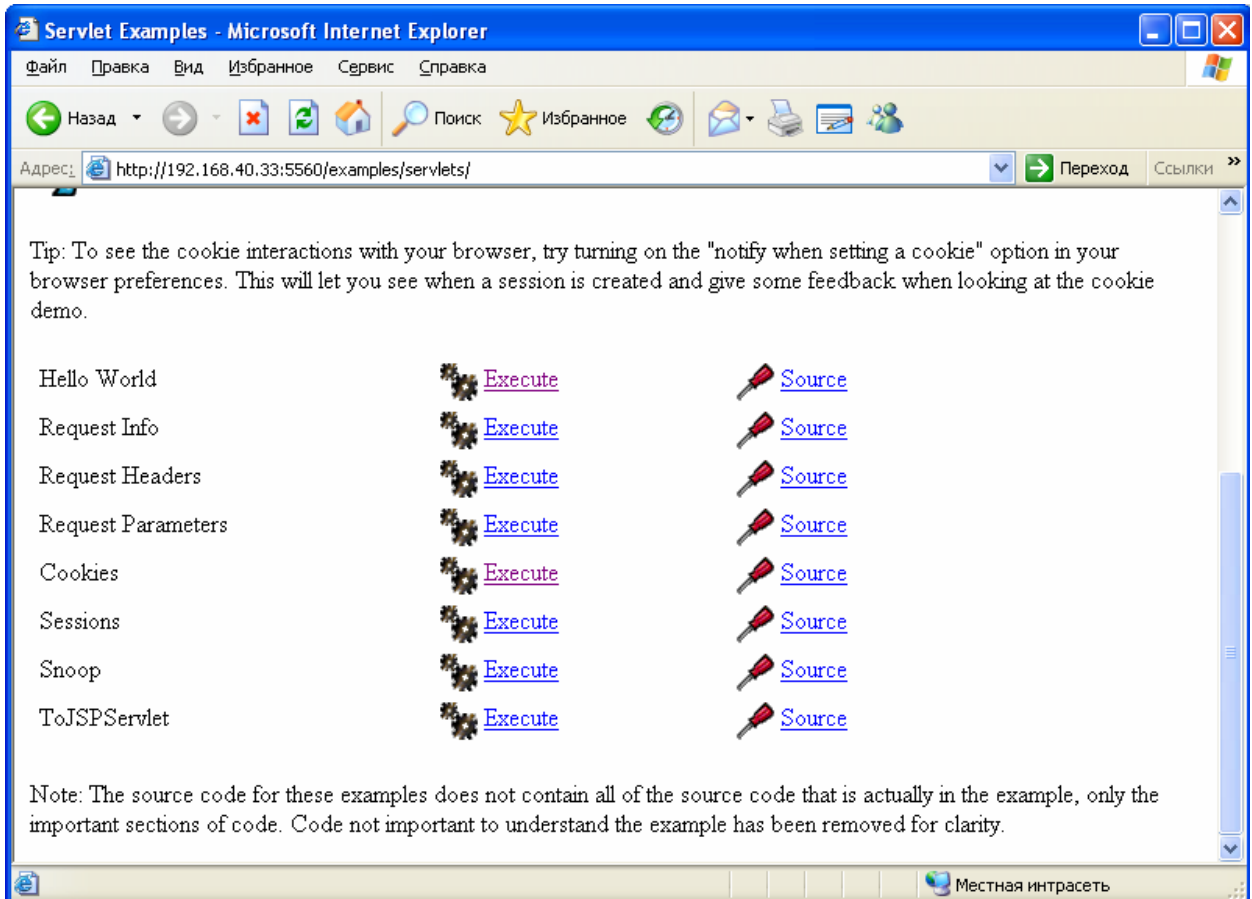
This is one of the most popular ways to get `SERVICE_NAME` through *Enterprise Manager Control* web interface. When installing Oracle database 10g R2 there is *Enterprise Manager Control* installed by default and listening port `1158/tcp`. If we connect using a browser to `http://hostname:1158/em/console` we will see a welcome page with login and password form which also contains a database `SERVICE_NAME` value.



