Lab Guide

How to Secure Oracle WebLogic 12c

Version 1.0

Christopher Parent, Instructor

Table of Contents

Lab Guide Overview

This lab guide will teach you how to perform common WebLogic administrative tasks using a variety of methods and tools. Each lab builds upon the previous, so it is highly recommended you perform all the labs in sequence.

The References section of this document contains links to the software needed for this lab, as well as links to online documentation for reference.

Organization

Each lab is organized in the following structure:

- **Skills Learned** describes what you will get out of the lab
- **Overview** describes the overall details of the lab
- **Configuration Parameters** defines parameters and values you will need to perform the lab
- **Instructions** provide the details steps needed to perform the lab

Formatting Convention

The lab uses various text formatting to convey meaning and action:

- **Bold** font will be used to emphasize actions to be taken:
  - **Click** the button
  - **Open** a terminal window
  - **Navigate** to **Environment > Servers**
- Courier font will be used to emphasize command syntax and command line execution
  - `cd <LAB_HOME>/domains/demo_domain`

File and Path Naming Convention

Oracle WebLogic is supported on Linux, Windows, and even OS X to a degree. This lab will work on all these systems. WebLogic for Windows includes .cmd or .bat versions of any scripts, whereas OSX and Linux include bash shell versions.

For the purposes of this lab, filenames and file paths will be referenced UNIX style in this document. If there are Windows-specific instructions for performing a particular task, they will be called out separately.

Lab Guide Directory Structure

This lab guide defines a standard set of directories for supporting the installation of WebLogic and the creation of WebLogic domains. It is highly recommended to use the directories outlined in Table 1 Lab Directories as the lab will reference these directories using its reference name and not the absolute path. You will see directory references throughout the lab in the following form:

*Navigate to **<DOMAIN_HOME>**/bin and execute startWebLogic.sh*

Wherever you see this variable notation, either for navigating a file system or specifying a file or directory path as part of some WebLogic resource configuration, be sure to use the fully qualified path and not the variable notation.

Table 1 Lab Directories

Directory Reference	Path
LAB_HOME	/u01/oracle or c:\oracle
ORACLE_ROOT	<LAB_HOME>
ORACLE_HOME	<LAB_HOME>/Middleware/Oracle_Home
JAVA_HOME	Where JDK is installed. Varies by system.
DOMAIN_ROOT	<LAB_HOME>/domains
DOMAIN_HOME	<DOMAIN_ROOT>/<domain_name>
APPLICATION_ROOT	<LAB_HOME>/applications
DERBY_HOME	<ORACLE_HOME>/wlserver/common/derby

Hardware Requirements

This lab requires a computer with sufficient memory and storage for installing WebLogic and running 2 or more WebLogic server instances. Suggested minimum requirements:

- 4GB ram, 6GB recommended
- 3GB disk space

Software Requirements

The following software is required:

- Oracle WebLogic Server 12.1.3. Generic Installer
- JDK 1.7.x or JDK 1.8.x
- Any text editor (notepad, vi, TextMate, etc)

Links to required software can be found in the References section of this lab guide.

Need Help?

If you need help with the labs or have questions please visit www.learnweblogiconline.com and contact me directly.

Lab #1: **Secure Installation**

Duration 45 minutes

Skills Learned

At the end of this exercise, you will be able to:

- Create dedicated OS user/group
- Avoid installing development components
- Identify and remove development components
- View, apply, and rollback patches

Overview

In this lab you will install Oracle WebLogic 12c using the Generic Installer using a dedicated user and group. After installation you will then identify and remove any development components that should not be used in a production environment.

Once the installation is complete, this lab will show you how to use OPatch to view current patch levels and apply the latest PSU.

Prerequisites

You need to have the following steps completed before using this lab guide:

- Oracle JDK 7 or JDK 8 installed
- Oracle WebLogic 12c (12.1.3) Generic Installer downloaded

You also need to have an existing My Oracle Support account that allows you to access and download PSUs for Oracle WebLogic Server.

If you do not have an account, you will not be able to patch WebLogic in this lab. However, you can at least view the steps involved with the process.

