IBM® WebSphere® Application Server V7– LAB EXERCISE

WebSphere Application Server security auditing

What this exercise is about

The objective of this lab is to introduce some of the new security auditing features in WebSphere Application Server Network Deployment V7 edition on distributed platforms. This exercise is split into two main sections. The first half goes through the process of enabling security auditing, setting basic audit configurations, and viewing the audit reports. The second half, which is optional, goes through some slightly more advanced features of the auditing functionality, including encrypting and digitally signing the audit logs.

Lab requirements

The list of system and software required for the student to complete the lab.

- A system that meets that requirements for running WebSphere Application Server Version 7, with approximately 500 MB of disk space for creating profiles
- The most current version of WebSphere Application Server V7
- An application server profiles with administrative security enabled, and with the administrative console and the default application deployed.

What you should be able to do

At the end of this lab you should be able to:

- Enable security auditing
- Configure security auditing for different administrative users
- Generate and view security audit report
- Configure new event filters
- Configure digital signing for the audit logs
- Configure encryption settings for security auditing

Introduction

WebSphere Application Server Version 7 builds on improvements made in Version 6.1. A few of the major enhancements introduced in this release are the capabilities to:

Part 1: Create an audit User ID

Since it may be desirable to distinguish those console users that have administrative access from those that have auditing console, a separate administrator user is created and mapped to the Audit role. This user is then used to configure and enable auditing features.

Part 2: Configure and enable WebSphere security auditing

This portion of the exercise configures and enables the auditing service. Before actually enabling the auditing, you need to configure how notifications will take place. For this exercise, you configure auditing to report the events to a log file.

Part 3: View the audit logs

After enabling the auditing, you verify that events are being reported to the log file. You also generate an html report, which is more readable that the text based log files.

Part 4: (Optional) Create a new event filter

Security auditing reports only four types of events by default, but there are many additional events which can be configured as well. This section adds an additional event filter, and maps it to the configurations for the service provider and event factory.

Part 5: (Optional) Digitally sign the audit log entries

In order to ensure the integrity of the log entries, digital signing can be configured. Once signing is enabled, the log entries are also 64-bit encoded. This portion of the exercise enables digital signing for the audit logs.

Part 6: (Optional) Encrypt the audit logs

This part adds encryption on top of the digital signing. This requires the addition of a new keystore and certificate which will be specific to encrypting the audit logs. Once that keystore exists, the encryption is enabled and verified.

Part 7: (Optional) Verbose logging and reporting

The final section of the lab enables verbose audit logging. This provides some additional information in the log entries that were not available previously. You also produce a new “complete” html audit report.

Exercise instructions

Instructions and subsequent documentation use symbolic references to directories which are listed as follows:

Reference Variable	Windows Location	Linux UNIX Location
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	Linux /opt/WebSphere/AppServer AIX /usr/WebSphere/AppServer
<TEMP>	C:\temp	/tmp
<hostname>	Host name or host address for the machine where the profiles are being created	Host name or host address for the machine where the profiles are being created