Finding a SERVICE_NAME using Enterprise Manager Database Control

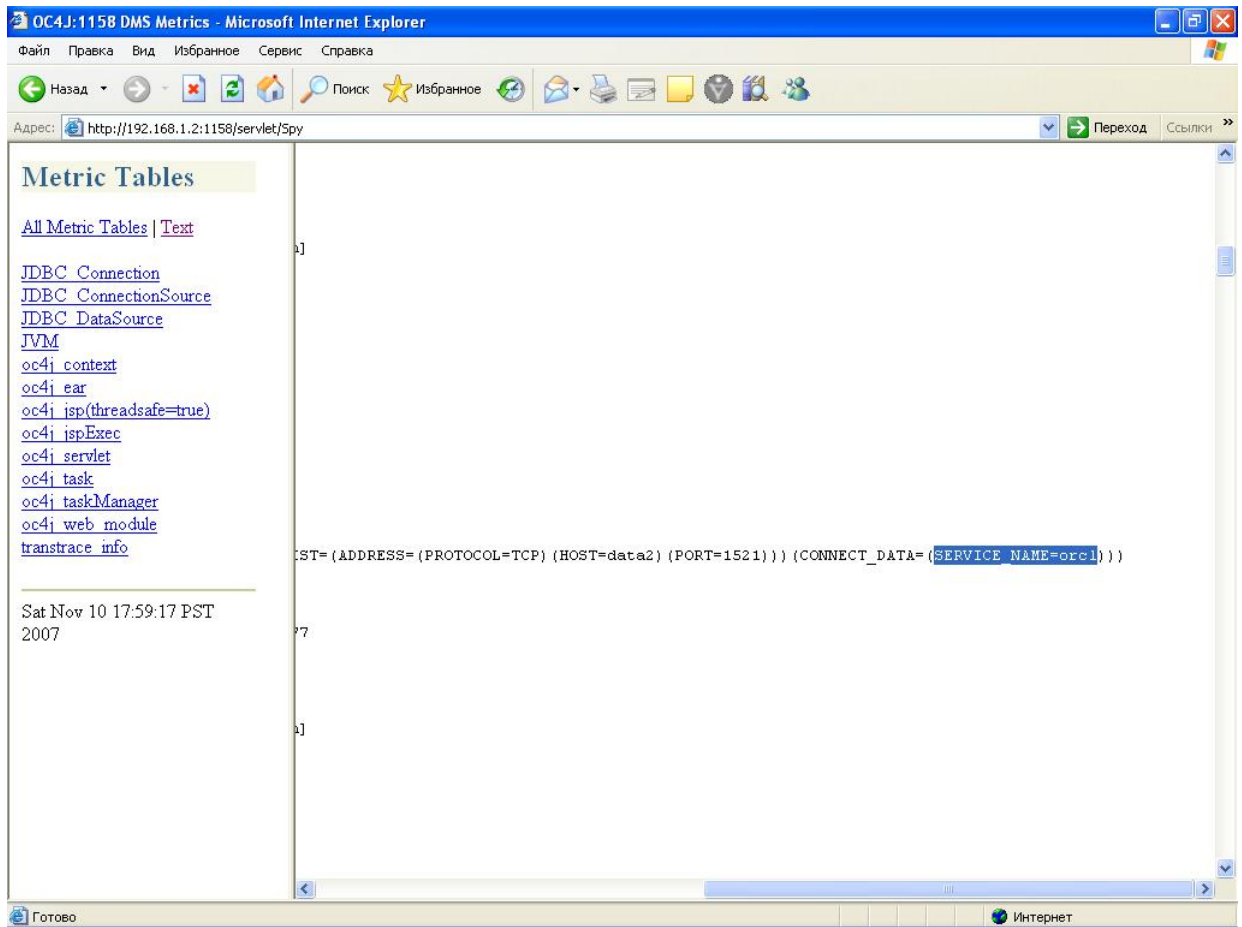
Oracle Application Server

When installing Oracle database 10g there is one component which is installed by default, it is *Oracle Application Server Containers for J2EE*. There is some default servlets that installed with *Oracle Application Server Containers*. We can see this servlets using this link: <http://hostname:5560/exmples/servlets>.



List of default Servlets

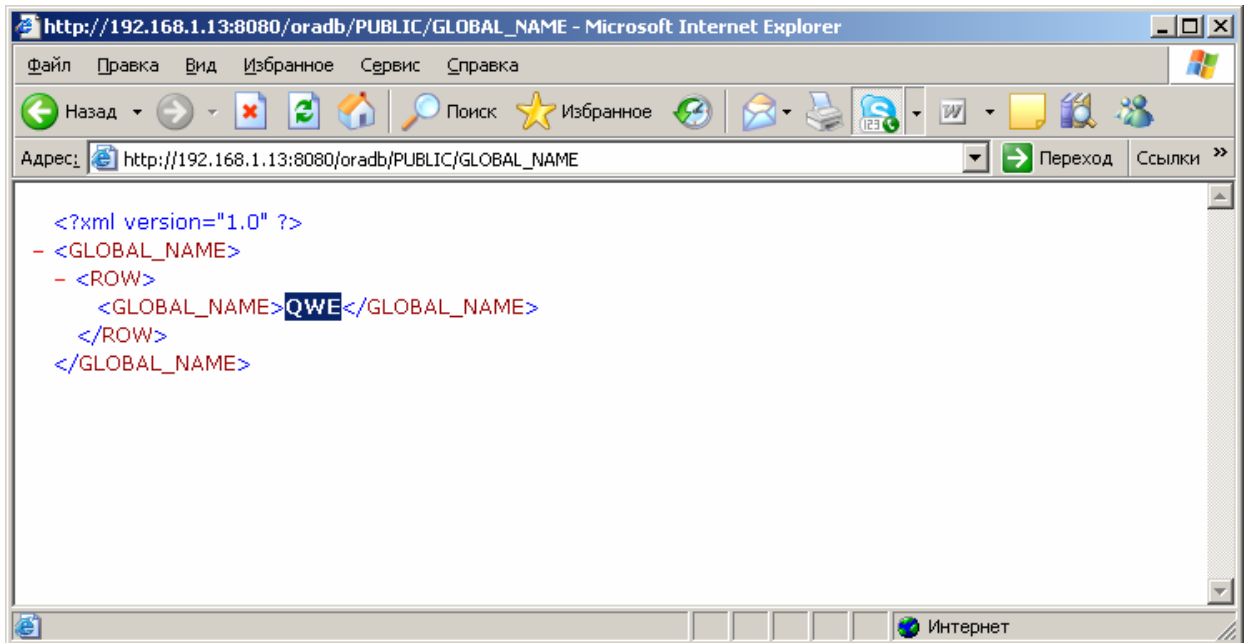
There is also some servlets that is not shown in this page. One of this servlets is named "Spy". We can use this servlet to get a database *SERVICE_NAME*. To do this we must follow this link <http://hostname:5560/servlets/Spy>.