Configuration Parameters

Table 2 LAB_HOME Directory

Directory	OS
/u01/oracle	Linux, Unix, Mac
C:\oracle	Windows

Table 3 Generic Installer Parameters

Parameter	Value
Inventory Directory	<LAB_HOME>/oraInventory
Operating System Group	oracle
Oracle_Home	<ORACLE_ROOT>/Middleware/Oracle_Home

Table 4 Quickstart Configuration Wizard

Parameter	Value
Name	weblogic
Password	weblogic123
Domain Root Location	<LAB_HOME>/domains
Application Root Location	<LAB_HOME>/applications

Instructions

1	Preparing your Environment
1.1	Log into a terminal window
1.2	Elevate privileges to root
1.3	Create a new OS group named 'oracle' `[root@veyron ~]# groupadd oracle`
1.4	Create a new user named weblogic and add them to the oracle group `[root@veyron ~]# useradd -m -g oracle weblogic`
1.5	Set a password for the weblogic user `[root@veyron ~]# passwd weblogic`
1.6	Create your <LAB_HOME> directory for this Lab Guide according to Table 2 `[root@veyron /]# mkdir -p /u01/oracle`

1.7	Change permissions on LAB_HOME so that weblogic:oracle owns the directory and has rwx permissions
	
	`[root@veyron /]# chown -R weblogic:oracle /u01/oracle`
	
	`[root@veyron /]# chmod g+w /u01/oracle`
	
	Keep in mind that the weblogic user or oracle group will need rx permission on the root level directory – in this case, /u01. If weblogic or oracle does not have rx permission, they will not be able to navigate into /u01.

1.8	Verify that the user weblogic and group oracle have rwx permissions on <LAB_HOME> by listing the parent directory
1.9	Log into a new shell session as the weblogic user
2	**Installing Oracle WebLogic 12c using the Generic Installer**
2.1	Download **Oracle WebLogic 12c (12.1.3) Generic Installer** from oracle.com. See the Appendix for a URL.
2.2	After the generic installer has been downloaded, copy the installer to <LAB_HOME>

2.3	As the weblogic user run the installer by executing the following command:
	
	`<JAVA_HOME>/bin/java –jar`
	`<LAB_HOME>/fmw_12.1.3.0.0_wls.jar`
	
	The generic installer should now launch.

2.4	On the Installation Inventory Setup screen, specify oraInventory according to Table 3 Click **Ok**
2.5	The Welcome screen for Oracle Fusion Middleware will appear. Click **Next**.
2.6	On the Installation Location screen specify **Oracle Home** according to Table 3
2.7	On the Installation Type screen, **select** 'WebLogic Server' and click **Next**. ****Important** – if you select the option 'Complete with Examples,' you will install a variety of development tools and artifacts mentioned in the lecture that should not be left on production systems, such as demo certificates and sample domains. When you select either 'WebLogic Server' or 'Coherence', none of these components are installed.
2.8	On the Prerequisite Checks screen, verify that all checks pass. If any of the checks fail, view the logs (View Log button) and remediate as necessary. Click **Next**.
2.9	On the Security Updates screen, uncheck "I wish to receive security updates via My Oracle Support" and click **Next**. Click **Yes** on the confirmation dialog.
2.10	The Installation Summary screen summarizes the installation location, required disk space and the software components that will be installed. Back on the **Installation Summary** screen, click **Install**.
2.11	On the Installation Progress screen, click **Next**.

