GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Implementing SSL to Protect Oracle 9i Network Communications

Joe Keegan

GSEC Practical Assignment
v. 1.4

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

TABLE OF CONTENTS

1 Introduction....3

2 Oracle 9i Security Features....3

3 Protecting Network Communications With SSL....3

3.1 SSL Provides Confidentiality....3

3.2 SSL Provides Authentication....4

3.3 SSL Provides Integrity....5

3.4 Components of SSL in an Oracle Environment....5

4 Implementing Oracle Net8 over SSL....6

4.1 Create a Certificate Authority....6

4.2 Configure Oracle Client....7

4.2.1 Create a New Wallet on the Client....7

4.2.2 Sign Client Certificate....8

4.2.3 Installing the Certificates into the client's Oracle Wallet....9

4.2.4 Configure SSL with the Oracle Net Manager....10

4.3 Configure Oracle Server....11

4.3.1 Create a New Wallet on the Server....11

4.3.2 Sign Server Certificate....12

4.3.3 Installing the Certificates into the Server's Oracle Wallet....12

4.3.4 Configure SSL with Oracle Net Manager....13

4.3.5 Create a new Oracle listener....15

4.4 Configure Client Connection....16

4.4.1 Create a Net Service Name....16

4.4.2 Edit the Client's tnsnames.ora File....17

4.5 Connect to the Database Server....17

5 Verification and Troubleshooting....17

6 Oracle SSL and Firewalls....18

7 References....19

© SANS Institute 2000 - 2002, Author retains full rights.

1 Introduction

At the heart of critical systems and housing confidential data, databases are often the crown jewels of most companies. In spite of the importance of a company's database, they are typically overlooked in many aspects of information security.[1] Though database security varies from organization to organization, it seems that databases are commonly administrated with unencrypted network protocols. Most communication with the database is often within an organization, but that does not remove the risk that your database will be attacked. An inside attacker or outside attacker who has compromised other systems on your network could sniff the clear-text network traffic gaining valuable information about the data stored in your database. It's for that reason that CERT and others recommend that all remote administration of systems, including databases, be performed via an encrypted protocol. [2]

Oracle provides advanced security options to encrypt SQL traffic via the industry standard Secure Socket Layer (SSL) protocol. This paper will cover the features available in Oracle's advanced security options and how to implement SSL to protect communications with the database.

2 Oracle 9i Security Features

Oracle 9i has many new features which can greatly improve the security of your database. [3]

Strong Authentication - Oracle 9i supports user authentication via SecurID, Kerberos, Cybersafe, RADIUS, DCE and PKI authentication using X.509v3 digital certificates.

Encryption – In addition to SSL, Oracle 9i provides data encryption of network communication supporting RC4_40, RC4_56, RC4_128, RC4_256, DES, DES_40, 2-Key 3DES, and 3-Key 3DES encryption algorithms. Data integrity is provided with support for both MD5 and SHA1.

Enterprise User Security – Oracle 9i can store users, roles and credentials in an LDAP compliant enterprise directory. SSL is available to provide data encryption, data integrity and authentication.

3 Protecting Network Communications With SSL

Oracle 9i implements the SSL protocol for encryption of data exchanged between the database client and server. [4] Developed by Netscape, SSL has become a de facto standard for transport layer encryption and provides connection security with three basic properties: Confidentiality, Authentication, and Integrity. [5]

3.1 SSL Provides Confidentiality

Oracle 9i by default encrypts user credentials stored in the database and while in transit over the network. [6] Though Oracle provides confidentiality for user credentials, it does not protect the data transferred in the session, allowing an attacker to eavesdrop on the

© SANS Institute 2000 - 2002, Author retains full rights.

connection. An attacker eavesdropping on a session could obtain confidential information such as credit card numbers or user credentials for web based applications. Additionally it is possible for an attacker to gather information needed to compromise the database. For instance the SQL statement `alter user system identified by G00g13 account unlock` changes the system password to `g00g13`. This information can be sniffed with most common packet sniffers and would allow an attacker to login into the database as the system user.
Below is an example of a tcpdump with the SQL command bolded.

```
16:46:40.819058 192.168.1.107.1194 > 192.168.1.10.1521: P 149:297(148) ack 108 win 7396
(DF)
0x0000   4500 00bc 5334 4000 8006 2342 c0a8 016b        E...S4@...#B...k
0x0010   c0a8 010a 04aa 05f1 004f d684 8a9d 28f6        .........O....(.
0x0020   5018 1ce4 75c2 0000 0094 0000 0600 0000        P...u...........
0x0030   0000 1169 1f04 d18d 0001 0101 0103 5e20        ...i.........^.
0x0040   0280 2100 3891 8d00 0136 108a 8c00 010a        ..!.8....6......
0x0050   0000 0000 348a 8c00 0001 0100 0000 0000        ....4...........
0x0060   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0070   0036 8a8c 0036 616c 7465 7220 7573 6572        .6...6alter.user
0x0080   2073 7973 7465 6d20 6964 656e 7469 6669        .system.identifi
0x0090   6564 2062 7920 4730 3067 3133 2061 6363        ed.by.G00g13.acc
0x00a0   6f75 6e74 2075 6e6c 6f63 6b0a 0101 0101        ount.unlock.....
0x00b0   0000 0000 0001 0700 0313 0005                  ............
```

After an initial handshake to define a secret key, SSL encrypts all session data with symmetric cryptography, providing confidentiality for all data transferred in the session. Once SSL has been implemented all database administration traffic will be encrypted and an attacker will no loner be able to obtain confidential information by sniffing traffic. Below is a tcpdump of the same SQL statement from above, now protected by SSL.

```
17:15:24.878473 192.168.1.107.1212 > 192.168.1.10.32906: P 16948846:16949023(177) ack
2011182713 win 8222 (DF)
0x0000   4500 00d9 1194 4000 8006 64c5 c0a8 016b        E.....@...d....k
0x0010   c0a8 010a 04bc 808a 0102 9e6e 77e0 3679        ...........nw.6y
0x0020   5018 201e d814 0000 1703 0000 acb0 184f        P..............O
0x0030   d6be 1968 c4d5 b1ff 5e04 b0b4 b490 fbf6        ...h....^.......
0x0040   b1c9 4062 9625 5daa 8053 649c 48cd b79f        ..@b.%]..Sd.H...
0x0050   db18 e770 57fa 2481 e9fb 6017 c53f 3134        ...pW.$...`..?14
0x0060   c412 8d38 8958 405c 4b0c caaa a96a 1175        ...8.X@\K....j.u
0x0070   59f8 a12a e2e5 9c46 62ac cddd a7d6 e976        Y..*...Fb......v
0x0080   5109 270e c880 bf6f db80 9333 608c e246        Q.'....o...3`..F
0x0090   bf13 895f 8891 bb86 6506 2a6f d4da 40e5        ..._....e.*o..@.
0x00a0   229e 7ffc f074 9bfd 020d 06aa f096 0e0a        "....t..........
0x00b0   d1b1 fb47 86b3 ded8 55fd 5717 d1d1 a2f5        ...G....U.W.....
0x00c0   5d64 ed05 f2e8 09a7 0ae7 6669 07b2 7992        ]d........fi..y.
0x00d0   09d8 4fa6 4132 04d6 a4                         ..O.A2..
```