Getting *SERVICE_NAME* using "Spy" servlet

Oracle XDB

If we have username and password to connect to database but we don't know *SID* we can try to connect *Oracle XML DB Enterprise Edition httpd* (this component is installed by default in Oracle database version 9 and 10). To get database *SERVICE_NAME* we must follow this link: http://hostname:8080/oradb/PUBLIC/GLOBAL_NAME.



Getting *SERVICE_NAME* using Oracle XML DB

After getting *SERVICE_NAME* we can connect to database using client utility such as *sqlplus*.

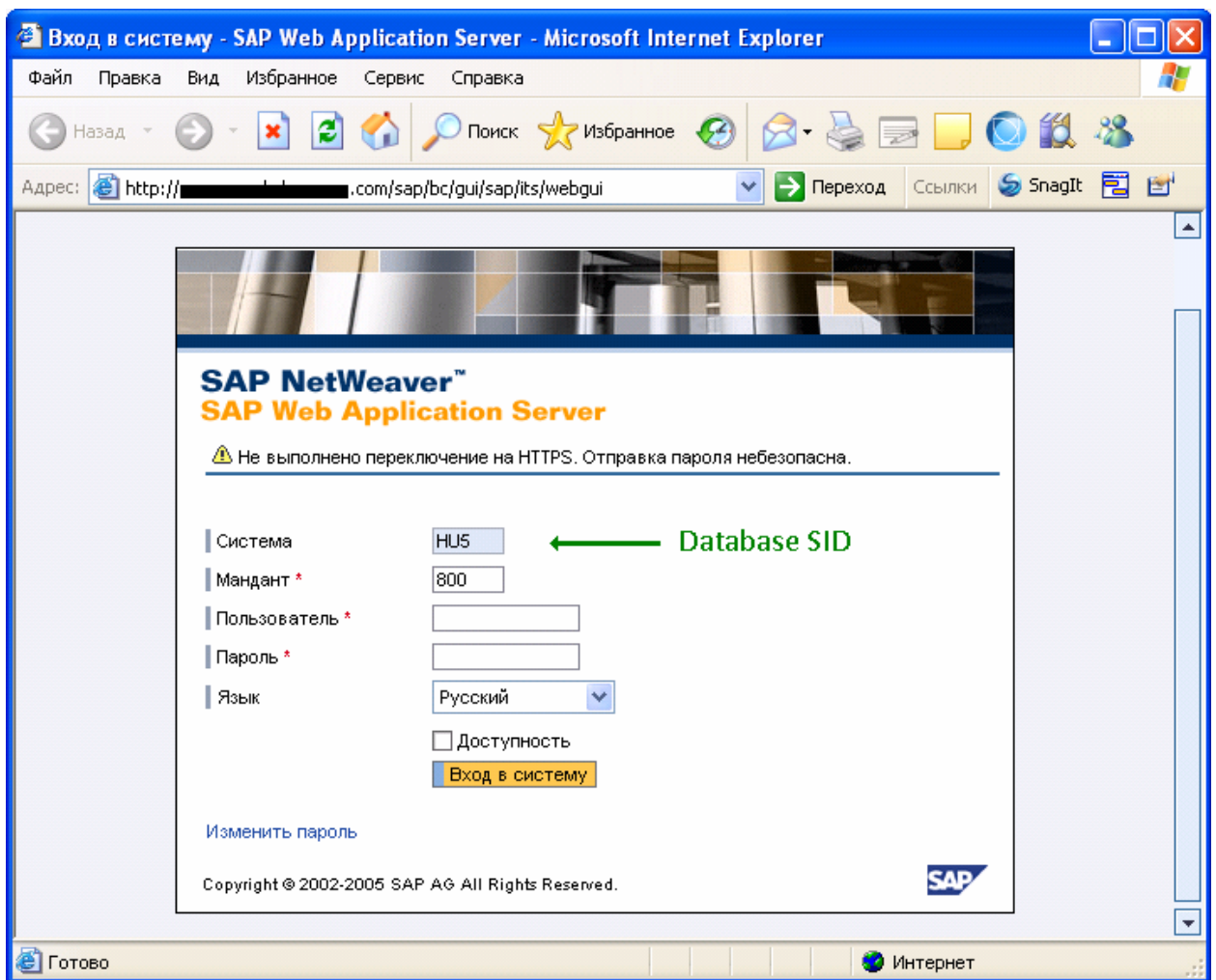
SAP

People are often use Oracle as a backend for SAP/R3. During our research in SAP systems we find that if Oracle database is using as a backend for SAP systems there are at least 4 ways to find Oracle *SID* which was successfully tested during our security audits.

There are two ways to find database *SID* using SAP Web Application Server which is listening by default port 8000/tcp.