2.12	On the Installation Complete screen, **disable** 'Automatically Launch the Quickstart Configuration Wizard' and click **Finish**.
2.13	In the terminal window, view the contents of <LAB_HOME> and verify that the installation is owned by weblogic:oracle
2.14	Verify the following components are not installed by performing a simple case insensitive search under <ORACLE_HOME>: Derby (Derby database) demo.crt (Demo certificate) medrec (One of the many sample domains) `weblogic@veyron Oracle_Home]$ find . -print | grep -i demo`
3	**View Applied Patches using OPatch**
3.1	Run OPatch using the lsinventory flag to view the current patch level for WebLogic. `<ORCLE_HOME>/OPatch/opatch lsinventory` View the output of this command and note 2 things: 1. Location of Middleware Home as specified in the output 2. Patches applied – there should be no patches
3.2	Next **download** the latest PSU for the version of Oracle WebLogic Server you are using – in this lab we are using 12.1.3. Oracle maintains a web page on Critical Patch Updates and Security Alerts here: http://www.oracle.com/technetwork/topics/security/alerts-086861.html Use this link to navigate to the latest patches for a specific product. You will eventually have to log into My Oracle Support to download any patches. If you do not have a Support account, you will not be able to download and apply any product patches.
3.3	**Move** the downloaded patch into <LAB_HOME>
3.4	**Unzip** the downloaded patch. This will create a directory with the patch number.
3.5	**Run** opatch apply to apply the patch

	`<ORACLE_HOME>/OPatch/opatch apply <location_to_patch_directory>`
3.6	**Answer** 'Y' if the local system is ready for patching.
3.7	**Verify** the patch was successfully applied. Run opatch lsinventory as before to verify the patch has been inventoried. The Patch number should appear under Interim Patches.
4	**Rollback a Patch**
4.1	**Let's** assume that the patch process failed and the patch needs to be rolled back. **Use** the opatch rollback –id <patch_id> command to remove the patch.
4.2	**Verify** the patch was successfully rolled back.

Lab #2: **Domain Security**

Duration 60 minutes

Skills Learned

At the end of this exercise, you will be able to:

- Create a Production WebLogic domain using Configuration Wizard
- Create delegated admins
- Configure password policy and user lockout
- Audit security events
- Use Cross-Domain-Security

Overview

In this lab you will create two production Weblogic domains and walk through creating users, enabling password policies, and configuring auditing and CDS.

Configuration Parameters

Table 5 Domain A Configuration Wizard Parameters

Parameter	Value
Domain Location	<DOMAIN_ROOT>/domain_a
Domain username	weblogic
Domain password	weblogic123
Domain Mode	PRODUCTION
JDK	<JAVA_HOME>

Table 6 Domain B Configuration Wizard Parameters

Parameter	Value
Domain Location	<DOMAIN_ROOT>/domain_b
Domain username	weblogic
Domain password	weblogic123
Domain Mode	PRODUCTION
JDK	<JAVA_HOME>
Admin Server Listen Port	7011

Table 7 appManager User and Group Details

Parameter	Value

User name	appmanager
Provider	DefaultAuthenticator
Password	weblogic123
Group	Deployer

Instructions

1	**Create a WebLogic domain with Config Wizard**
1.1	Start the WebLogic Configuration Wizard by running the following script: `<ORACLE_HOME>/wlserver /common/bin/config.sh`
1.2	Select **Create a new domain**
1.3	Specify location of the domain then click **Next**. See Table 5.
1.4	Select the template **Basic WebLogic Server Domain** and click **Next**
1.5	Specify the domain **username** and **password** according to Table 5 and click **Next**
1.6	Specify domain mode and JDK according to Table 5
1.7	On the Advanced Configuration Screen, **disable** all checkboxes and click **Next**
1.8	On the Configuration Summary screen, click **Create**
1.9	On the Configuration Progress screen, click **Next** was you see the **Domain Created Successfully!** Message
1.10	On the Configuration Success page, make note of the Domain Location (known as DOMAIN_HOME) and the Admin Server URL. This will be used to log into the admin console.
1.11	Repeat the previous section to create another domain using the values from Table 6. It is important to define a unique listen port for the admin server. The default port is 7001. Refer to table 6 for specifying a unique port number.
2	**Create boot.properties**

2.1	Create a boot.properties for each WLS server in both domains under:
	
	<DOMAIN_HOME>/servers/<server_name>/security
	
	You may need to create the subdirectory <server_name>/security directory.
	
	Specify the following in the file using a text editor:
	
	`username=weblogic`
	`password=weblogic123`

2.2	Start the admin server for domain_a using the startWebLogic.sh script.
2.3	In a terminal window, verify the boot.properties file for the admin server has been encrypted.

