Oracle Database Listener Security Guide

January 2004

INTEGRIGY

Mission Critical Applications…
…Mission Critical Security

Oracle Database Listener Security Guide
October 2002
March 2003 – Updated
January 2004 – Updated

If you have any questions, comments, or suggestions regarding this document, please send them via e-mail to alerts@integrigy.com.

About Integrigy Corporation (www.integrigy.com)

Integrigy Corporation is a leader in application security for large enterprise, mission critical applications. Our application vulnerability assessment tool, AppSentry, assists companies in securing their largest and most important applications. Integrigy Consulting offers security assessment services for leading ERP and CRM applications.

Integrigy Corporation
2052 Lincoln Park West, Suite 1301
Chicago, Illinois 60614 USA
888/542-4802
www.integrigy.com

Table of Contents

1

Introduction

1.1 Introduction

Through our security consultations, we have consistently identified poor Oracle Database Listener (TNS Listener) security as a significant security risk. Few Oracle database installations have properly secured the Oracle listener as recommended by Oracle and security experts.

The information contained in this paper is not new, is not obscure. It may not be well known to many Oracle DBAs, but is well known to security experts and hackers.

The paper will outline the vulnerabilities in the Oracle TNS Listener and provide recommendations for properly securing it. Providing minimal security for the Listener is simple and should be done for all Oracle installations – development, test and production.

The information is this document is based on Oracle8i and Oracle9i. The information should be valid for 7.3.x and 8.0.x, but has not been tested.

2

Listener Overview

For detailed information on the Oracle TNS Listener and Oracle networking, see the "Oracle 9i Net Services Administrator's Guide."

The Oracle TNS Listener is the server-based process that provides basic network connectivity for clients, application servers, and other databases to an Oracle database. In addition to databases, the Listener can also be configured to support binary executables.

2.1 Listener Details

The relevant files for the Oracle TNS Listener are as follows –

$ORACLE_HOME/bin/lsnrctl	Listener control program
$ORACLE_HOME/network/admin/listener.ora	Configuration file for the listener
$ORACLE_HOME/bin/tnslnsr	Server listener process

The lsnrctl program is the mechanism for starting and stopping the listener process (tnslsnr). Tnslnsr starts and reads the listener.ora file for configuration information, such as port number and database sid.

The tnslnsr processes starts with the process owner of the lsnrctl program, usually the "oracle" account on UNIX or "administrator" on Windows NT/2000. Any successful exploit will thus gain the privileges for this account.

2.2 Listener Modes

The Listener can be configured in one of three modes (as configured in listener.ora) –

Database	Provides network access to an Oracle database instance
PLSExtProc	Method for PL/SQL packages to access operating system executables
Executable	Provides network access to operating system executables

The "Database" mode is the most widely used mode and is the standard mode used by every database for connectivity. "PLSExtProc" allows PL/SQL database packages to access external

programs and is configured by default for many instances. "Executable" mode allows an external program to be defined and accessed through a TNS connection. There is little documentation on this mode and is almost exclusively used by Oracle Applications.

2.3 Listener Remote Management

During our security evaluations, we are always amazed to discover that some DBAs are not aware that the Listener can be remotely managed using "lsnrctl" or a similar program from a remote machine.

To set up a computer to remotely administer a Listener:

1. Install Oracle software that contains SQL*Net or Net8. You may have to copy the lsnrctl (or lsnrctl.exe) executable from a server as it is not installed as part of the basic Oracle client software.

2. Configure the local listener.ora to resolve to the remote Listener.

   The syntax is identical to a tnsnames.ora entry. If the listener.ora does not exist, create and insert the entry. Otherwise, add the entry to the listener.ora file.

```
<alias> =
  (ADDRESS_LIST =
      (ADDRESS =
            (PROTOCOL = TCP)
            (Host = <host>)
            (Port = <port>)
      )
  )
```

3. Start lsnrctl and specify the Listener name –

```
lsnrctl <alias>
```

   All lsnrctl commands are valid, except for **start**.