SAP Web Application Server Default administration page

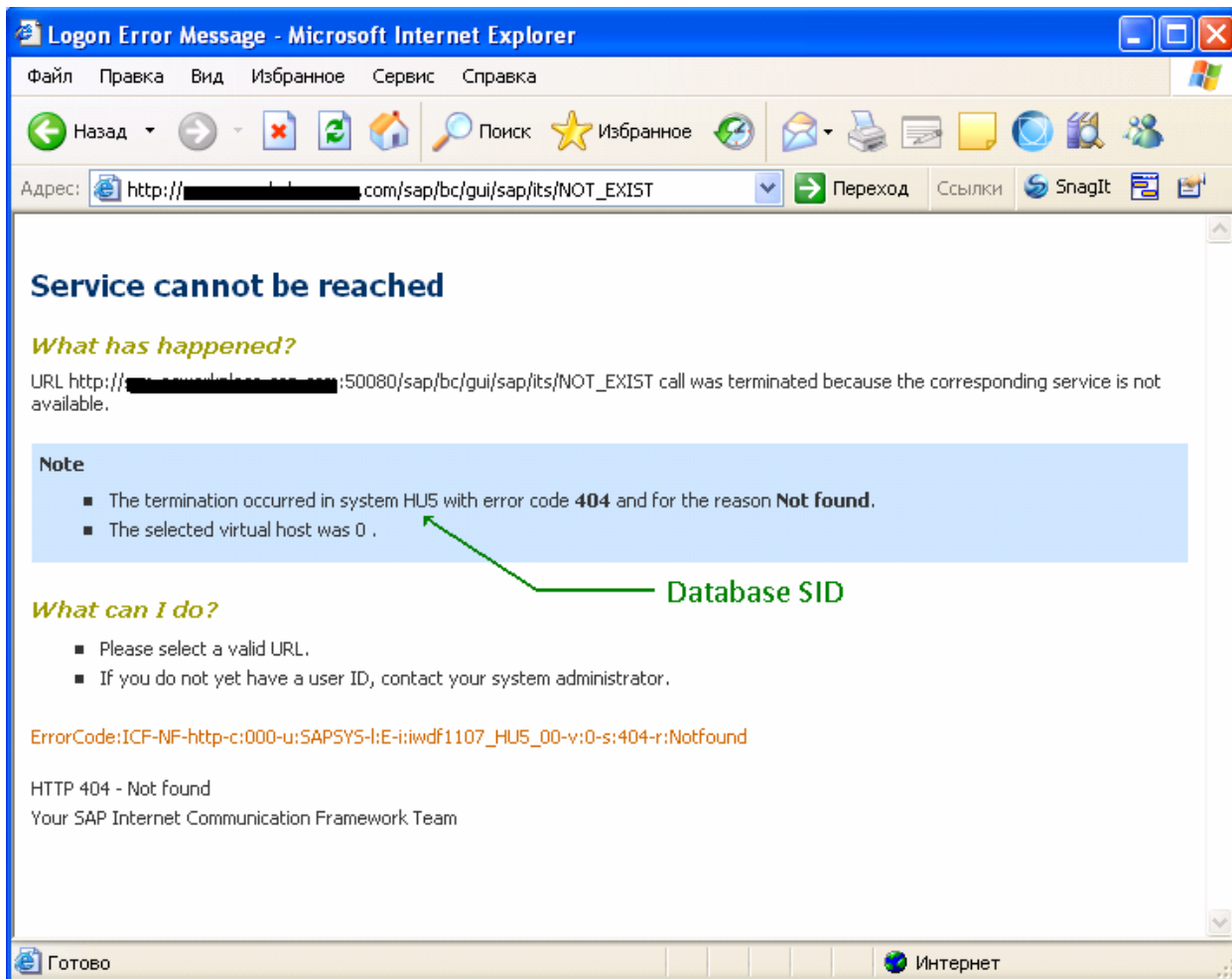
There are two ways to find database *SID* using SAP Web Application Server which is listening by default port 8000/tcp. To get database *SID* we can simply access to SAP web application interface on page `http://hostname:8000/sap/bc/gui/sap/its/webgui`. When we open this page using browser we will see a welcome page with login and password form which is also contains a database *SID* value.



Getting SID using SAP Web Application Server

SAP Web Application Server non-existent page

Alternative way to get database *SID* is to request non-existent page from *SAP Web Application Server*. Server will reply us a 404 page which is contains many debug information and also a database *SID*.



Getting SID using SAP Web Application Server 404 error

SAP RFC

Another way to get Oracle Database *SID* and other interesting information is rfcping utility which is using for testing SAP RFC interface. If RFC Interface is not working this method will not work too.

```
./rfcping ahost=172.16.1.13 sysnr=00
```

SAP System Information

```
-----  
Destination                test2_NSP_00  
  
Host                        test2  
System ID                   NSP  
Database                    NSP  
DB host                     test2  
DB system                   ORACLE  
  
SAP release                 700  
SAP kernel release         700  
  
RFC Protokoll              011  
Characters                  1100 (NON UNICODE PCS=1)  
Integers                   LIT  
Floating P.                IE3  
SAP machine id             560
```

In this example we see that database is Oracle and System ID = NSP.

SAP SID Bruteforcing

When creating database *SID* in SAP system there is one limitation. *SID* must contain only Latin symbols and digits and must be 3 or less symbols length. This means that we can simply get this *SID* by bruteforcing (It will take us about 10 minutes). We can use this method even if first 3 methods are not working (for example access to SAP web application server and RFC interfaces are restricted).

Getting database *SID* using additional rights on target system

Now we know how to get database *SID* using third-party applications without knowing any additional information such as authentication data. If all of described methods failed we can try to get database *SID* using additional rights on target server or on other resource in information system. Suppose we have additional rights on target server or applications installed on it.