3	**Change Control**
3.1	Log into the admin console using your browser.
3.2	In production mode you are required to obtain a lock in order to change domain settings. In the admin console notice the Change Center in the upper left. There is a button for obtaining a lock to make changes. Making a change within a domain involves the following steps, whether you are in the admin console or running a WLST script: 1. Log in to the admin server (console or WLST) 2. Start an edit session 3. Make any changes 4. Save those changes 5. Release the edit session
4	**Delegated Admins**
4.1	In this section you are going to create a new user and given them restricted administrative privileges. This new user will only be allowed to deploy applications to a domain. They will not have access to make any other domain changes.
4.2	**Log** into the admin console
4.3	**Navigate** to Security Realms > myrealm > Users and Groups
4.4	On the Users tab **create** a new user according to Table 7 appManager User and Group Details.
4.5	**Save** the user
4.6	Back on the Users tab **click** on the new user
4.7	**Click** on the Groups tab and **assign** the Deployers group. Make sure to click **Save**.
4.8	**Log out** of the console and log back in as the new user appmanager.
4.9	**Verify** this app manager only has access to deploy applications, but not make any domain changes or start and stop servers. **Navigate** to Environment > Servers > Control Observe how this new user does not have access to shutdown or startup any of the servers.
4.10	**Click** on the AdminServer and observe how all the attributes are not editable.
4.11	**Navigate** to Deployments and **observe** that the appmanager has access to install and deploy applications.
4.12	**Log** out of the admin console
5	**Define Password and Login Policies**
5.1	**Log** into the admin console as weblogic
5.2	**Navigate** to Security Realms > myrealm > User Lockout

	Observe the various user lock out attributes. By default, User Lock is enabled and allows for 5 login attempts.
5.3	**Navigate** to Security Realms > myrealm > Providers > Password Validation
5.4	**Click** on SystemPasswordValidator then **view** the attributes on the Provider Specific tab. WebLogic's SystemPasswordValidator allows you to define and implement a password enforcement policy.
5.5	**Click on Lock & Edit** in the Change Center
5.6	Under User Name Policies, **check** the first box for 'Reject if Password Contains the User Name'
5.7	**Save** the changes
5.8	**Activate** changes
5.9	**Navigate** to Security Realm > myrealm > Users and Groups
5.10	**Click** on appmanager and try to **change** its password to 'appmanager123' Observe the error message from WebLogic stating the password can not contain or equal to user name.
5.11	**Navigate** back to the SystemPasswordValidator and remove this constraint.
6	**Auditing Security Events**
6.1	In this section you will learn how to configure the default auditing provider to log security events.
6.2	**Navigate** to Security Realms > myrealm > Providers > Auditing
6.3	**Lock and Edit**
6.4	**Click** on New to create an Auditing Providers
6.5	**Enter '**SecAuditor' for name and **Click** OK
6.6	**Click** on SecAuditor then the Provider Specific tab
6.7	**Change** the Severity level to INFORMATION
6.8	**Save** changes
6.9	**Activate** changes
6.10	**Restart** the AdminServer
6.11	In a separate terminal window, navigate to <DOMAIN_HOME>/servers/AdminServer/logs and view the DefaultAuditRecorder.log file. Audit events will be logged to this log file for each WLS server.
6.12	Tail the log file
6.13	In a browser log back into the admin console and view the log entries from the tail. Observe audit events, specifically the Principal entries.

Lab #3: **Network Security**

Duration 60 minutes

Skills Learned

At the end of this exercise, you will be able to:

- View open ports
- Configure connection filters to secure ports
- Enable message limits

Overview

In this lab you will learn how to secure network access to WebLogic using a variety of techniques, to encrypt and filter network traffic.

