INTEGRIGY

WHITE PAPER

Oracle Database Listener Security Guide

APRIL 2007

ORACLE DATABASE LISTENER SECURITY GUIDE

October 2002
March 2003 – Updated
January 2004 – Updated
July 2004 – Updated
March 2005 – Updated
April 2007 - Updated

Authors: Stephen Kost and Jack Kanter

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

The Information contained in this document includes information derived from various third parties. While the Information contained in this document has been presented with due care, Integrigy Corporation does not warrant or represent that the Information is free from errors or omission. The Information is made available on the understanding that Integrigy Corporation and its employees and agents shall have no liability (including liability by reason of negligence) to the users for any loss, damage, cost or expense incurred or arising by reason of any person using or relying on the information and whether caused by reason of any error, negligent act, omission or misrepresentation in the Information or otherwise. Furthermore, while the Information is considered to be true and correct at the date of publication, changes in circumstances after the time of publication may impact on the accuracy of the Information. The Information may change without notice.

Table of Contents

OVERVIEW **5**

Introduction 5
Why Protect the Listener 5
Scope and Database Versions 6
Terminology 6

LISTENER OVERVIEW **7**

Listener Details 7
Listener Modes 7
Listener Remote Management 7

LISTENER EXPLOITS **9**

Listener Remote Management 9
Listener Information Leakage 11
Known Exploits – Oracle Security Alerts 11
Other Exploits 13

LISTENER INFORMATION **14**

Oracle Listener Password 14
Oracle 10g Local OS Authentication 14
Logging 15

SECURING THE ORACLE 8I AND 9I LISTENER **16**

Step 1 – Set the Listener Password 16
Step 2 – Turn on Logging 16
Step 3 – Set ADMIN_RESTRICTIONS in Listener.ora 17
Step 4 – Apply Listener Security Patches 17
Step 5 – Block SQL*Net on Firewalls 17
Step 6 – Secure the $TNS_ADMIN Directory 17
Step 7 – Secure tnslsnr and lsnrctl 18
Step 8 – Remove Unused Services 18
Step 9 – Change the TNS Port Number from 1521 18
Step 10 – Setup Valid Node Checking 18
Step 11 – Monitor the Logfile 19

SECURING THE ORACLE 10G LISTENER **20**

Step 1 – Turn on Logging 20
Step 2 – Apply Listener Security Patches 20
Step 3 – Block SQL*Net on Firewalls 20
Step 4 – Secure the $TNS_ADMIN Directory 20
Step 5 – Secure tnslsnr and lsnrctl 21
Step 6 – Remove Unused Services 21
Step 7 – Change the TNS Port Number from 1521 21
Step 8 – Setup Valid Node Checking 21
Step 9 – Monitor the Logfile 22

ORACLE TNS LISTENER PATCHES **23**

Previous Listener Security Patches 23

ORACLE TNS DEFAULT PORTS 24

REFERENCES 25

THIRD PARTY TOOLS 26

ABOUT INTEGRIGY 27

Overview

Introduction

The Oracle Database Listener is the database server software component that manages the network traffic between the Oracle Database and the client. The Oracle Database Listener listens on a specific network port (default 1521) and forwards network connections to the Database. The Listener is comprised of two binaries: (1) tnslsnr which is the Listener itself and (2) the Listener Control Utility (lsnrctl) which is used to administer the Listener on the server or remotely.

Through our security assessments, Integrigy has consistently identified poor Oracle Database Listener security as a significant security risk. The majority of Oracle Database Listeners are not properly secured as recommended by Oracle and security experts. Fortunately in Oracle 10g, the default Listener configuration is much more secure.

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

Why Protect the Listener

One of the most misunderstood security issues with the Oracle Database is the security of the Listener. Generally, DBAs are not aware that an attacker can easily remotely manage the Listener and potentially effectively take control of the server. The default installation of the Oracle Database prior to Oracle 10g, allows any client to remotely administer a Listener using the "lsnrctl" program or by issuing commands directly to the Listener. Oracle 10.1 and above by default restrict all remote administration of the Listener, unless security is explicitly turned off in the configuration file.