Getting *SID* using some rights on target server

There are 3 ways to get database *SID*:

- Getting *SID* using operation system account on server;
- Getting *SID* using FTP account on server;
- Getting *SID* using MSSQL account on server.

Getting *SID* using operating system account on server

This is very simple, if user have access to `$ORACLE_HOME/NETWORK/admin` directory he can get *SID* from database configuration file `tnsnames.ora`. Also we can simply try `“lsnrctl status”` command.

Getting *SID* using FTP account on server

If we have only FTP account on target server we can get database *SID* if ftp user have read access to `$ORACLE_HOME` directory. To get database *SID* we can read configuration file `tnsnames.ora`. If user doesn't havey read access to this file he can get *SID* using directory listing. In different database versions there are different locations of directory that named like *SID*. For example let's take Oracle database 10g R2.

First way is to list directories `ORACLE_HOME\..\admin\` directory:

```
C:\oracle\product\10.2.0\oradata >dir
Том в устройстве С не имеет метки.
Серийный номер тома: 8CFF-37FB

Содержимое папки C:\oracle\product\10.2.0\admin

21.09.2008  12:55    <DIR>          .
21.09.2008  12:55    <DIR>          ..
21.09.2008  12:55    <DIR>          ORCL102
```

Directory name `ORCL102` is a database *SID*.

Second way is to list directories directory: `$ORACLE_HOME\..\oradata\`:

```
C:\oracle\product\10.2.0\oradata>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 8CFF-37FB

Содержимое папки C:\oracle\product\10.2.0\oradata

21.09.2008  12:55    <DIR>          .
21.09.2008  12:55    <DIR>          ..
21.09.2008  12:55    <DIR>          ORCL102
```

Directory name *ORCL102* is a database SID.

Third way is to list directories directory `$ORACLE_HOME`:

```
Содержимое папки E:\oracle\product\10.2.0\db_1

28.01.2008  18:07    <DIR>          .
28.01.2008  18:07    <DIR>          ..
28.01.2008  18:07    <DIR>          192.168.40.14_orcl102
28.01.2008  17:30    <DIR>          admin
28.01.2008  17:30    <DIR>          assistants
19.06.2008  16:53    <DIR>          BIN
```

We see a directory name *192.168.40.14_ORCL102* where *ORCL102* is a database SID.

Getting SID using MsSQL account on server

In our security audits we often see that administrators install different databases on one server (this situation is not only in test servers but in production systems too). A frequently situation when Oracle database and MsSQL database are both installed on one server.

If we have any account on MsSQL server even with *public* rights on *master* table (for example we get this account by remote bruteforcing) we can get Oracle *SID* using MsSQL stored procedures. All methods of getting *SID* are different for different database versions but they are all based on two stored procedures:

- *Master..xp_regread* – reads registry key values.
- *Maste.r.xp_dirtree* – returns server directory tree.

For example in Oracle database 9g R2 *SID* stored in known registry key:

```
EXEC master..xp_regread 'HKEY_LOCAL_MACHINE', 'SOFTWARE\ORACLE\HOME0',
'ORACLE_SID'
GO
```

```
Выбрать SQLCMD
C:\Program Files\Windows Resource Kits\Tools>sqlcmd -S 172.16.1.13 -U test -P test -W
1> USE master
2> SELECT USER
3> GO
Changed database context to 'master'.

-
guest

<1 rows affected>
1> EXEC master..sp_helpuser 'guest'
2> GO
UserName  GroupName  LoginName  DefDBName  UserID  SID
-----
guest  public  NULL  NULL  2  0x00

1> EXEC master..xp_regread 'HKEY_LOCAL_MACHINE', 'SOFTWARE\ORACLE\HOME0', 'ORACLE_SID'
2> go
Value Data
-----
ORACLE_SID orcl9
1> _
```

Getting SID from registry using MsSQL stored procedures

To get database SID in later versions of Oracle database we must use another ways.

Getting SID using list of services

Main database service use a *SID* value in his title. To get a services list we can execute this command:

```
EXEC master..xp_regread 'HKEY_LOCAL_MACHINE',
'SYSTEM\CurrentControlSet\Services\Eventlog\Application', 'Sources'
GO
```

```

C:\> Выбрать SQLCMD
1>
2> EXEC master..xp_regread 'HKEY_LOCAL_MACHINE', 'SYSTEM\CurrentControlSet\Services\Eventlog\Application', 'Sources'
3> go
Value Value Data
-----
Sources - Item #1 WSH NULL
Sources - Item #2 WMIAdapter NULL
Sources - Item #3 WndmPmSN NULL
Sources - Item #4 WinMgmt NULL
Sources - Item #5 Winlogon NULL
Sources - Item #6 Windows Product Activation NULL
Sources - Item #7 Windows 3.1 Migration NULL
Sources - Item #8 WebClient NULL
Sources - Item #9 USS NULL
Sources - Item #10 vntools NULL
Sources - Item #11 UBRuntime NULL
Sources - Item #12 Userinit NULL
Sources - Item #13 Userenv NULL
Sources - Item #14 UploadM NULL
Sources - Item #15 TrustMonitor NULL
Sources - Item #16 Tlntsvr NULL
Sources - Item #17 SysmonLog NULL
Sources - Item #18 SQLSERUERAGENT NULL
Sources - Item #19 SQLFTHNDLR NULL
Sources - Item #20 SQLCTR NULL
Sources - Item #21 SpoolerCtrs NULL
Sources - Item #22 Software Restriction Policies NULL
Sources - Item #23 Software Installation NULL
Sources - Item #24 SclgNtfy NULL
Sources - Item #25 SceSrv NULL
Sources - Item #26 SceCli NULL
Sources - Item #27 safrsrv NULL
Sources - Item #28 SAFrdms NULL
Sources - Item #29 Remote Assistance NULL
Sources - Item #30 PerfProc NULL
Sources - Item #31 PerfOS NULL
Sources - Item #32 PerfNet NULL
Sources - Item #33 Perfmon NULL
Sources - Item #34 Perflib NULL
Sources - Item #35 PerfDisk NULL
Sources - Item #36 Perfctrs NULL
Sources - Item #37 PassportManager NULL
Sources - Item #38 OracleOraDb10g_home1$SQL*Plus NULL
Sources - Item #39 OracleDBConsoleorcl NULL
Sources - Item #40 Oracleorcl NULL
Sources - Item #41 Offline Files NULL