3.2 SSL Provides Authentication

A database server authenticates a user by verifying the user's credentials, such as a user name and password. But by default the client does not authenticate the server and a user has no way to ensure they have connected to the legitimate server. Without server authentication it's possible for an attacker to set up a service which imitates a database login. This would allow an attacker to harvest database accounts and circumvent any cryptographic protection that may have been implemented. A

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

sophisticated hacker could employ a "man-in-the-middle" attack which would proxy all information in the session through this dummy service to the database. The connection would appear normal and the user would be unaware that the connection was being watched.

SSL authenticates peer's using public key cryptography. The server has a X.509v3 digital certificate which includes it's public key and the name of the database service. When the client connects to a database server using SSL the digital certificate is downloaded by the client. The public key from the digital certificate is used to encrypt a symmetric session key. Since only the servers private key can decrypt the session key only the server can continue to communicate with the client.

A client must also verify that the digital certificate is indeed from the legitimate server. Without verification, what would prevent an attacker from creating his own digital certificate and yet again masquerade as the server? The servers digital certificate is signed by a trusted certificate authority (CA), which can be operated "in-house" or by an outside company providing a CA (such as Verisign). When the client downloads the server's digital certificate, it will verify that it is signed by a trusted CA before it continues with it's communication. The attacker would only be able to fool the client by compromising the CA or tricking the CA into signing a forged certificate.

3.3 SSL Provides Integrity

Like most unencrypted network protocols, Oracle's Net8 protocol is vulnerable to data integrity attacks and session hijacking. An attacker that can see all pertinent session data of a network connection could modify the data in transit. This could be done by intercepting the network communication (such as a "man-in-the-middle") and changing important values. For example an attacker changing the accounts involved in a balance transfer. An attacker that could not perform a "man-in-the-middle" attack could perform a session hijacking attack. A session hijacking attack involves launching a denial-of-service attack against the client and then spoofing the client by forging network packets. The server would be unaware the real client was no longer responsive and accept the forged packets as genuine (since client authentication only happens at the beginning of the session).

SSL connections are encrypted and prevent an attacker from gathering the information needed to perform data integrity attacks. It would not be possible for an attacker to alter the correct data in transit since all the data would appear as random bits. Even if the attack did manage to change something in transit, SSL uses secure hashing functions to provide message integrity checking. If the message was altered the hash would no longer match the message and the message would be discarded. Session hijacking would also be impossible, given that the attacker does not have access to the session key. The attacker would not be able to encrypt the forged network packets and the server would not accept them.

3.4 Components of SSL in an Oracle Environment

The following components are required to implement Net8 over SSL. [7]

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Certificate Authority – A Certificate Authority (CA) is required to sign the digital certificates involved with SSL. In this example OpenSSL (**www.openssl.org**) will be used to generate a CA and sign the digital certificates. [8] Any CA can be used to sign the certificates, but the CA's digital certificate will have to be included in the Oracle client's and server's trusted certificates.

Certificate – A digital certificate for the server will need to be generated and signed by the CA. If you wish to have the server authenticate an authorized client, the client will also need to generate a certificate. Oracle's Wallet Manager will be used to generate the needed certificates and the CA described above will be used to sign them.

Wallet – A wallet is a 3DES encrypted file which stores the digital certificate, private key and trusted certificates used by the Oracle client or server. Oracle Wallet Manager is used to administrate the wallet.

4 Implementing Oracle Net8 over SSL

The steps documented below will guide an administrator to configure an Oracle Windows client and Oracle Unix server to use SSL for Server Authentication, Data Confidentiality and Data Integrity.

4.1 Create a Certificate Authority

As stated above a Certificate Authority (CA) is used to ensure that a digital certificate can be trusted. OpenSSL has a very simple script called CA.pl [9] which can be used to create a CA and to sign certificates. This script is only used for this example. It's not recommended to use the CA.pl script for large PKI installations.

- Download and install the OpenSSL software (www.openssl.org)
- Make sure to include the $PATH_TO_INSTALL/ssl/bin directory (usually `/usr/local/ssl/bin`, which will be used in this example) as part of your $PATH.
- Create the CA's private/public key pair and certificate with the `CA.pl –newca` command. The command will ask you for the information to generate the CA's certificate. There is no requirement for the Common Name of the certificate.

```
# cd /usr/local/ssl/misc
# ./CA.pl –newca
CA certificate filename (or enter to create)

Making CA certificate ...
Using configuration from /usr/local/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....................................................++++++
.......++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase: [This is the pass phrase to the CA's private key]
Verifying password - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

© SANS Institute 2000 - 2002, Author retains full rights.