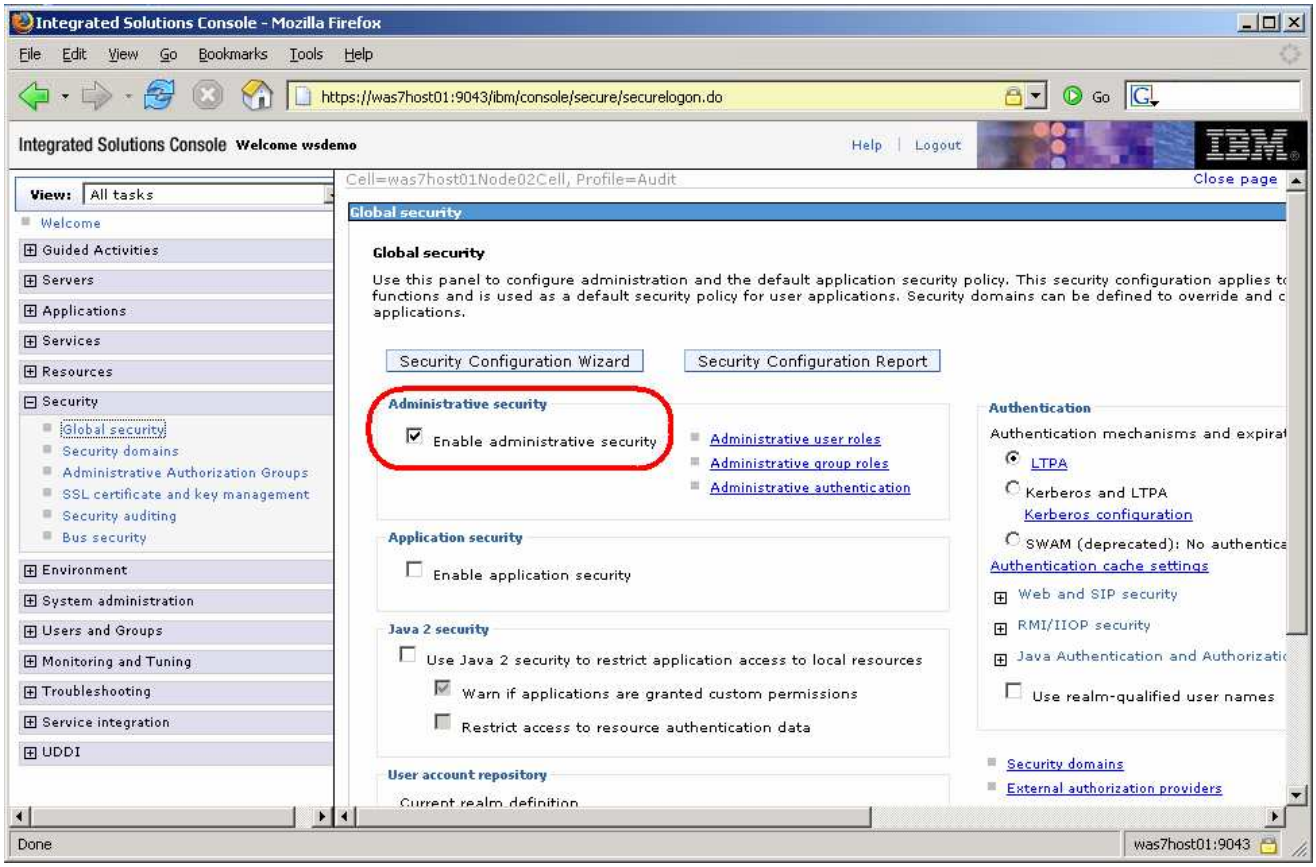
Part 1: Create an audit User ID

WebSphere Application Server has the ability to grant administrative users different roles to distinguish between the sorts of access they have within a cell or application server. With WebSphere Application Server version 7, a new role of Auditor has been added and is required to configure and enable any of the auditing features. By having a separate role for auditing, it is possible to distinguish between administrative users and those users you want to grant access to auditing functions.

This part of the lab creates a new administrative user called wsaudit and maps them to the auditor role.

____ 1. Start by ensuring that the application server is running.

____ 2. Open an administrative console and verify that administrative security is enabled.

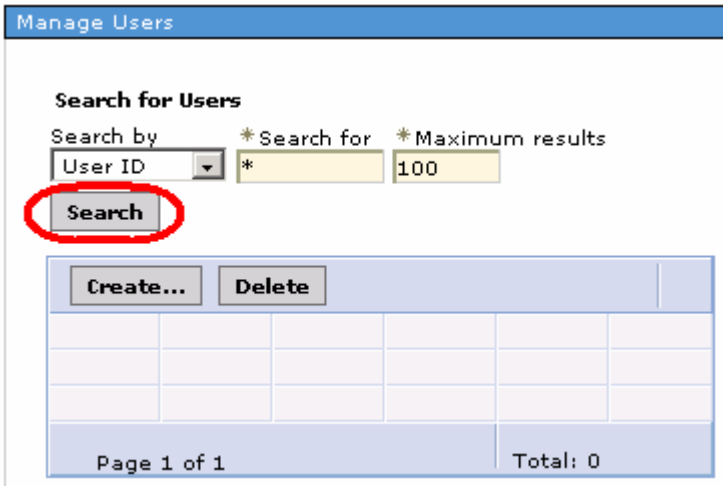


__ a. If administrative security is not enabled, enable it (using a file-based repository) and restart the server.

____ 3. For security reasons, it is not necessarily desirable to have your administrators be able to configure and control the audit settings. The primary security user has implicit rights to the audit functionality, but other administrators do not (unless they have explicitly had the Audit role granted to their user). This step goes through adding a new user named **wsaudit** and assigning it to the **Auditor**.