```

Getting a database SID from services list using Mssql stored procedures

Number 40 is a main Oracle database service named *Oracle.orcl* and "ORCL" is database SID. This method is working on Oracle versions 10g R1, 10g R2 and 11g R1.

Getting database SID using registry key HKLM\SOFTWARE\ORACLE

When installing Oracle database by default in registry created a folder with name depend on database version. For example in Oracle database 10g the name of the folder is *KEY_OraDb10g_home1*, and in version Oracle database 11g is *KEY_OraDb11g_home1*. To get Oracle SID we must read key *ORACLE_SID* stored in this folder:

```

EXEC master..xp_regread 'HKEY_LOCAL_MACHINE',
'SOFTWARE\ORACLE\KEY_OraDb10g_home1', 'ORACLE_SID'
GO

```

```
C:\> Выбрать SQLCMD
1> EXEC master..xp_regread 'HKEY_LOCAL_MACHINE', 'SOFTWARE\ORACLE\KEY_OraDb10g_h
ome1', 'ORACLE_SID'
2> GO
Value Data
-----
ORACLE_SID orcl
1>
2>
```

Getting SID from registry key

Getting SID using directory listing

If described methods are not succeeded we can try to get $\$ORACLE_HOME$ folder and then read a directory list of this folder where we can find a *SID*. (Method described in chapter “Getting SID using FTP account on server”). For example in Oracle 11g we can get $\$ORACLE_HOME$ value by this command:

```
EXEC master..xp_regread 'HKEY_LOCAL_MACHINE',
'SOFTWARE\ORACLE\ODP.NET\1.111.6.0', 'DllPath'
GO
```

In version 10g we can test default $\$ORACLE_HOME$ values such as:

```
C:\oracle\product\10.2.0\
C:\oracle\product\10.1.0\
```

Example of getting SID using directory listing in Oracle 10g r1

```
EXEC master..xp_dirtree '$ORACLE_HOME'
GO
C:\Program Files\Windows Resource Kits\Tools>sqlcmd -S 192.168.30.102 -U test
-P
test -W
1> EXEC master..xp_dirtree 'C:\oracle\product\10.1.0\oradata\'
2> go
subdirectory depth
-----
1> EXEC master..xp_dirtree 'D:\oracle\product\10.1.0\oradata\'
2> go
subdirectory depth
-----
1> EXEC master..xp_dirtree 'D:\oracle\product\10.2.0\oradata\'
2> go
subdirectory depth
-----
1> EXEC master..xp_dirtree 'C:\oracle\product\10.2.0\oradata\'
2> go
subdirectory depth
-----
orcl 1
```

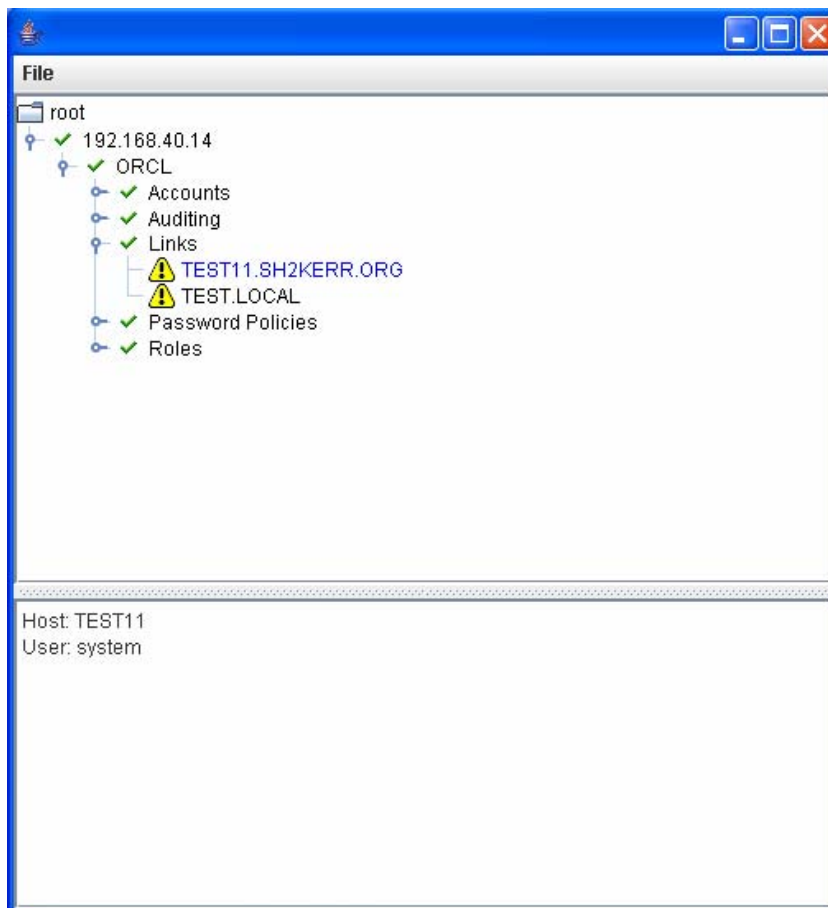
Finally we find that $\$ORACLE_HOME = 'C:\oracle\product\10.2.0'$ and database $SID = "orcl"$.