```
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Sometown
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GIAC Security, Inc
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:GIAC Security, Inc Certificate Authority
Email Address []:ca-master@giac-domain.com
```

Note: This command will ask for the "PEM pass phrase". This is the pass phrase to the CA's private key and it is very important that it is secure. Anyone who gains access to the CA's private key and can guess or obtain the private key pass phrase can impersonate your CA, compromising SSL's authentication mechanisms.

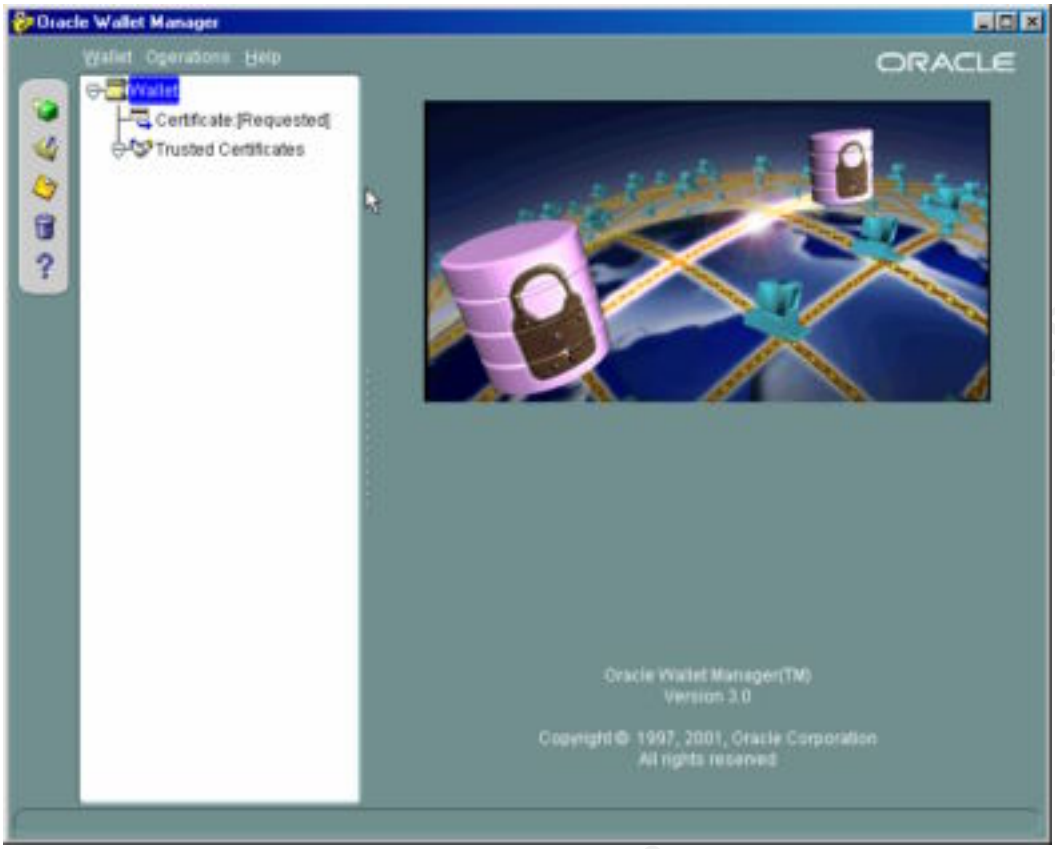
- The new CA's certificates and keys will be located in the `/usr/local/ssl/misc/demoCA` directory.

4.2 Configure Oracle Client

4.2.1 Create a New Wallet on the Client

- Open the Oracle Wallet Manager by selecting Start → Programs → Oracle - *<Oracle_Home_Name>* → Integrated Management Tools → Wallet Manager.
- Create a new wallet file by selecting the Wallet menu and clicking on "New"
- You will be prompted for a wallet password. The wallet password must be eight characters long, include a number or special character and is case sensitive.
- When prompted, click to create a certificate request. The Common Name for the certificate will usually be a fully qualified domain name, but can be anything. Make sure to select 1024 bits for the key size.
- After the certificate request has been created, you need to export the certificate request by right-clicking on the Certificate:[Requested] under the Wallet and then selecting "Export Certificate Request".
- Remove all the Trusted Certificates by right-clicking on the certificate and selecting "Remove Trusted Certificate". Removing the unused trusted certificates, prevents the client from trusting a certificate signed by an untrusted CA.

© SANS Institute 2000 - 2002 Author retains full rights.



Oracle Wallet Manager with a new wallet

4.2.2 Sign Client Certificate

- Copy the certificate to the `/usr/local/ssl/misc` directory on the server running the CA. The certificate must be named **`newreq.pem`** due to simplicity of the CA.pl script.
- Now sign the certificate with the CA (again, make sure `/usr/local/ssl/bin` is in your $PATH)

```
bash-2.03# ./CA.pl -sign
Using configuration from /usr/local/ssl/openssl.cnf
Enter PEM pass phrase:[enter CA private key pass phrase]
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'Ca'
localityName          :PRINTABLE:'Sunnyvale'
organizationName      :T61STRING:'jjk3.com'
commonName            :T61STRING:'client'
Certificate is to be certified until May 17 21:31:14 2003 GMT (365
   days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
```

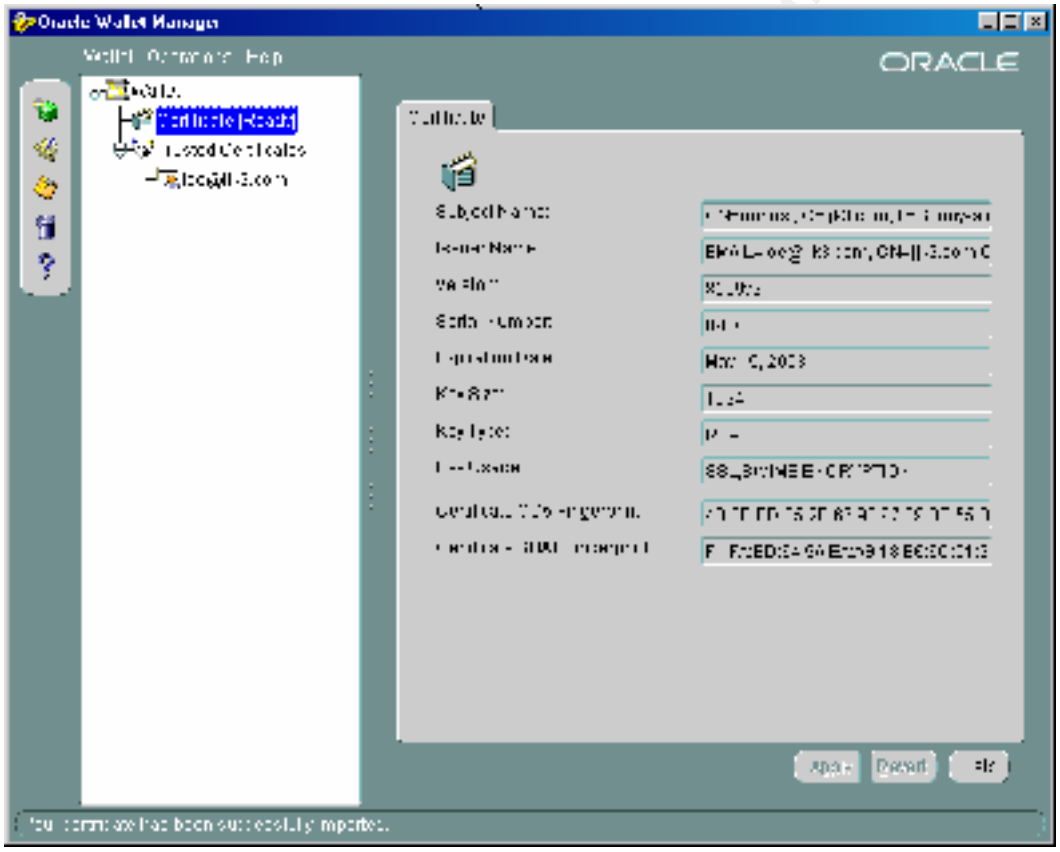
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