The following are some examples of possible attacks against an Oracle 8i/9i Listener which has a default configuration and is not properly secured. These attacks can be used to exploit a database which even has the most recent Oracle Critical Patch Update security patches applied.

Attack	Description
Execute SQL as DBA	It is possible to overwrite the ORACLE_HOME/sqlplus/admin/glogin.sql by changing the location of the log file and then sending SQL statements in Listener commands to the file. When the SQL*Plus is executed locally on the server (usually by a DBA), then the SQL statements are executed during the SQL*Plus startup.
Allow Login via rlogin	The Listener log can be used to overwrite an .rlogin file with additional host information, thus allowing an attacker access the server using rlogin.
Denial of Service (DoS)	An attacker is able to – ▪ Stop the Listener ▪ Set a Listener password so that the Listener ca not be started without a password, although the DBA simply has to edit the listener.ora file and remove the password line
Denial of Service (DoS)	Undermine the stability of the server and database by overwriting arbitrary files by changing the directory and filename of the log and trace files to any location accessible by the operating system account that owns the database (usually "oracle").
Denial of Service (DoS)	Setting the Listener trace level to "support" may cause performance degradation on a heavily accessed database server.
Information Disclosure	Obtain detailed information on the Listener configuration and database installation such as – ▪ Database Service Names (e.g., SIDs) ▪ Database and Listener versions ▪ Log and trace settings including directory and file names ▪ Security settings ▪ Database server operating system ▪ Oracle environmental variables (ORACLE_HOME, etc.)

Scope and Database Versions

This paper will focus only on the most widely deployed Listener configurations running supported versions of the Oracle Database. This paper assumes the Listener has been configured to use TCP/IP and local naming (TNSNAMES.ORA) is used. No references will be made to Oracle Connection Manager, other naming methods (like LDAP), or advanced configurations like load balancing or Transparent Application Failover (TAF). Use of Advanced Security Option (ASO) with encryption of SQL*Net traffic should have no impact on any of the information in this paper.

The information is this document is based on Oracle 8i, Oracle 9i, and Oracle 10g. All testing was performed on 8.1.7.4, 9.0.1.4, 9.2.0.7, 10.1.0.4, and 10.2.0.1 using Linux, Solaris, and Windows Server operating systems. The information should be valid for 7.3.x, 8.0.x, and "11g", but has not been tested.

Terminology

For clarity and convenience, the Oracle Database Listener will be referred to as the "Listener" throughout this document. The Listener may also be referred to as the "Oracle Net Listener" or the "Oracle TNS Listener". Transparent Network Substrate (TNS) is the network protocol used by Oracle for connectivity to Oracle Databases.

Listener Overview

The Oracle Database Listener is the server 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.

Listener Details

The relevant files for the Listener are as follows –

`$ORACLE_HOME/bin/lsnrctl`	Listener control program
`$ORACLE_HOME/network/admin/listener.ora`	Configuration file for the Listener
`$ORACLE_HOME/network/admin/sqlnet.ora`	Configuration file for the Listener
`$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** and **sqlnet.ora** files for configuration information, such as port numbers and database service names.

The **tnslnsr** processes starts with the process owner of the **lsnrctl** program, usually the "oracle" account on UNIX or the local system account on Windows NT/2000/2003. Any successful exploit will thus gain the privileges for this account.

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 products, such as the Oracle E-Business Suite and Oracle Collaboration Suite.

Listener Remote Management

During our security assessments, we are always amazed to discover that some DBAs are not aware that the Listener in Oracle 8i/9i can be remotely managed using **lsnrctl** or a similar program from a remote machine. The Oracle 10g Listener by default cannot be remotely managed unless local OS authentication is disabled.

The simplest method to remotely issue commands to a Listener is to use **lsnrctl** with command-line parameters as such –

```
lsnrctl <command> <ip_address>:<port>

lsnrctl status 192.168.1.100
lsnrctl stop 192.168.1.100:1522
```

The Listener can also be remotely managed by creating one or more alias entries in a **listener.ora** file. The Listener uses the TNS_ADMIN environmental variable to locate the **listener.ora** file, so multiple configuration files can be setup by setting TNS_ADMIN prior to executing **lsnrctl**.