Getting SID using additional rights in Company's network

In real life information system consists from many different database servers connected with each other. Some of the servers are more or less secured then others. If we can get access to less secured servers in some situations it will help us to get access to more secured servers. Let's see how we can get database SID using data from another database servers or sniffing the network.

Getting SID from another database's

If we have access to some Oracle databases in target information system we can try to get *SID* from database links. To find all database links we can use *Oscanner* utility. Also in database links we can find not only database *SID* but username and password for connection.



Getting SID using database links

In our example we see two links to databases *SID=TEST11* and *SID=TEST*.

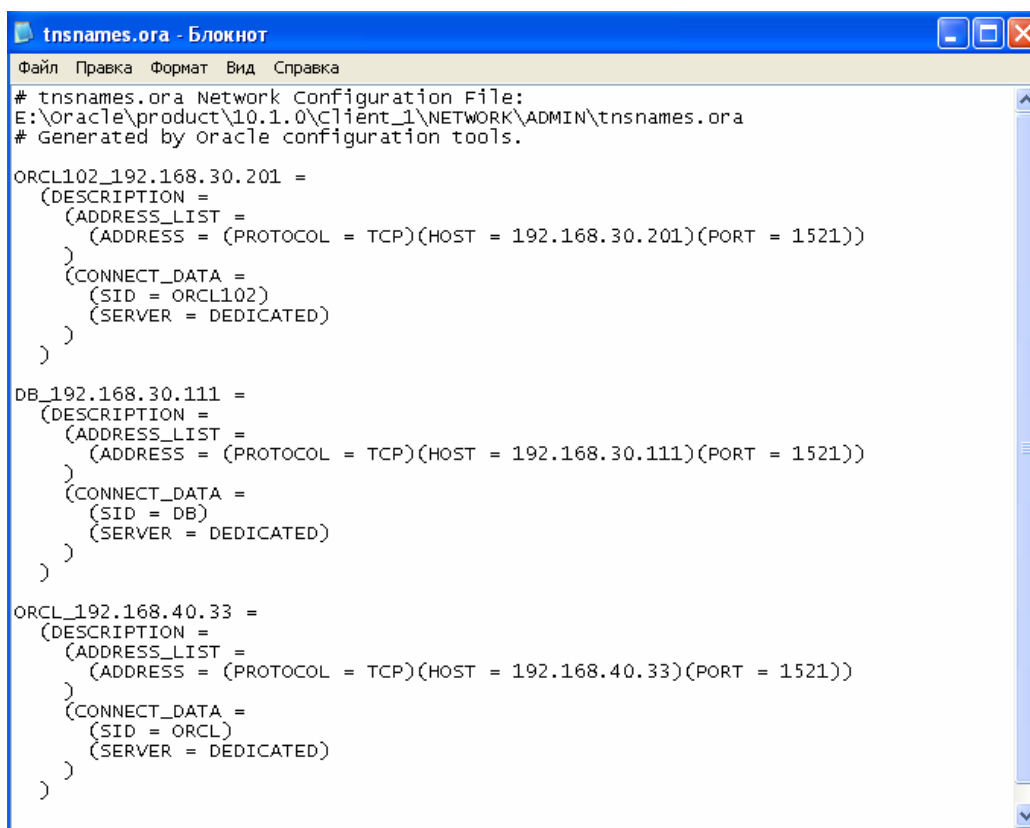
In our statistics of penetration testing about 20 percent of databases use public database links

Getting SID from another servers in target information system

If we somehow get OS access to one of the database servers in our information system we can try to find configuration files with links to another databases.

Usually database links are stored in configuration file `$ORACLE_HOME/NETWORK/admin/tnsnames.ora`. Also we can try to find old copies of the `tnsnames.ora` file. In UNIX-like OS we can find old configuration files using this command:

```
find / -name tnsnames*
```



```
tnsnames.ora - Блокнот
Файл  Правка  Формат  Вид  Справка
# tnsnames.ora Network Configuration File:
E:\Oracle\product\10.1.0\client_1\NETWORK\ADMIN\tnsnames.ora
# Generated by oracle configuration tools.

ORCL102_192.168.30.201 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.30.201)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ORCL102)
      (SERVER = DEDICATED)
    )
  )

DB_192.168.30.111 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.30.111)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = DB)
      (SERVER = DEDICATED)
    )
  )

ORCL_192.168.40.33 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.40.33)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ORCL)
      (SERVER = DEDICATED)
    )
  )
```

Example of tnsnames.ora file with SID's

In our example of *tnsnames.ora* we see information about 3 servers with their IP addresses and *SID*'s.

In our statistics of penetration testing about 60 percent of tnsnames.ora store links to another databases

Sniffing database SID from network

If we can sniff traffic in our network between database users and database server we can get *SID* while it is transmitting through the network. To sniff transmitting data we can use any network analyzer such as Wireshark (Ethereal).