Table 8 Managed Server 1 Details

Parameter	Value
Name	ms1
Listen Port	8001

Instructions

1	**Create a Manager Server**
1.1	In this section you will create a managed server to host an application and to demonstrate a typical production domain environment.
1.2	Start the admin server for domain_a
1.3	Log into the admin console and navigate to Environment > Servers
1.4	Create a managed server per Table 8 Managed Server 1 Details
1.5	In a terminal window, create a boot.properties for this server like you did in a previous lab for the Admin Server
1.6	Start ms1 using the following command: `<DOMAIN_HOME>/bin/startManagedWebLogic.sh ms1 t3://<admin_host>:7001`
1.7	
2	**View open ports (Unix/Linux systems only)**
2.1	In this section you will use tools netstat and nmap to view open ports in your environment. These tools are essential in understanding what ports are being exposed on a server.
2.2	Start the admin server and ms1 if they are not already started.

2.3	In a terminal window, switch to root or use sudo for the following steps.
2.4	Run nmap to scan for all open ports on your server. `[root@veyron ~]# nmap -sT -O localhost`

2.5	Observe the output from nmap. You should see the listen port defined for the AdminServer and ms1.
	
	Your nmap output will vary. The important item to observe are the WebLogic listen ports.
	
	`Starting Nmap 5.51 ( http://nmap.org ) at 2016-03-07 04:41 MST`
	`Nmap scan report for localhost (127.0.0.1)`
	`Host is up (0.000026s latency).`
	`Other addresses for localhost (not scanned): 127.0.0.1`
	`Not shown: 992 closed ports`
	`PORT STATE SERVICE`
	`22/tcp open ssh`
	`25/tcp open smtp`
	`111/tcp open rpcbind`
	`139/tcp open netbios-ssn`
	`445/tcp open microsoft-ds`
	`631/tcp open ipp`
	`7001/tcp open afs3-callback`
	`8001/tcp open vcom-tunnel`

2.6	Use netstat to list active connnections associated with ports and processes `[root@veyron ~]# netstat -tlnp`
2.7	**Observe** the output from nestat and note the Local Address column and the PID/Program columns. This command allows you to associate a PID with a listen address and port. **Locate** 7001 and 8001 and **note** the PIDs. Notice how the AdminServer port is listed several times – this is because Weblogic, by default, establishes several network channels or listen addresses.
2.8	**Using** the netstat command you inspect what ports are open on your system and verify their function. **Notice** port 1527 open on localhost, which is associated with a java process. Note the PID for port 1527. Example: `tcp 0 0 ::ffff:127.0.0.1:`**`1527`** `:::*` `LISTEN` **`19675/java`**

	Run ps to obtain a process listing to confirm what application is running on port 1527. `[root@veyron ~]# ps -ef | grep <PID from netstat>`
3	**Filter Connections using the WebLogic Connection Filter**
3.1	In this section you will create a Network Connection Filter to restrict access to the admin server from your computer. Connection filter allows you to create a software-based firewall/ACL in essence.
3.2	**Log** into the admin console
3.3	**Navigate** to Domain > Security > Filter
3.4	**Obtain** a Lock and Edit session
3.5	**Specify** the name of the Java class that implements a connection filter. For this lab, use the Java class that comes with Weblogic: `weblogic.security.net.ConnectionFilterImpl`

3.6	**Specify** the following rules for the connection filter. Substitute the name of your server for <server_hostname>
	
	`127.0.0.1 * 7001 allow #local ipv4`
	`<server_hostname> * 7001 allow #local hostname`
	`0:0:0:0:0:0:0:1 * 7001 allow #local ipv6`
	`0.0.0.0/0 * 7001 deny # all other traffic`

3.7	**Save and activate** changes.
3.8	Restart AdminServer and ms1 for the changes to take place.
3.9	**Log into** the admin console
3.10	If you are running WebLogic on a different machine than your browser, you will be denied access since we did not define a filter rule to allow your workstation to connect.
3.11	Shutdown the AdminServer by killing the process
3.12	Edit the config.xml and add a rule to allow your workstation to connect. **Locate** the <connection-filter-rule> XML tag

3.13	Insert a new rule before the last deny all rule. Be sure to substitute [your_IP_addr] with the IP address of your computer. `<connection-filter-rule>`**`[your_IP_addr]`** `* 7001 allow #my workstation</connection-filter-rule>`
3.14	Save the config.xml file
3.15	Restart the AdminServer
3.16	Verify you can log into the admin console
3.17	**IMPORTANT**! Remove the connection filter configuration via the admin console once you have finished this section.
4	**Preventing DoS Attacks**
4.1	DoS attacks can be thwarted in a couple of ways at the WebLogic level. 1. Restricting the size of messages 2. Set complete message timeouts
4.2	**Log** into the admin console
4.3	**Navigate** to Environment > Servers > AdminServer > Protocols > General
4.4	**Observe** the two parameters for setting Complete Message Timeout and Maximum Message Size. Note their default values.