To set up a computer to remotely administer a Listener:

1. Install Oracle software that contains SQL*Net or Net8, which is usually the Oracle client. You may have to copy the **lsnrctl** (or **lsnrctl.exe**) executable from a server installation 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 similar 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. A **listener.ora** file can contain multiple entries.

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

   The **lsnrctl** control utility support multiple syntaxes for the alias entry, but the above syntax appears to work on most supported versions of the **lsnrctl**. The **lsnrctl** control utility must be restarted after any changes to **listener.ora** since it only reads the file upon startup.

3. Start from the command line **lsnrctl** and specify the Listener name –

   ```
   lsnrctl
   LSNRCTL> set current_listener <alias>
   ```

   By default, all **lsnrctl** commands can be issued remotely, except for **start**. Instead of using the **set current_listener** command, the alias name can be included in the command as such –

   ```
   LSNRCTL> services <alias>
   ```

Listener Exploits

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	Risk	Blocked by admin restrictions	Password Required if Set	Comments
change_password	medium		x	Change password and prevent DBA from controlling the Listener, although listener.ora can be manually edited to remove the password
reload	-		x	Activate some changes made using set commands
save_config	-		x	Make Listener changes permanent
services	leakage		x	Detailed information on Listeners including full paths and environmental variables
set connect_timeout (8.1.7 only)	low	x		A possible denial of service for some connections by setting to 1 second or the maximum value (2,147,483,647) and flood the listener with connection attempts
set inbound_connect_timeout (10g only)	low	x		A possible denial of service for some connections by setting to 1 second or the maximum value (3600) and flood the listener with connection attempts
set display_mode	-	x		Sets the amount of information displayed by some commands such as **services**
set log_directory **set log_file** **set log_status**	high	x	x x x	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 password	-	x		Does not set the password, but rather provides the password for authentication. It is important to note that the encrypted password string can also be supplied prior to Oracle 10g. See **change_password** for setting a new password.
set save_config_on_stop	-	x		Saves the configuration when the listener is stopped
set statup_waittime	low	x	X	Set to maximum value (2,147,483,647 seconds) so the Listener never starts
set trc_directory **set trc_file** **set trc_level** **trace**	medium	x	Only trc_level and trace	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 (8.1.7 only)	-	x		Register the server with an Oracle Names server.
show	leakage		x*	* all shows commands, except on 8.1.7 password only needed for show save_config_on_stop and show use_plugandplay
spawn	medium		x	Very specific to the implementation
start	-		local only	
status	leakage		9.2+	Detailed information regarding operating system, databases, and services
stop	high		x	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 1 – Listener Commands

Listener Information Leakage

The Listener provides information regarding the enabled services, which includes database services 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 if a password is not set. Prior to 9.2, the **status** command did not require a password. The Listener version (and database version) can always be retrieved using the **version** command, which does not require a password. The **services** command does require a password if it is enabled.

<table>
<tr><th>Command</th><th>Password Required if Set</th><th>Information/Sample Output</th></tr>
<tr><td><b>services</b></td><td>X</td><td><pre>
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
</pre>
(additional information if “displaymode = verbose”, including environmental variables and paths)</td></tr>
<tr><td><b>status</b></td><td>9.2+</td><td><pre>
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
</pre></td></tr>
<tr><td><b>version</b></td><td></td><td><pre>
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,,
</pre></td></tr>
</table>

Table 2 – Information Leakage Commands

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.

Critical Patch Update January 2007

- DB13 – A local buffer overflow exists in the **tnslsnr** executable in Oracle 8.1.7.4 and 9.0.1.x, which may allow an attacker with gain local privileges of the oracle account

Critical Patch Update January 2006

- DB09, DB10, DB11 – Undisclosed vulnerabilities exist in the Listener (most likely buffer overflows), which may allow an attack to perform at least a Denial of Service (DoS) attack

#68 – Denial of Service Vulnerability (8/31/04)

- DB13 – A Denial of Service (DoS) vulnerability in all releases through 10.1.0.2

#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

OTHER EXPLOITS

1. 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 password** commands 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.