```
Data Base Updated
Signed certificate is in newcert.pem
```

- The signed certificate is located at `/usr/local/ssl/misc/newcert.pem` and needs to be copied to the workstation with the Oracle client.
- The CA's certificate, `/usr/local/ssl/misc/demoCA/cacert.pem`, also needs to be copied to the work station.

4.2.3 Installing the Certificates into the client's Oracle Wallet

- In the Oracle Wallet Manager, right click on Trusted Certificates and select "Import Trusted Certificate". Choose to "Select a file that contains the certificate" and then browse to the CA's certificate (`cacert.pem`).
- Next Right click on Certificate:[Requested], under the Wallet icon, and select "Import User Certificate". Choose to "Select a file that contains the certificate" and then browse to the CA's certificate (`newcert.pem`). Once the certificate has been imported the Certificate:[Requested] will change to Certificate:[Ready].
- The Oracle Wallet Manager should now look something like the image below



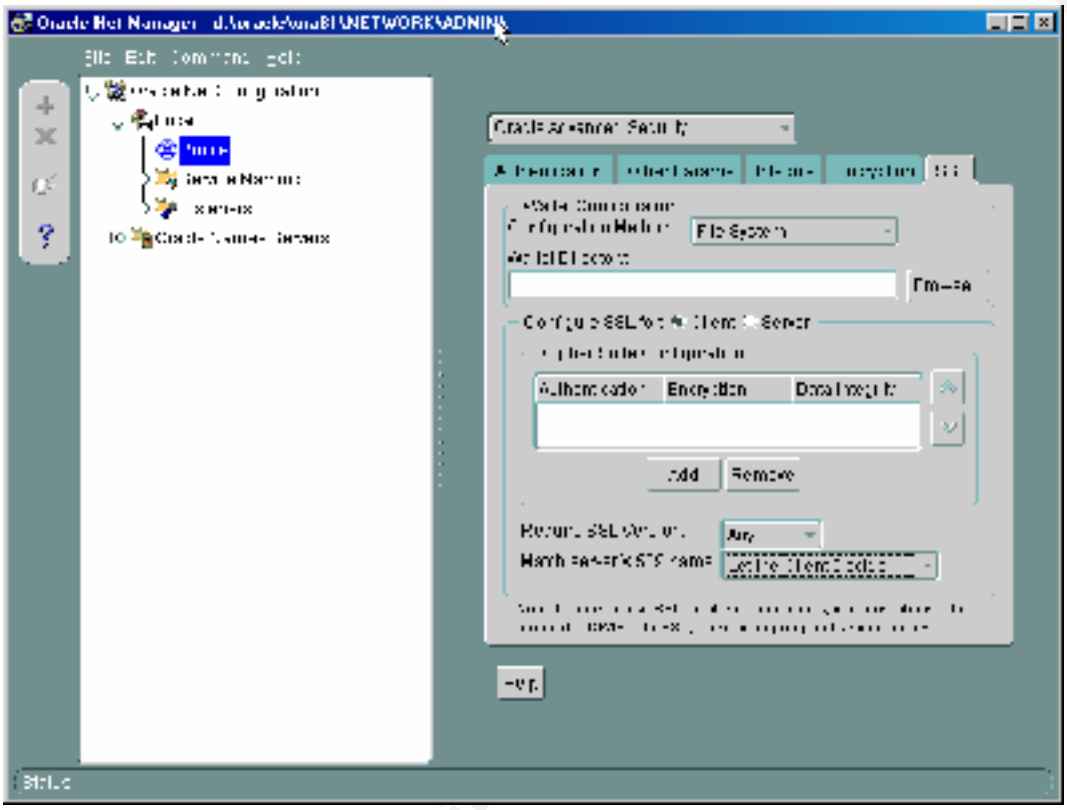
Oracle Wallet Manager after configuration

- Now that both the client and the CA certificates have been installed, choose to save the Wallet, by clicking on the Wallet menu and selecting "Save In System Default" (`C:\TEMP\ORACLE\WALLETS`). After saving the wallet, close the Oracle Wallet Manager. **Note:** Saving the wallet in a location other then the system default causes problems later in the configuration.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

4.2.4 Configure SSL with the Oracle Net Manager

- Open Oracle's Net Manager by selecting Start → Programs → Oracle - *<Oracle_Home_Name>* → Configuration and Migration tools → Net Manager.
- Expand the Local icon and then select Profile
- Click on the drop down menu on the right hand side and select "Oracle Advance Security"
- Select the SSL Tab
- Select the "Client" Radio Button for the "Configure SSL for" parameter



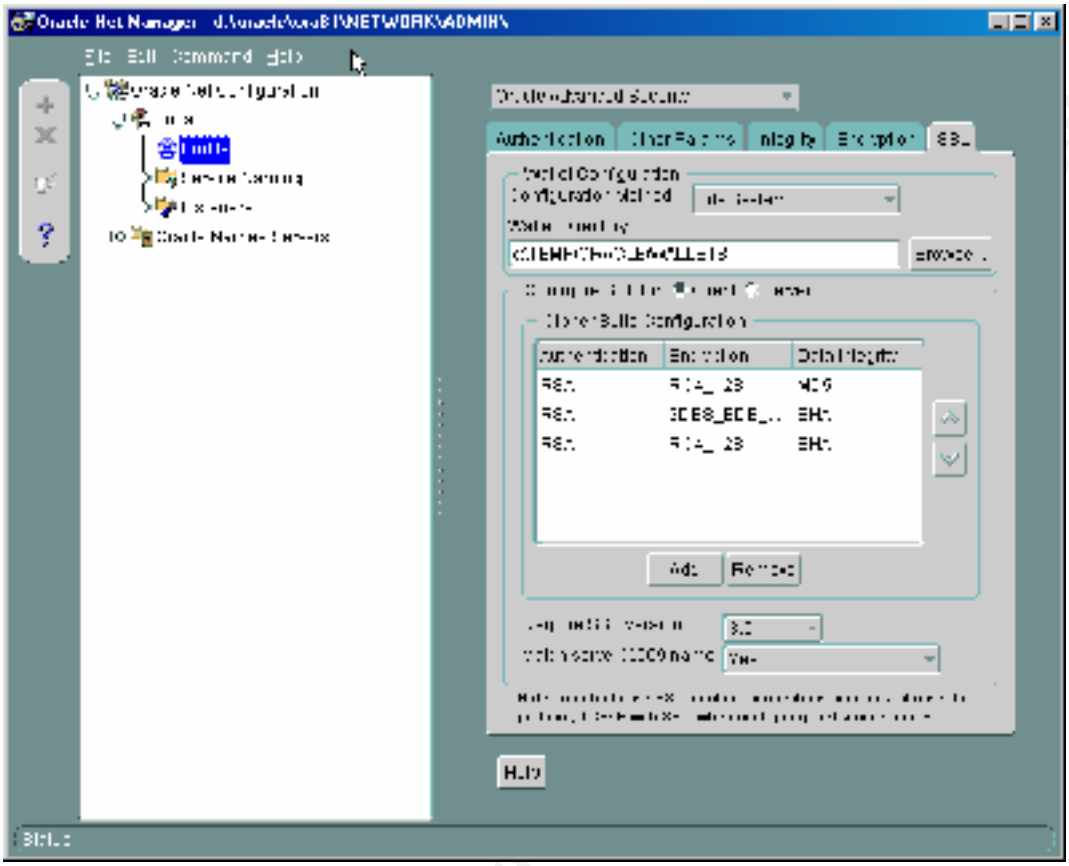
Oracle Net Manager Ready for Client Configuration

- Enter `C:\TEMP\ORACLE\WALLETS` for the location of the Wallet Directory
- Click on the Add button under the "Cipher Suite Configuration". By Default only exportable cipher suites are available, select the "Show US Domestic Cipher Suites" to show all supported suites. Refer to Oracle Advanced Security Administrator's Guide, Release 9.0.1 for a complete list of cipher suites supported (http://download-west.oracle.com/otndoc/oracle9i/901_doc/network.901/a90150/toc.htm)
- Select "RSA – 3DES_EDE_CBC – SHA" and then click OK
- Continue to add both "RSA – RC4_128 – MD5" and "RSA – RC4_128 – SHA"
  Note: Only ciphers with 128-bit encryption keys and RSA authentication are being added. There may be local restrictions on the key size for your specific installation. RSA authentication must be selected for the client to use SSL for authentication, if you do not want to use SSL for authentication the DH-anon (Diffie-Hellman Anonymous) can be selected.
- Select "3.0" for "Require SSL Version"
- Select "Yes" for "Match server X.509 name". This requires the common name of the servers certificate to match it's service name. This adds an extra layer of

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

protection against someone stealing a certificate and attempting to use it for another database.

- Save the changes by clicking on the File menu and select "Save Network Configuration"



Oracle Net Manager for the client configured

4.3 Configure Oracle Server

Most of the utilities used to configure SSL on the Oracle server must display to an X server. If the server that is running Oracle has a video card and monitor you should be able to perform configurations locally. Usually it is not feasible to perform the configuration changes locally and you will need to run an X server on a Windows client and display the GUI's locally. There are a few X servers available for Windows. I have had the best experience with Exceed from Hummingbird Software (http://www.hummingbird.com/). Exceed is a commercial product, though they do have a trial available for download. A free X server available for Windows is Xfree86, by the Cygwin project (http://www.cygwin.com). Xfree86 works well and is the X server used for this paper.

4.3.1 Create a New Wallet on the Server

- Log into the Oracle server as the "Oracle" user
- Open up the Oracle Wallet Manager located at $ORACLE_HOME/bin/owm
- Create a new wallet file by selecting the Wallet menu and clicking on "new"
- You will be prompted for a wallet password. The wallet password must be eight characters long, include a number or special character and is case sensitive.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

- When prompted, click to create a certificate request. The Common Name for the certificate must be the same as the database's global name/service name. Make sure to select 1024 bits for the key size.
- After the certificate request has been created, you need to export the certificate request by right-clicking on the Certificate:[Requested] under the Wallet and then selecting Export Certificate Request.
- Remove all the Trusted Certificates by right-clicking on the certificate and selecting "Remove Trusted Certificate". Removing the unused trusted certificates, prevents the server from trusting a certificate signed by an untrusted CA.

4.3.2 Sign Server Certificate

- Copy the certificate to the `/usr/local/ssl/misc` directory on the server running the CA. The certificate must be named **`newreq.pem`** due to simplicity of the CA.pl script.
- Now sign the certificate with the CA (again, make sure `/usr/local/ssl/bin` is in your PATH)