As we can see in screenshot that client with IP-address *192.168.40.14* trying to connect to database server with IP-address *192.168.40.33* and transmit a database *SERVICE_NAME* (we also can sniff a *SID* by the same way).

(Untitled) - Ethereal

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
323	56.721220	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [ACK] Seq=1 Ack=1 win=65535 Len=0
324	56.721348	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [PSH, ACK] Seq=1 Ack=1 win=65535 Len=262
325	56.825493	192.168.40.33	192.168.40.14	TCP	4229 > 10014 [PSH, ACK] Seq=1 Ack=263 win=65273 Len=32
326	56.861258	192.168.40.14	192.168.40.1	TCP	1041 > microsoft-ds [ACK] Seq=13301 Ack=35715 win=64082
327	56.863752	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [PSH, ACK] Seq=263 Ack=33 win=65503 Len=156
328	56.864109	192.168.40.33	192.168.40.14	TCP	4229 > 10014 [PSH, ACK] Seq=33 Ack=419 win=65117 Len=127
329	56.930293	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [PSH, ACK] Seq=419 Ack=160 win=65376 Len=37
330	56.930615	192.168.40.33	192.168.40.14	TCP	4229 > 10014 [PSH, ACK] Seq=160 Ack=456 win=65080 Len=17
331	57.032326	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [PSH, ACK] Seq=456 Ack=332 win=65204 Len=62
332	57.032741	192.168.40.33	192.168.40.14	TCP	4229 > 10014 [PSH, ACK] Seq=332 Ack=518 win=65018 Len=22
333	57.063608	192.168.40.14	192.168.40.33	TCP	10014 > 4229 [PSH, ACK] Seq=518 Ack=354 win=65182 Len=22

```

0000 00 50 8d d1 a4 17 00 04 61 6f f6 89 08 00 45 00 .P.....ao....E.
0010 01 2e 3d e4 40 00 80 06 ea 65 c0 a8 28 0e c0 a8 ..=.@... .e.(...
0020 28 21 27 1e 10 85 49 a4 0c a9 7e c6 79 b8 50 18 (!'...I. ...y.P.
0030 ff ff 6b 70 00 00 01 06 00 00 01 04 00 00 01 39 ..kp.... ....9
0040 01 2c 00 00 08 00 7f ff 86 0e 00 00 01 00 00 cc .....AA .....
0050 00 3a 00 00 02 00 41 41 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0070 28 44 45 53 43 52 49 50 54 49 4f 4e 3d 28 43 4f (DESCRIP TION=(CO
0080 4e 4e 45 43 54 5f 44 41 54 41 3d 28 53 45 52 56 NNECT_DA TA=(SERV
0090 49 43 45 5f 4e 41 4d 45 3d 4f 52 43 4c 29 28 43 ICE_NAME =ORCL)((
00a0 49 44 3d 28 50 52 4f 47 52 41 4d 3d 45 3a 5c 4f ID=(PROG NAME:\O
00b0 72 61 63 6c 65 5c 70 72 6f 64 75 63 74 5c 31 30 racle\pr oduct\lo
00c0 2e 31 2e 30 5c 43 6c 69 65 6e 74 5f 31 5c 62 69 .1.0\cl1 ent_1\b1
00d0 6e 5c 73 71 6c 70 6c 75 73 2e 65 78 65 29 28 48 n\sqlplu s.exe)(H
00e0 4f 53 54 3d 57 53 30 31 34 29 28 55 53 45 52 3d OST=WS014)(USER=
00f0 41 6c 65 78 61 6e 64 72 2e 50 6f 6c 79 61 6b 6f Alexandr .Polyako
0100 76 29 29 29 28 41 44 44 52 45 53 53 3d 28 50 52 v)))(ADD RES=(PR
0110 4f 54 4f 43 4f 4c 3d 54 43 50 29 28 48 4f 53 54 OTOCOL=T CP)(HOST
0120 3d 31 39 32 2e 31 36 38 2e 34 30 2e 33 33 29 28 =192.168 .40.33)(
0130 50 4f 52 54 3d 31 35 32 31 29 29 29 PORT=152 1))
  
```

Database SID ←

File: "C:\DOCUME~1\ALEXAN~1\A [P: 410 D: 410 M: 0 Drops: 0

Sniffing SERVICE_NAME

Conclusion

In this document I collected all popular and specific methods of getting a database *SID* from simple bruteforce to new methods of *SID* in third-party applications. As I said before, this is very important step in process of getting access to database. Now when we know how to get a database *SID*, we can try to bruteforce database accounts, escalate privileges using PL/SQL injections, get access to OS and so on.

Links

1. List of default SID's

http://www.red-database-security.com/whitepaper/oracle_default_sid.html

2. Backtrack Oracle Tutorial

http://www.red-database-security.com/wp/backtrack_oracle_tutorial.pdf

3. Pentesting / Hacking Oracle databases with

<http://www.red-database-security.com/wp/itu2007.pdf>

4. Utilities to bruteforce SID

<http://www.red-database-security.com/software/sidguess.zip>

http://www.cqure.net/tools/osscanner_bin_1_0_6.zip

<http://www.vulnerabilityassessment.co.uk/oak.htm>

http://www.cqure.net/tools/SIDGuesser_win32_1_0_5.zip

<http://inguma.sourceforge.net/index.php>