2. Passwords Transmitted in Clear Text

Using the **set password** command remotely will transmit the password across the network in clear text with every command. If encryption is setup for the listener using the Advanced Security Option (ASO), then the passwords will be sent encrypted across the network. The **change password** command does encrypt the password when the **lsnrctl** program is used.

Listener Information

Oracle Listener Password

The password for the Listener is stored in the **listener.ora** file. If the **PASSWORDS_<listener name>** parameter is manually set, then the password remains in plain-text. If set using **lsnrctl** and the **change_password** command, then the password is encrypted as 8-byte string. Unlike the database, the Listener password is case-sensitive.

Prior to Oracle 10g, the encrypted password string could be substituted for the actual password when issuing the **set password** command. This is useful in executing scripts to stop the Listener. If a password is set for the Oracle 10g Listener, scripts must use the actual password rather than the encrypted string.

If the Listener password is set to "mypassword", then the **listener.ora** file will have the encrypted string. The following **lsnrctl** commands using either the plain-text password or encrypted string will both work prior to Oracle 10g.

Listener.ora

```
PASSWORDS_LISTENER = F4BAA4A006C26134

LSNRCTL> set password
Password: mypassword
LSNRCTL> set password
Password: F4BAA4A006C26134
```

Oracle 10g Local OS Authentication

A major change to Listener security in Oracle 10g (10.1 and 10.2) was the introduction of Local OS Authentication. By default, the Listener cannot be remotely managed and can only be managed locally by the owner of the **tnslsnr** process (usually oracle).

If another operating system user attempts to manage the Listener, the following message will be displayed in the Listener log file –

```
TNS-01190: The user is not authorized to execute the requested listener command
```

If someone attempts to managed the Listener remotely, the following message will be displayed in the Listener log file –

```
TNS-01189: The listener could not authenticate the user
```

Local OS Authentication can be disabled by setting the **LOCAL_OS_AUTHENTICATION_<listener>** parameter in **listener.ora** file as such –

```
LOCAL_OS_AUTHENTICATION_<listener> = OFF
```

When Local OS Authentication is disabled, the Listener behaves exactly as in Oracle 8i/9i. Thus, it should have a password set and **ADMIN_RESTRICTIONS** set to On.

Even though Local OS Authentication is enabled, a local user is still able to issue the **status**, **services**, and **version** commands.

The current state of Local OS Authentication can be displayed using the **status command** in **lsnrctl**. The following line will be display in the status command output –

Message	Password Set	Local OS Authentication
`Security ON: Local OS Authentication`	No	On
`Security ON: Password or Local OS Authentication`	Yes	On
`Security ON: Password`	Yes	Off

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.

The following are TNS errors that may signify an attack or inappropriate activity –

Error Code	Message	Comments
`TNS-01169`	`The listener has not recognized the password`	An attempt was made to issue a command, but a password is set
`TNS-01189`	`The listener could not authenticate the user`	Oracle 10g – Local OS Authentication is enabled and attempt was made to manage the Listener remotely or locally by another user
`TNS-01190`	`The user is not authorized to execute the requested listener command`	Oracle 10g – Local OS Authentication is enabled and attempt was made to manage the Listener locally by another user
`TNS-12508`	`The listener could not resolve the COMMAND given`	This error occurs when an invalid command is issue (e.g., **statusx** instead of **status**) or when a **set** command is issued and **ADMIN_RESTRICTIONS** is set to no.

Securing the Oracle 8i and 9i Listener

This section is for the Oracle Listener versions 8.1.7.x, 9.0.1, and 9.2. For the Oracle 10g Listener, if Local OS Authentication is disabled then these steps should be taken to properly secure 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>**. It is important to remember that the actual encrypted password can be used in place of the actual password prior to Oracle 10g.