```
bash-2.03# ./CA.pl -sign
Using configuration from /usr/local/ssl/openssl.cnf
Enter PEM pass phrase:[enter CA private key pass phrase]
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'Ca'
localityName          :PRINTABLE:'Sunnyvale'
organizationName      :T61STRING:'jjk3.com'
commonName            :T61STRING:'oracle'
Certificate is to be certified until May 17 21:31:14 2003 GMT (365
  days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

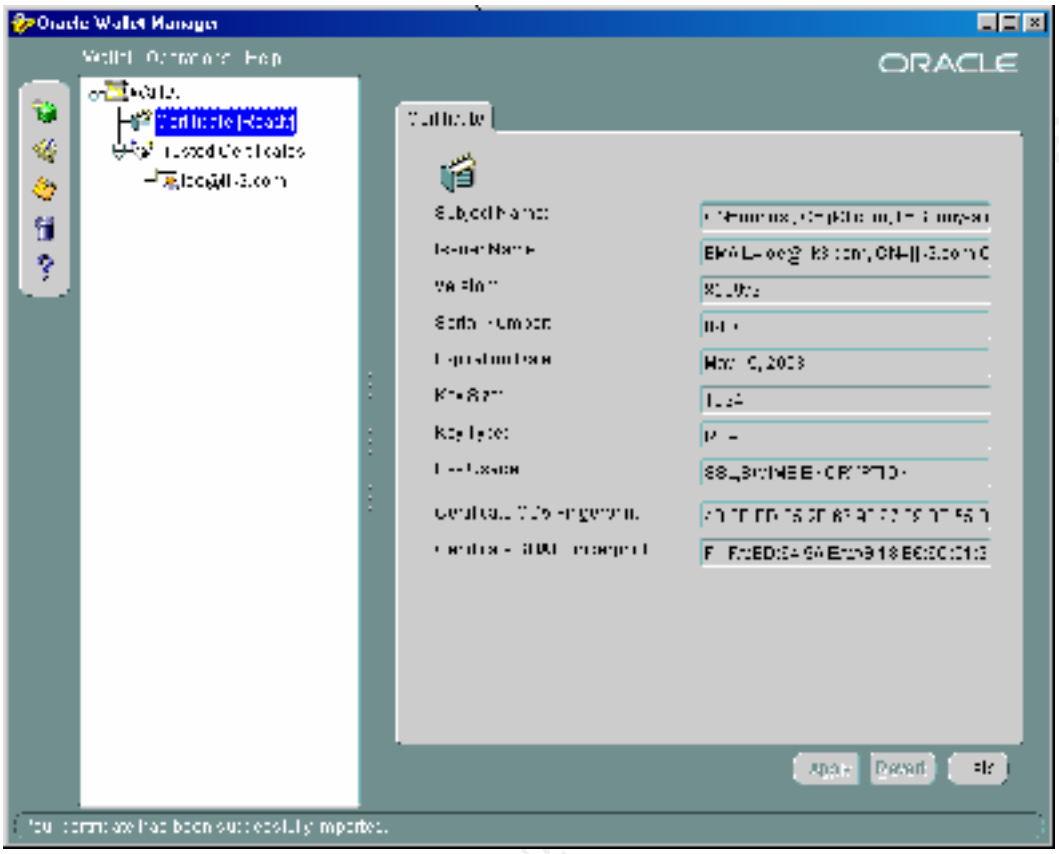
- The signed certificate is located at `/usr/local/ssl/misc/newcert.pem` and needs to be copied to the Oracle server.
- The CA's certificate, `/usr/local/ssl/misc/demoCA/cacert.pem`, also needs to be copied to the Oracle server.