3

Listener Exploits

3.1 Listener Remote Management

If a password is not set on the Listener, someone who knows just a hostname and port number (default port is 1521) has full control over the Listener. They can do the following –

- Stop the Listener
- Set a password and prevent others from controlling the Listener
- Write trace and log files to any file accessible to the process owner of tnslnsr (usually Oracle)
- Obtain detailed information on the Listener, database, and application configuration

Command	Password Required if Set	Risk	Comments
change_password	x	medium	Change password and prevent DBA from controlling the Listener – listener.ora can be manually edited to remove the password
dbsnmp_start	local only	-	
dbsnmp_status	local only	-	
dbsnmp_stop	local only	-	
help		-	
reload	x	-	Activate some changes made using set commands
save_config	x	low	Make Listener changes permanent
services	x	leakage	Detailed information on Listeners including full paths and environmental variables It is possible to bypass the password
set connect_timeout		-	
set display_mode		-	
set log_directory set log_file set log_status	x x x	high	Create a log file in an arbitrary location – 1. log file could be a externally readable location – such as a web server directory 2. overwrite any file accessible to the tnslnsr process owner (e.g., .htaccess, .rhost, .profile)

set save_config_on_stop		-	
set statup_waittime	x	low	Set to arbitrary high number so the Listener never starts
set trc_directory set trc_file set trc_level trace	x	medium	Create a log file in an arbitrary location – 1. log file could be a externally readable location – such as a web server directory 2. overwrite any file accessible to the tnslnsr process owner (e.g., .htaccess, .rhost, .profile)
set use_plugandplay		-	
show	x*	leakage	* for show save_config_on_stop and show use_plugandplay
spawn	x	medium	Very specific to the implementation
start	local only	-	
status		leakage	Detailed information regarding operating system, databases, and services
stop	x	high	Stop the listener
version		leakage	Displays versions for TNS and protocol stack – version numbers are usually the same as the database (e.g., 8.1.7.1.0)

Table 3.1 – Listener Commands

3.2 Listener Information Leakage

The Listener provides information regarding the enabled services, which includes database SIDs and external programs. If someone knows the hostname and port number (default port is 1521), they can determine the databases running and any other configured services for the Listener regardless if a password is set or not.

From a single command (TNS "status" command) even if a password is set, someone can determine all the services running for the listener (all databases), database version and minor patch levels, if a password is set, if tracing is enabled, $ORACLE_HOME path, and if SNMP is enabled.

The services command does require a password if it is enabled, however, it is possible to bypass the password. The status and version commands do not.

Command	Password Required if Set	Information/Sample Output

services	x	`Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROCdev1))`
		`Services Summary...`
		`Service "PLSExtProc"  has 1 instances.`
		`  Instance "PLSExtProc"`
		`  Status: READY  Total handlers: 1  Relevant handlers: 1`
		`  DEDICATED established:0 refused:0 current:0 max:0 state:ready`
		`Service "dev1"  has 1 instances.`
		`  Instance "dev1"`
		`  Status: READY  Total handlers: 2  Relevant handlers: 2`
		`  DEDICATED established:5 refused:0 current:0 max:0 state:ready`
		`  DEDICATED established:0 refused:0 current:10 max:87 state:ready`
		`  Session: NS`
		
		(additional information if "displaymode = verbose", including environmental variables and paths)

status		`STATUS of the LISTENER`
		`------------------------`
		`Alias  dev1`
		`Version  TNSLSNR for Solaris: Version 8.1.7.1.0 - Production`
		`Start Date  20-AUG-2002 22:16:09`
		`Uptime  0 days 13 hr. 20 min. 26 sec`
		`Trace Level  off`
		`Security  OFF`
		`SNMP  OFF`
		`Listener Parameter File  /u01/dev1db/8.1.7/network/admin/listener.ora`
		`Listener Log File  /u01/dev1db/8.1.7/network/admin/dev1.log`
		`Listening Endpoints Summary...`
		`  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROCdev1)))`
		`  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sun)(PORT=1521)))`
		`Services Summary...`
		`Service "PLSExtProc"  has 1 instances.`
		`  Instance "PLSExtProc"`
		`  Status: READY  Total handlers: 1  Relevant handlers: 1`
		`Service "dev1"  has 1 instances.`
		`  Instance "dev1"`
		`  Status: READY  Total handlers: 2  Relevant handlers: 2`

version		`TNSLSNR for Solaris: Version 8.1.7.1.0 - Production`
		`TNS for Solaris: Version 8.1.7.1.0 - Production`
		`Unix Domain Socket IPC NT Protocol Adaptor for`
		`  Solaris: Version 8.1.7.1.0 - Production`
		`Oracle Bequeath NT Protocol Adapter for Solaris:`
		`  Version 8.1.7.1.0 - Production`
		`TCP/IP NT Protocol Adapter for Solaris: Version`
		`  8.1.7.1.0 - Production,,`

Table 3.2 – Information Leakage Commands

3.3 Known Exploits – Oracle Security Alerts

This is a list of most of the known and released exploits for the Oracle TNS Listener. The number refers to the Oracle Security Alert. Patches or Metalink Note IDs are referenced for each exploit.