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 Security Patches [MANDATORY]

Apply at least the January 2006 Critical Patch Update for the latest Listener security patches (as of March 2007). Critical Patch Updates are cumulative, therefore, the latest patch will contain all previous security patches for the Listener.

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. SQL*Net traffic from application servers in the DMZ should be permitted only to access specific database servers.

Few applications require direct SQL*Net connections to a database from the Internet. SQL*Net performs poorly over high latency networks, thus is seldom used in Internet 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 or changed. If the password was manually added to the file, it is stored in clear text. When added through **lsnrctl**, it will be stored as an encrypted string.

The permissions on the **listener.ora**, **sqlnet.ora**, and **protocol.ora** files in the $TNS_ADMIN directory (usually $ORACLE_HOME/network/admin) should be read/write/execute for only the primary oracle account and no permissions for any other account (for UNIX and Linux 0600). The **tnsnames.ora** file permissions should be set to 0644 on UNIX and Linux.

Step 7 – Secure tnslsnr and lsnrctl

[OPTIONAL]

The **tnslsnr** and **lsnrctl** executables in the $ORACLE_HOME/bin directory should be protected and file permissions should be set to 0751 on UNIX and Linux as recommended by Oracle. It is possible to change the file permissions to 0700 which would be more secure, although this should be thoroughly tested in your environment.

Step 8 – 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 application 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 sometimes 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 9 – Change the TNS Port Number from 1521

[OPTIONAL]

In order to help stop automated attacks and detection of the Listener in networks, the default NTS port number should be changed from 1521 to a port outside of the 1521-1550 and 1600-1699 ranges. This will provide only minimal additional security through obscurity, but may thwart an automated attack or simply scanning for Oracle Databases on port 1521.

The port number can be changed using Oracle Net Manager (**netmgr**) or editing the **listener.ora** file directly. All **tnsnames.ora** files on the database server and any clients must be updated to reflect the change in the port number. The database initialization parameter **LOCAL_LISTENER** must be set so that the database is able to dynamically register with the Listener. See Metalink Note ID 359277.1 "Changing Default Listener Port Number" for more information.

Step 10 – 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/10g, 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.

STEP 11 – MONITOR THE LOGFILE

[OPTIONAL]

The logfile in Step 2 may contain TNS-01169, TNS-01189, TNS-01190, or TNS-12508 errors, which may signify attacks or inappropriate activity. Using a simple shell script or management tools, monitor the logfile and generate an alert whenever these errors reaches are encountered.

Securing the Oracle 10g Listener

Due to changes in the Oracle 10g Listener (10.1 and 10.2), separate instructions are required. With Local OS Authentication, a Listener password does not need to be set nor ADMIN_RESTRICTIONS. However, there are undisclosed security vulnerabilities that may allow an attacker to bypass Local OS Authentication. If maximum security is required, a Listener password should be set and ADMIN_RESTRICTIONS should be enabled.

Step 1 – 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 2 – Apply Listener Security Patches

[MANDATORY]

Apply at least the January 2006 Critical Patch Update for the latest Listener security patches (as of March 2007). Critical Patch Updates are cumulative, therefore, the latest patch will contain all previous security patches for the Listener.

Step 3 – 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. SQL*Net traffic from application servers in the DMZ should be permitted only to access specific database servers.

Few applications require direct SQL*Net connections to a database from the Internet. SQL*Net performs poorly over high latency networks, thus is seldom used in Internet applications. If applications do require direct SQL*Net access, configure firewall filters based on a specific host and port number.

Step 4 – 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 or changed. If the password was manually added to the file, it is stored in clear text. When added through **lsnrctl**, it will be stored as an encrypted string.

The permissions on the **listener.ora**, **sqlnet.ora**, and **protocol.ora** files in the $TNS_ADMIN directory (usually $ORACLE_HOME/network/admin) should be read/write/execute for only the primary oracle account and no permissions for any other account (for UNIX and Linux 0600). The **tnsnames.ora** file permissions should be set to 0644 on UNIX and Linux.