4.3.3 Installing the Certificates into the Server's Oracle Wallet

- In the Oracle Wallet Manager, right click on Trusted Certificates and select "Import Trusted Certificate". Choose to "Select a file that contains the certificate" and then browse to the CA's certificate (`cacert.pem`).

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

- Next Right click on Certificate:[Requested], under the Wallet icon, and select "Import User Certificate". Choose to "Select a file that contains the certificate" and then browse to the CA's certificate (`newcert.pem`). Once the certificate has been imported the Certificate:[Requested] will change to Certificate:[Ready].
- The Oracle Wallet Manager should now look something like the image below



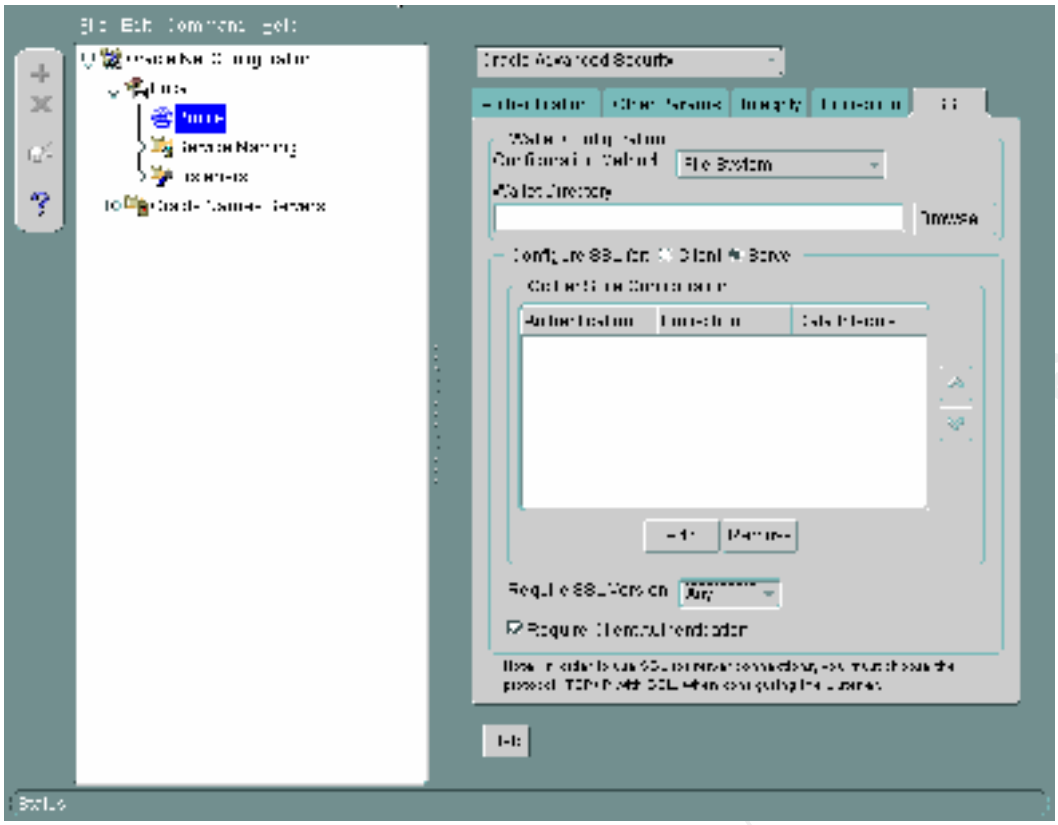
Oracle Wallet Manager after configuration

Now that both the client and the CA certificates have been installed, choose to save the Wallet, by clicking on the Wallet menu and selecting "Save In System Default" (`/etc/ORACLE/WALLETS/oracle`). After saving the wallet, close the Oracle Wallet Manager. **Note:** Saving the wallet in a location other then the system default causes problems later in the configuration.

4.3.4 Configure SSL with Oracle Net Manager

- As the "Oracle" user open the Oracle Net Manager located at $ORACLE_HOME/bin/netmgr
- Click on the drop down menu on the right hand side and select "Oracle Advance Security"
- Select the SSL Tab
- Select the "Server" Radio Button for the "Configure SSL for" parameter