#54 – Buffer Overflow Vulnerability (4/25/03)

Buffer Overflow and Denial of Service (DoS) vulnerability in all releases – Multiple Patches

#42 – Denial of Service Vulnerability (10/4/02)

Denial of Service (DoS) vulnerability in all releases – Patch 2540219

#40 – Format String Vulnerabilities (8/8/02)

Format string vulnerabilities in all releases – Patch 2395416

#38 – Denial of Service Vulnerability (8/8/02)

Denial of Service (DoS) vulnerability when an invalid command request is sent to the listener causing it to crash in Oracle 9i only – Patch 2467947

#34 – DoS on Windows and VMS with Small Data Sizes (6/6/02)

Excessive CPU utilization on Windows and VMS servers when a small amount of data is sent – Metalink Note 198544.1

#17 – Malformed Packet DoS (7/02/01)

Listener can be corrupted or core dumps with large amounts of connect data – Metalink Note 151260.1

#16 – Buffer Overflow in Oracle 8i Listener (7/02/01)

A buffer overflow occurs when large amounts of command data are sent – Metalink Note 151259.1

#15.4 – Oracle Net8 Fragmentation Attack (7/02/01)

DoS attack when fragmented TNS packets are not sent properly – Metalink Note 151292.1

#15.3 – Maximum Transport Data Size Too Small (7/02/01)

Listener crashes when the Maximum Transport Data Size set to 0 on Sun Solaris only – Metalink Note 151291.1

#15.2 – Requester_version Value Incorrect (7/02/01)

Listener crashes when an invalid value is sent in the client version field on Unix server – Metalink Note 151290.1

#15.1 – Offset_to_data Value Too Large (7/02/01)

Listener crashes when the offset_to_data field contains an arbitrary large value – Metalink Note 151261.1

#14 – Oracle Redirect Denial of Service Vulnerability (7/27/01)

Vulnerability in redirected Net8 connections can consume all the CPU on Windows NT systems running Oracle 7.3.4, Oracle 8, and Oracle 8i – Metalink Note 153289.1

#2 – LOG_FILE and TRC_FILE Spoofing (11/16/00)

LOG_FILE and TRC_FILE can be spoofed and any extension used – Metalink Note 124742.1

3.4 Other Exploits

Brute Forcing Listener Password

The Listener password can easily be brute forced, since there is no automatic lockout facility and no requirements for strong passwords. Repetitive "set passwords" can be sent to the listener using a hacking program. If logging is enabled (set log_status on), invalid password attempts will appear with an error code of TNS-01169.

Passwords Transmitted in Clear Text

Using the "set password" remotely will transmit the password across the network in clear text on the next command. The "change password" command does encrypt the password.

3.5 Logging

By default, logging is not enabled (LOG_STATUS=OFF). When logging is enabled, the default directory is $ORACLE_HOME/network/admin and the log file default is <sid>.log. The logfile contains a history of listener commands issued both locally and remotely.

The logfile shows a timestamp, command issued, and result code. If an Oracle error is returned, it will include the error message. The logfile does not contain passwords or other significant information.

The logfile does NOT show any information related to IP address, client name, or other identifying information for remote connections. It may show the client's current user name, but this can easily be spoofed or not provided.

```
21-AUG-2002 14:15:00 * stop * 1169
TNS-01169: The listener has not recognized the password
21-AUG-2002 14:15:08 * stop * 1169
TNS-01169: The listener has not recognized the password
21-AUG-2002 14:15:39 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=sun100)(USER=oracle))(
COMMAND=status)(ARGUMENTS=64)(SERVICE=LISTENER)(VERSION=135295232)) * status * 0
21-AUG-2002 14:15:49 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=sun100)(USER=oracle))(
COMMAND=status)(ARGUMENTS=64)(SERVICE=dev1)(VERSION=135295232)) * status * 0
21-AUG-2002 14:16:33 * log_status * 0
21-AUG-2002 14:16:43 * log_file * 0
21-AUG-2002 14:19:01 * service_update * dev1 * 0
```

Tracing may be enabled, but it is rare to have tracing set at the Listener level. Tracing captures all SQL*Net traffic, not just Listener commands. Usually only for debugging purposes is trace enabled. Local Listener commands are not captured in trace files, only remote commands since they traverse the network using SQL*Net.