Step 5 – Secure tnslsnr and lsnrctl

[OPTIONAL]

The **tnslsnr** and **lsnrctl** executables in the $ORACLE_HOME/bin directory should be protected and file permissions should be set to 0751 on UNIX and Linux as recommended by Oracle. It is possible to change the file permissions to 0700 which would be more secure, although this should be thoroughly tested in your environment.

Step 6 – 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 application 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 sometimes 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 7 – Change the TNS Port Number from 1521

[OPTIONAL]

In order to help stop automated attacks and detection of the Listener in networks, the default NTS port number should be changed from 1521 to a port outside of the 1521-1550 and 1600-1699 ranges. This will provide only minimal additional security through obscurity, but may thwart an automated attack or simply scanning for Oracle Databases on port 1521.

The port number can be changed using Oracle Net Manager (**netmgr**) or editing the **listener.ora** file directly. All **tnsnames.ora** files on the database server and any clients must be updated to reflect the change in the port number. The database initialization parameter **LOCAL_LISTENER** must be set so that the database is able to dynamically register with the Listener. See Metalink Note ID 359277.1 "Changing Default Listener Port Number" for more information.

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/10g, 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.

Step 9 – Monitor the Logfile

[OPTIONAL]

The logfile in Step 2 may contain TNS-01169, TNS-01189, TNS-01190, or TNS-12508 errors, which may signify attacks or inappropriate activity. Using a simple shell script or management tools, monitor the logfile and generate an alert whenever these errors reaches are encountered.

ORACLE TNS LISTENER PATCHES

With the introduction of the Oracle Critical Patch Updates (CPU), the security patch process for the Listener has been greatly simplified. The Critical Patch Updates are cumulative, therefore, only the latest database CPU patch is required. As of March 2007, the last remotely exploitable Listener security vulnerability was fixed in the January 2006 CPU.

PREVIOUS LISTENER SECURITY PATCHES

This is information is provided for reference only. The latest Critical Patch Update is the only security patch that is required for all supported releases of the Oracle Database. The following is a summary of previous Listener security patches available per Oracle 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

Oracle TNS 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

REFERENCES

- *Oracle 9iR2 Net Services Administrator's Guide*
- *Oracle 9iR2 Net Services Reference Guide*
- Metalink Note ID 92602.1 "How to Password Protect the Listener"
- Metalink Note ID 124742.1 "Vulnerability in the Oracle Listener Program"
- Metalink Note ID 260986.1 "Setting Listener Passwords With an Oracle 10g Listener"
- Metalink Note ID 359277.1 "Changing Default Listener Port Number"
- Metalink Note ID 332785.1 "How To Disable Local Os Authentication For Lsnrctl Utility"
- http://www.jammed.com/~jwa/hacks/security/tnscmd/tns-advisory.txt
- http://www.jammed.com/~jwa/hacks/security/tnscmd/tnscmd-doc.html
- Oracle Metalink – http://metalink.oracle.com
- Oracle Security Alerts – http://technet.oracle.com/deploy/security/alerts.htm
- Oracle TNS Listener Exploits -- http://www.red-database-security.com/exploits/oracle_exploit_listener.html

Third Party Tools

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

- ***Integrigy AppSentry Listener Check (Recommended)***
  A simple Windows GUI tool that checks a number of Listener security settings
  http://www.integrigy.com/security-resources/downloads/lsnrcheck-tool

- ***tnscmd.pl*** - James W. Abendschan
  The original Perl script that queries information from the Listener
  http://www.jammed.com/~jwa/hacks/security/tnscmd/

- ***Oracle Auditing Tools* – cqure.net**
  A set of Java-based tools that can query the Listener for information
  http://www.cqure.net/wp/?page_id=2

- ***Getsids SID Enumeration* – cqure.net**
  A windows command-line tool to get the available databases from the Listener
  http://www.cqure.net/wp/?page_id=13

- ***SidGuesser* – cqure.net**
  A windows command-line tool used to find databases using a dictionary attack
  http://www.cqure.net/wp/?page_id=41

ABOUT INTEGRIGY

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
Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60681 USA
888/542-4802
www.integrigy.com