Lab #4: **Encrypting Traffic**

Duration 60 minutes

Skills Learned

At the end of this exercise, you will be able to:

- Generate Self-Signed certificates
- Configure Identity and Trust
- Configure SSL for WebLogic and NodeManager
- Enable TLS and Strong Ciphers

Overview

In this lab you will configure WebLogic to use SSL to encrypt all network traffic by configuring Identity and Trust.

Keep in mind that self-signed certificates are not suitable for systems exposed to the internet. In some organizations, self-signed certificates are used for secure internal administrative traffic only.

Publicly exposed services should be secured using a trusted CA-signed certificate.

Configuration Parameters

Table 9 Managed server parameters

Parameter	Value
Server Name	ms1
Server Listen Address	<Leave blank>
Server Listen Port	8001
Should this server belong to a cluster	No, this is a stand-alone server

Table 10 Machine parameters

Parameter	Value
Machine name	machine1
NodeManager Listen Address	Fully qualified domain name
NodeManager Listen Port	5556

Table 11 NodeManager.properties parameters

Parameter	Value
ListenAddress	Fully qualified domain name

ListenPort	5556
SecureListener	true
CustomIdentityAlias	From Lab 4.1
CustomIdentityKeyStoreFileName	From Lab 4.1
CustomIdentityKeyStorePassPhrase	changeit
CustomIdentityKeyStoreType	JKS
CustomIdentityPrivateKeyPassPhrase	changeit
KeyStores	CustomIdentityAndCustomTrust

Instructions

1	**Creating Identity and Identity Keystore using keytool**
1.1	In this section you will create a self-signed certificate representing the identity of a Weblogic host using keytool.
1.2	In a terminal window as weblogic, navigate to <LAB_HOME>
1.3	Create a new directory: certs under <LAB_HOME>
1.4	Change to the certs directory

1.5	Use the keytool command to create a strong identity certificate and identity keystore. The following command should be entered on a single line.
	
	`keytool -genkey`
	`-alias` `` `hostname -f` ``
	`-keyalg RSA`
	`-keysize 2048`
	`-sigalg SHA256withRSA`
	`-validity 1095`
	`-keypass changeit`
	`-storetype jks`
	`-keystore` `` `hostname -f`_identity.jks ``
	`-storepass changeit`
	`-dname "CN=` `` `hostname -f` `"`
	`-ext BasicConstraints:critical=ca:true,pathlen:0`

1.6	**Verify** the identity keystore was created properly. Use keytool –list to list entries in the keystore. `keytool -list -keystore <keystore_filename> -storepass changeit -storetype jks`
2	**Create the Trust Keystore**
2.1	In this section you will create the trust keystore that WebLogic will use to trust a certificate. Since we are using self-signed certificates for this lab, trust is established by

	placing a copy of the identity certificate in the trust keystore. If we were using a CA-signed certificate, trust is established by the CA itself, validating the signature of the signed identity certificate.

2.2	In order to configure trust for self-signed certificates, **export** the identity certificate created in the previous section using the keytool –export command.
	