4

Securing the Listener

Step 1 – Set the Listener Password

[MANDATORY]

Set the Listener password to stop most attacks and security issues. This is usually a simple process. You should set the password using LSNRCTL, which will encrypt the password stored in listener.ora. Setting the password manually in listener.ora using the "PASSWORDS_<listener name>" parameter will result in the password being stored in cleartext.

```
LSNRCTL> set current_listener <listener name>
LSNRCTL> change_password
Old password: <hit enter if no password is set>
New password: <enter new listener password>
Reenter new password: <enter new listener password again>
LSNRCTL> set password
Password: <enter listener password>
LSNRCTL> save_config
```

Check the listener.ora file to see if there is now a parameter PASSWORDS_<listener name>.

Step 2 – Turn on Logging

[MANDATORY]

Turn on logging for all listeners in order to capture Listener commands and brute force password attacks.

```
LSNRCTL> set current_listener <listener name>
LSNRCTL> set password
Password: <enter listener password>
LSNRCTL> set log_directory <oracle_home path>/network/admin
LSNRCTL> set log_file <sid name>.log
LSNRCTL> set log_status on
LSNRCTL> save_config
```

Step 3 – Set ADMIN_RESTRICTIONS in Listener.ora

[MANDATORY]

All runtime modifications to the Listener can be disabled by setting the parameter ADMIN_RESTRICTIONS_<listener name> to ON in the listener.ora file. This parameter stops all

SET commands from being executed either locally or remotely. All changes must be made manually to the listener.ora file.

```
LISTENER.ORA

ADMIN_RESTRICTIONS_<listener name> = ON
```

Restart the listener using the RELOAD command in LSNRCTL for this change to take effect. Any future changes must be made in the listener.ora file, not using SET commands in LSNRCTL. After making any changes to the listener.ora file, use the RELOAD command (or STOP and START) in LSNRCTL.

Step 4 – Apply Listener Patches

[MANDATORY]

Apply the latest listener patches for your database version. See Chapter 5 for the latest patches as of August 2002.

Step 5 – Block SQL*Net on Firewalls

[MANDATORY]

SQL*Net traffic should not be allowed to pass through firewalls unless absolutely necessary. Firewall filters should be designed to only allow SQL*Net traffic from known application and web servers. Few applications require direct SQL*Net to a database from the Internet. SQL*Net performs poorly over high latency networks, thus is seldom used in Interent applications. If applications do require direct SQL*Net access, configure firewall filters based on a specific host and port number.

Step 6 – Secure the $TNS_ADMIN Directory

[MANDATORY]

The Listener password is stored in the listener.ora file. Manually editing the file, the password can easily be removed. If the password was manually added to the file, it is stored in clear text. When added through lsnrctl, it will be stored using a simple hash algorithm (which most likely can easily be broken).

The permissions on the $TNS_ADMIN (usually $ORACLE_HOME/network/admin) should be read/write/execute for only the primary oracle account and no permissions for any other account.

Step 7 – Remove Unused Services

[MANDATORY]

Many default installations have a listener entry for PL/SQL External Procedures (ExtProc). The entry name is usually ExtProc or PLSExtProc. Often ExtProc is installed by default, but is not used.

Check with your application development team or package documentation to determine if ExtProc is used.

If ExtProc is not used, remove it from the listener.ora file. There are several exploits directed at ExtProc.

Since listener.ora files are often copied between instances, they may contain old and unused entries. Check all the other services to determine if they are used. Delete any services not actively used.

Step 8 – Setup Valid Node Checking

[OPTIONAL]

Depending on the type of application and network configuration, valid node checking can be a powerful tool to restrict most traffic from the Listener. Most web applications only require access to the Listener from the application servers and a limited number of clients for administration.

The simplest method to determine valid IP addresses for node checking is through database auditing. We recommended you always have session level auditing enabled.

For Oracle 9i, the valid node checking lines are added to the $ORACLE_HOME/network/admin/sqlnet.ora file. For Oracle 8/8i, the lines are added to the $ORACLE_HOME/network/admin/protocol.ora file.

```
tcp.validnode_checking = yes
tcp.invited_nodes = (x.x.x.x | name, x.x.x.x | name)
tcp.excluded_nodes=( x.x.x.x | name, x.x.x.x | name)
```

Include either the invited_nodes or excluded_nodes, but do not use both. Wildcards, subnets, etc. are not valid, only individual IP addresses or host names are allowed. For more sophisticated checking, use Oracle Connection Manager.