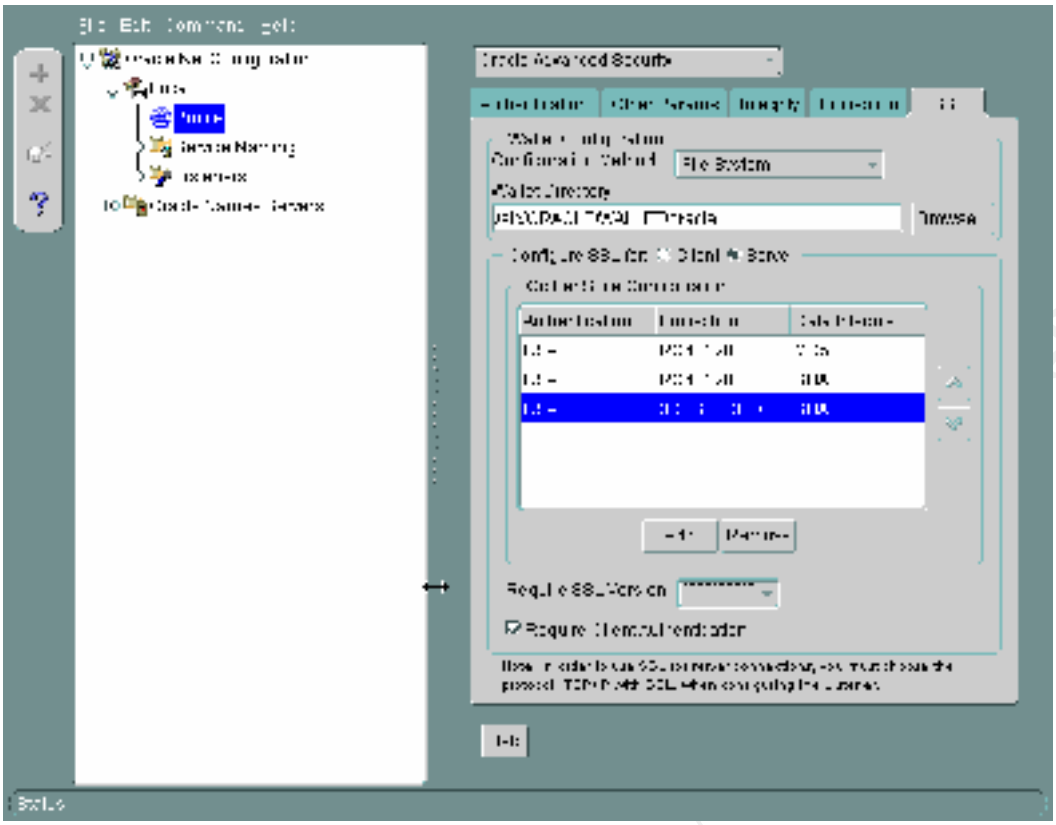
© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.



Oracle Net Manager Ready for Server Configuration

- Enter `/etc/ORACLE/WALLET/oracle` for the location of the Wallet Directory
  Click on the Add button under the "Cipher Suite Configuration". By default only exportable cipher suites are visible. Select the "Show US Domestic Cipher Suites" to show all supported cipher suites. Refer to Oracle Advanced Security Administrator's Guide, Release 9.0.1 for a complete list of cipher suites supported. [7]
- Select "RSA – 3DES_EDE_CBC – SHA" and then click OK
- Continue to add both "RSA – RC4_128 – MD5" and "RSA – RC4_128 – SHA"
  Note: Only ciphers with 128-bit encryption keys and RSA authentication are being added. There may be local restrictions on the key size for your specific installation. RSA authentication must be selected for the client to use SSL for authentication. If you do not want to use SSL for authentication the DH-anon (Diffie-Hellman Anonymous) can be selected.
- Select "3.0" for "Require SSL Version"
- Make sure the "Require Client Authentication" has been selected (it is selected by default). This setting will check to make sure the Oracle client has a valid certificate that has been signed by the trusted CA. If you do not plan to use digital certificate on the client side, de-select this option.
- Save the changes by clicking on the File menu and select "Save Network Configuration"

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.



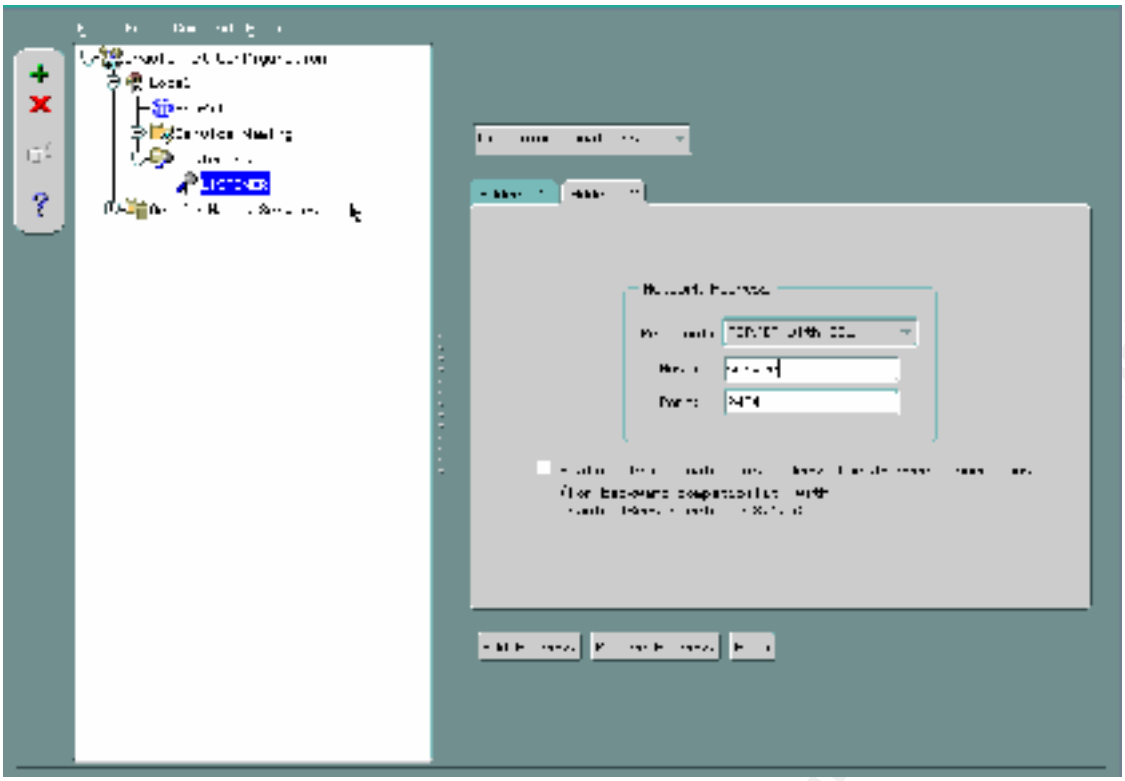
Oracle Net Manager for the server configured

4.3.5 Create a new Oracle listener

- In the Oracle Net Manager expand “Listeners” and select your listener (in this example the listener is named LISTENER).
- Click on the “Add Address” button located at the right hand bottom of the window.
- Select “TCP/IP with SSL” for the Protocol
- Enter the host name of the server for Host
- Enter “2484” for the Port. TCP port 2484 is the default Oracle port for SSL communications.

© SANS Institute 2000 - 2002

Author retains full rights.