	`keytool -export`
	`-alias `hostname -f``
	`-file `hostname -f`.cer`
	`-keystore `hostname -f`_identity.jks`
	`-storetype jks`
	`-storepass changeit`

2.3	Create the trust keystore by importing the identity certificate into the trust store using keytool:
	
	`keytool -import`
	`-alias `hostname -f``
	`-file `hostname -f`.cer`
	`-keystore `hostname -f`_trust.jks`
	`-storetype jks`
	`-storepass changeit`
	`-noprompt`

2.4	Verify the trust keystore has also been created successfully.
3	**Configure Identity and Trust for WebLogic**
3.1	In this step you, you will configure identity and trust for each WebLogic server. In this lab it is assumed that the managed server and AdminServer are on the same host.
3.2	**Shut down** ms1 if it is running
3.3	**Log** into the admin console
3.4	**Navigate to** Servers > AdminServer > Keystores
3.5	**Enable** Lock and Edit
3.6	**Note** the keystores default to DemoIdentity and DemoTrust. These were physical removed from the file system from an earlier lab and no longer exist.
3.7	**Change** the Keystores type to Custom Identity and Custom Trust and **click** Save.
3.8	Under Custom Identity Keystore, enter the path to the identity keystore created above.
3.9	Under Custom Identity Keystore Type, enter JKS

3.10	Specify and confirm the identity keystore password
3.11	Repeate the above steps for specifying the trust keystore
3.12	**Save** all changes.
3.13	**Repeat** these steps to configure Identity and Trust keystores for ms1.
3.14	**Save and activate all changes.**
4	**Configure SSL**
4.1	In this section you will now configure WebLogic to use SSL.
4.2	In the admin console, **navigate** to Environment > Servers > AdminServer > SSL
4.3	**Enable** Lock and Edit
4.4	**Specify** the Private Key Alias used when creating the identity keystore from earlier. The syntax used earlier specified the hostname as the alias.
4.5	**Specify** and confirm the identity keystore passphrase
4.6	**Save changes**
4.7	**Repeat** all steps for ms1
4.8	**Navigate to** Environment > Servers > Admin Server
4.9	On the General tab, **enable** the SSL Listen Port checkbox and **specify** 7002 as the SSL Listen Port.
4.10	**Navigate to** Environment > Servers > ms1
4.11	**Configure** ms1 to use 8002 for SSL
4.12	**Save** and **activate** changes**.**
4.13	**View** the log output from the AdminServer and confirm that the AdminServer is now listening securely on 7002.
4.14	**Configure** the trustedCAKeyStore JVM flag **Edit** <DOMAIN_HOME>/bin/startWebLogic.sh **Find** the comment "# START WEBLOGIC" in the script and add the following JAVA_OPTION statement on the next line: `JAVA_OPTIONS="${JAVA_OPTIONS} -Dweblogic.security.SSL.trustedCAKeyStore=<location_of_trust_keystore>"` Save the changes
4.15	**Start** ms1 using a secure connection: `./startManagedWebLogic.sh ms1 t3s://<admin_server_host>:7002`
5	**Troubleshooting SSL**
5.1	In this section you will learn how to enable the SSL debug flags to troubleshoot SSL issues. Most SSL issues occurring during the handshake and are most commonly due to: 1. Malformed certificates 2. Bad certificate chains

	3. Invalid CA

5.2	To enable SSL debugging for all servers, **edit** the <DOMAIN_HOME>/bin/startWebLogic.sh script:
	
	Append the following JAVA_OPTIONS
	
	`-Djavax.net.debug=all`
	`-Dssl.debug=true`
	`-Dweblogic.StdoutDebugEnabled=true`

5.3	**Save** the file
5.4	**Restart** the admin server
5.5	**Restart** the managed server and observe the SSL debug statements as ms1 negotiates and SSL connection with the AdminServer. The SSL debug statements will detail every SSL handshake, including the keys that were used, ciphers negotiated, and any cert chain validation issues.
5.6	Be sure to **remove** the SSL debug JAVA_OPTIONS when finished.
6	**Configure SSL for NodeManager**
6.1	Create a Machine in the admin console per Table 10 Machine parameters
6.2	Assign ms1 to the machine
6.3	Update the properties defined in <DOMAIN_HOME>/nodemanager/nodemanager.properties per Table 11 NodeManager.properties parameters
6.4	Add the custom trust store created earlier to <DOMAIN_HOME>/bin/startNodeManager.sh **Find** the JAVA_OPTION that specifies the –Dweblogic.security.SSL.trustedCAKeyStore flag and set its value accordingly: `JAVA_OPTIONS="-Dweblogic.security.SSL.trustedCAKeyStore="<location_of_trust_keystore>"`
6.5	Shutdown ms1 if running
6.6	Start NodeManager
6.7	Start ms1 via the admin console
7	**Enable TLS Only and Strong Ciphers**
7.1	Shutdown NodeManager, Admin Server, and ms1
7.2	Force WebLogic to use TLS1.0 as minimum by specifying the following JVM system property: `-Dweblogic.security.SSL.minimumProtocolVersion=TLSv1`
7.3	To configure specific ciphers, edit the domain's config.xml and add the following XML element to each server's SSL definition: `<ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA256</ciphersuite>`