The Listener must be stopped and started for valid node checking to become active. There is no hard limit on the number of nodes that can be included, but for a large number of entries using Oracle Connection Manager may be a better solution.

If many clients require direct SQL*Net access to the database, it is often difficult to use valid node checking due to constantly changing network configurations. In these configurations, SQL*Net traffic must be properly blocked at the network perimeter.

Step 9 – Monitor the Logfile

[OPTIONAL]

The logfile in Step 2 will contain a TNS-01169 error for each invalid password attempt. A brute force password attack will generate hundreds or thousands of these errors. Using a simple shell script or management tools, monitor the logfile and generate an alert whenever the number of TNS-01169 errors reaches a predetermined threshold.

5

Oracle TNS Listener Patches

To find the latest patches, search for available patches on Metalink setting the "Product Family" as "Oracle Net".

The following is a summary of patches available per Oracle release. This list may not be complete as new alerts may have been issued or as patches are ported to different releases. Please check Metalink for the latest patches available for your release.

Oracle9i Release 2 (9.2.x)

2540219 – Oracle Security Alert #42
2395416 – Oracle Security Alert #40
2467947 – Oracle Security Alert #38

Oracle9i (9.0.x)

2540219 – Oracle Security Alert #42
2395416 – Oracle Security Alert #40
2467947 – Oracle Security Alert #38
2367681 – Oracle Security Alert #34 (VMS and Windows only)

Oracle8i (8.1.x)

2540219 – Oracle Security Alert #42
2395416 – Oracle Security Alert #40
Metalink Note 151292.1 -- Oracle Net8 Fragmentation Attack
Metalink Note 151291.1 – Maximum Transport Data Size Too Small (Solaris only)
Metalink Note 151290.1 – Requester_version Value Incorrect (UNIX only)
Metalink Note 151261.1 – Offset_to_data Value Too Large (UNIX only)
Metalink Note 151260.1 – Malformed packet DoS
Metalink Note 151259.1 – Buffer Overflow
Metalink Note 124742.1 – LOG_FILE and TRC_FILE Spoofing

Oracle8 (8.0.x)

2540219 – Oracle Security Alert #42
1864109 – Multiple Vulnerabilities
Metalink Note 124742.1 – LOG_FILE and TRC_FILE Spoofing

Oracle7 (7.3.4)

2540219 – Oracle Security Alert #42
2395416 – Oracle Security Alert #40
Metalink Note 124742.1 – LOG_FILE and TRC_FILE Spoofing

6

Oracle TNS Listener Default Ports

Port Number	Description
1521	Default port for the TNS Listener. This port number may change in the future as Oracle has officially registered ports 2483 and 2484 (SSL).
1522 – 1540	Commonly used ports for the TNS Listener
1575	Default port for the Oracle Names Server
1630	Default port for the Oracle Connection Manager – client connections
1830	Default port for the Oracle Connection Manager – administrative connections
2481	Default port for Oracle JServer/JVM listener
2482	Default port for Oracle JServer/JVM listener using SSL
2483	New officially registered port for the TNS Listener
2484	New officially registered port for the TNS Listener using SSL

Table 6.1 – Default Ports

7

References

- *Oracle 9i Net Services Administrator's Guide*
- *Oracle 8.1.6 Net8 Administrator's Guide, December 1999*
- Metalink Note: 92602.1 "How to Password Protect the Listener"
- Metalink Note: 124742.1 "Vulnerability in the Oracle Listener Program"
- http://www.jammed.com/~jwa/hacks/security/tnscmd/tns-advisory.txt
- "Hack Proofing Oracle", Howard Smith, Oracle Corporation UK Limited
- Oracle Metalink – http://metalink.oracle.com
- Oracle Security Alerts – http://technet.oracle.com/deploy/security/alerts.htm

8

Third Party Tools

There are a few third-party tools available that can remotely control or obtain information from the listener.

- ***Oracle Auditing Tools* – cqure.net**

  A set of Java-based tools that can query the Listener for information.
  http://www.cqure.net/tools07.html

- ***Get SID Enumeration* – cqure.net**

  A set of Java-based tools that can query the Listener for information.
  http://www.cqure.net/tools05.html

- ***tnscmd.pl*** - James W. Abendschan

  The original perl script that queries information from the Listener.
  http://www.jammed.com/~jwa/hacks/security/tnscmd/