Listener with SSL

- Save the network configuration by clicking on the file menu and selecting "Save Network Configuration"
- Restart the listener as the "oracle" user by issuing the commands
  - $ORACLE_HOME/bin/lsnrctl stop
  - $ORACLE_HOME/bin/lsnrctl start

4.4 Configure Client Connection

4.4.1 Create a Net Service Name

- Open Oracle's Net Configuration Assistant by selecting Start → Programs → Oracle - *<Oracle_Home_Name>* → Configuration and Migration tools → Net Configuration Assistant.
- Select "Local Net Service Name configuration" and then click Next
- Select to "Add" and then click Next
- Select "Oracle 8i or later database or server" and then click Next
- Enter the Service Name of the database and then click Next. This service name must match the global database name of the server for SSL to work.
- Select "TCPS" for the network protocol and then click Next.
- Enter the host name of the server and choose the default port, then click Next.
- Choose "no, do not test" and then click Next. (The test would fail since the client needs additional configuration).
- Enter the net service name for this connection and then click Next. The net service name can be anything and is what is used to identify this connection.
- Select "No" when asked to configure another net service name and click Next.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

4.4.2 Edit the Client's tnsnames.ora File

- Open the clients $ORACLE_HOME\network\admin\tnsnames.ora file with an editor, like notepad.
- Add the following line to the entry for the database connection

```
(security=(SSL_SERVER_DN="DN of the server's certificate"))
```

- After the tnsnames.ora file has been edited it should look like the example below

```
ORACLE =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = server)(PORT = 2484))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = oracle)
    )
    (security=(SSL_SERVER_DN="CN=oracle, O=jjk3.com, L=Sunnyvale,
ST=Ca, C=US"))
  )
```

- Save the tnsnames.ora file

4.5 Connect to the Database Server

- From the client you should be able to use `SQL Plus` to connect to the Oracle database via an SSL protected session.

5 Verification and Troubleshooting

Now that you have configured the Oracle client and server to communicate over SSL, you can ensure that the session is protected by SSL and that things are working correctly by enabling a "Trace" on the client. The Trace will dump all the session details to a log file which should indicate whether the SSL connection is working correctly.

To enable the Trace open up the Oracle Net Manager

- Expand Local and click on Profile
- Click on the drop down menu on the right hand side and select "General"
- The selected tab should be "Tracing" and select "Admin" for the client trace level.
- Enter the information for the location and name of the trace files
- Save the configuration and then make a `SQL Plus` connection

Once you have made a connection to the database server you should be able to view the trace log and verify that the connection has completed successfully. Valuable information in the trace file can be found on or below the following lines.

© SANS Institute 2000 - 2002, Author retains full rights.

"PARAMETER TABLE HAS THE FOLLOWING CONTENTS"
Below this line is a dump of many of the relevant settings related to the connection. This is a good place to verify most of the settings.

"found value for "ssl_cipher_suites""
This line should exist for each cipher suite supported. Below this line is the name of the cipher suite which can be used for the connection.

"The Final Negotiated SSL Cipher Suite is:"
The line will log which cipher suite will be used for the SSL connection.

In addition to the lines cited above, the trace file provides valuable information needed to troubleshoot a connection. A Trace can also be enabled on the server via the Net Manager.

If your connections to the Oracle server are failing, you will receive an error message in `SQL Plus`. This message is not always the correct error message. The trace file will have the exact error number for the problem encountered. Once you discover the error number for the error message, enter it into Oracle's error search page to get help on resolving the error.
(http://technet.oracle.com/docs/products/oracle9i/doc_library/901_doc/nav/error_search.htm)

6 Oracle SSL and Firewalls

When an Oracle client connects to a database via an SSL protected session, the client first contacts the server on TCP port 2484 and completes an SSL handshake. Once the SSL handshake has been completed, the server then redirects the client to a random high port on the server. The client then connects to the high port, performs another SSL handshake and then starts the interactive user session.

The redirection to a random high port can be a problem if there is a firewall or other filtering device in-between the client and the server. Oracle has two ways to resolve this problem.

Oracle's Advanced Security Administrator's Guide states that an Oracle Net Firewall Proxy kit is available for both proxy and stateful based firewalls. (http://download-west.oracle.com/otndoc/oracle9i/901_doc/network.901/a90150/asossl.htm#1009042) The Proxy Kit allows the firewall to participate in the SSL handshake (requiring access to the server's key) and to determine what ports need to be open for the SSL connection to be successful. I was unable to find any information about the Oracle Net Firewall Proxy kit on Oracle's site or the Internet.

Another Option is to use Oracle's Connection Manager. The Connection Manager is located behind the firewall and is able to support SSL connections on a single port. The Connection Manager can then route the Net8 connections to the correct database. SSL can not be used between the Connection Manager and the Oracle database and native Oracle encryption should be used to protect the session. For more information on the

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Oracle Connection Manager see the Oracle9i Net Services Administrator's Guide. (http://technet.oracle.com/docs/products/oracle9i/doc_library/901_doc/network.901/a90154/toc.htm)

7 References

1. Internet Security Systems, "Securing Database Servers", <http://documents.iss.net/whitepapers/securedbs.pdf> (5/10/2002)

2. CERT Coordination Center, "Configure computers for secure remote administration", 5/2/2001, <http://www.cert.org/security-improvement/practices/p073.html> (5/10/2002)

3. Oracle Technology Network, "Oracle Advanced Security Release 9i", <http://technet.oracle.com/deploy/security/aso/pdf/ASODataSheet.html> (5/10/2002)

4. Oracle Technology Network, "Oracle Database Security for Ebusiness", 6/2001, <http://technet.oracle.com/deploy/security/oracle9i/pdf/9isecbpa.pdf> (5/14/2002)

5. Alan O. Freier & Philip Karlton & Paul C. Kocher, "The SSL Protocol, Version 3.0", 3/1996, < http://www.netscape.com/eng/ssl3/ssl-toc.html> (5/14/2002)

6. Oracle Technology Network, "Database Security in Oracle8i, An Oracle Technical White Paper", 11/1999, <http://technet.oracle.com/deploy/security/pdf/oow99/dbswp86.pdf> (5/14/2002)

7. Oracle Technology Network, "Oracle Advanced Security Administrator's Guide, Release 9.0.1", <http://download-west.oracle.com/otndoc/oracle9i/901_doc/network.901/a90150/toc.htm >, (5/15/2002)

8. Scott McDermott, "Creating a Certificate Authority and Certificates with OpenSSL", <http://www.octaldream.com/~scottm/talks/ssl/opensslca.html> (5/15/2002)

9. OpenSSL Documents, "CA.pl (1)", <http://www.openssl.org/docs/apps/CA.pl.html> (5/15/2002)

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.