	*** A word of caution. Certain cipher suites are not supported with certain browsers. Specifying an unsupported cipher may prevent admin console access.
7.4	To configure specific ciphers for NodeManager, use the following JVM system property: `-Dweblogic.security.SSL.Ciphersuites=TLS_RSA_WITH_AES_256_CBC_SHA256`
7.5	Start admin server
7.6	Start NodeManager
7.7	Enable SSL debug for ms1
7.8	Start ms1 using NodeManager and verify TLS1.0 at a minimum is specified and the specified cipher suite are enabled in the ms1 server log.

Lab #5: **Administrative Security**

Duration 30 minutes

Skills Learned

At the end of this exercise, you will be able to:

- Configure WebLogic to use administration port
- Generate user config and keyfiles for authentication

Overview

In this lab you will learn how to enable the domain administration port and how to reconfigure WebLogic and NodeManager to use this administration port.

The second part of the lab will teach you how to avoid using plaintext passwords when interacting with WebLogic using WLST.

Configuration Parameters

Table 12 Data source properties

Parameter	Value
Name	derbyDS
JNDI Name	demo.derbyDS
Database Type	Derby
Database Name	demodb
Host Name	localhost
Port	1527
Database User Name	<blank>
Password	<blank>

Instructions

1	**Enable Administration Port**
1.1	**Remove** any specific ciphersuite modifications made during the previous lab.
1.2	**Restart** the admin server for domain_a under Domain Structure
1.3	**Obtain** a Lock and Edit session
1.4	**Check** the box to enable the administration port
1.5	**Click** Save
1.6	**Navigate** to Environment > Servers > AdminServer > General tab
1.7	On the General tab, **expand** the Advanced subsection
1.8	**Set** the Local Administration Port Override to 9001
1.9	**Repeat** the previous steps to **set** the Local Administration Port Override for

	ms1 to **9002**
1.10	**Save** changes
1.11	**Activate** changes
1.12	After you activate the changes you will be presented with an error message stating that Console/Management requests can only be made through an administration channel. This is confirmation that you have successfully enabled the administration port.
1.13	Verify you can log into the admin console using the admin port specified for the admin server
2	**Connect MS1 using Admin Port**
2.1	In another terminal window start ms1 using the startManagedWebLogic.sh script using the following syntax: `./startManagedWebLogic.sh ms1 t3s://<host>:7002` Verify that ms1 cannot connect to the admin server. A 403 Forbidden error should appear in the output for ms1.
2.2	**Restart** ms1 using the correct admin port and verify ms1 can connect successfully.
2.3	**Shutdown** ms1
2.4	**Start** NodeManager
2.5	**Start** ms1 via the Admin Console
2.6	**Observe** NodeManager did not require any configuration changes to support the administration port

References

Create an Oracle Account
https://login.oracle.com/mysso/signon.jsp

JDK7 Download Page
http://www.oracle.com/technetwork/java/javase/downloads/index.html

JDK7 Installation Guide
http://docs.oracle.com/javase/7/docs/webnotes/install/index.html

JCE 8 Extensions
http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

Oracle WebLogic Server 12c Downloads Page
http://www.oracle.com/technetwork/middleware/weblogic/downloads/wls-main-097127.html

Oracle WebLogic 12.1.3 Online Documentation
http://docs.oracle.com/middleware/1213/wls/index.html

WLST Command Reference
https://docs.oracle.com/middleware/1213/wls/WLSTC/intro.htm#WLSTC107