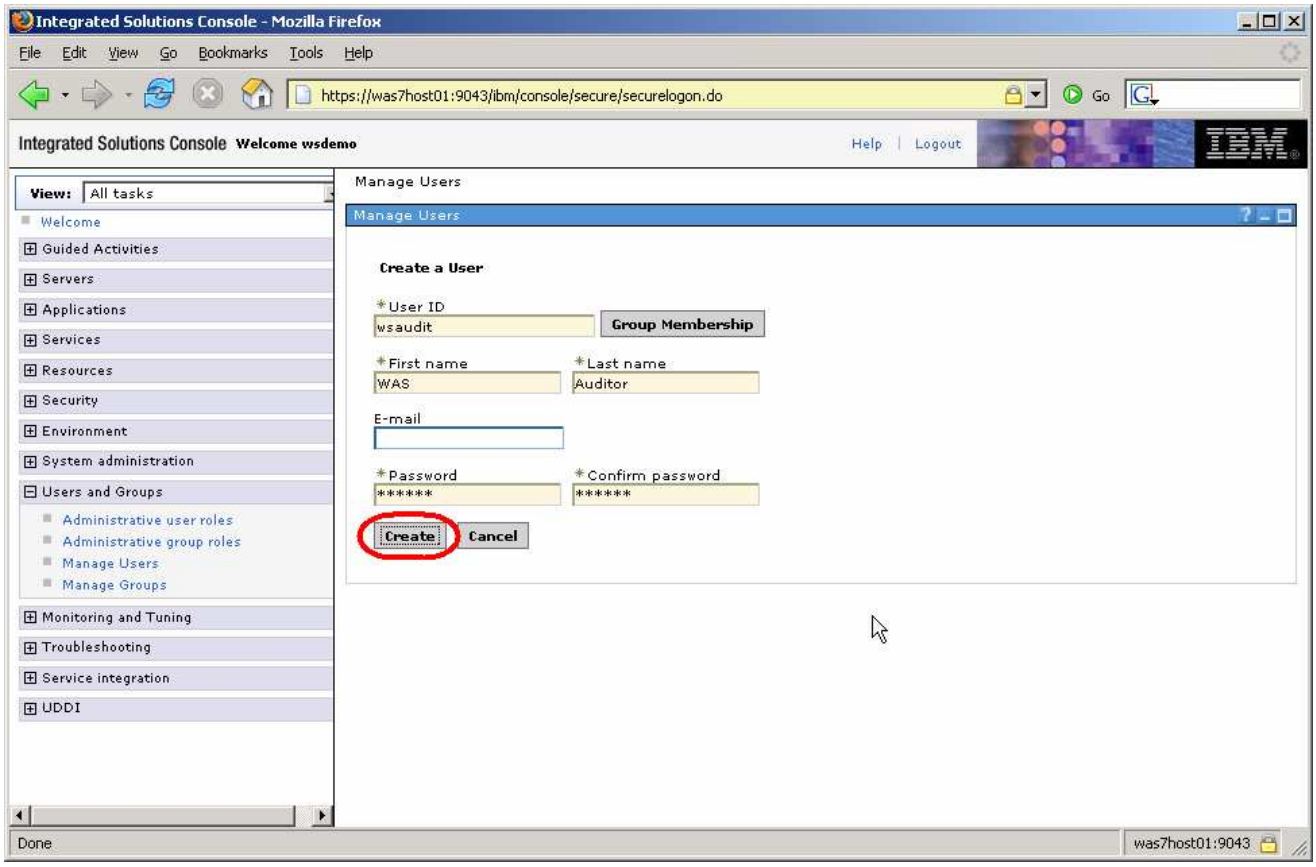
__ a. In the administrative console, under **Users and Groups**, click **Manage Users**.

__ b. Click **Search** to verify that **wsaudit** does not already exist.



__ c. Click **Create** to add the new user. On the next screen enter:

- **wsaudit** for the **User ID**
- **WAS** for the **First name**
- **Auditor** for the **Last name**
- **wsdemo** for the **Password** and confirmation password

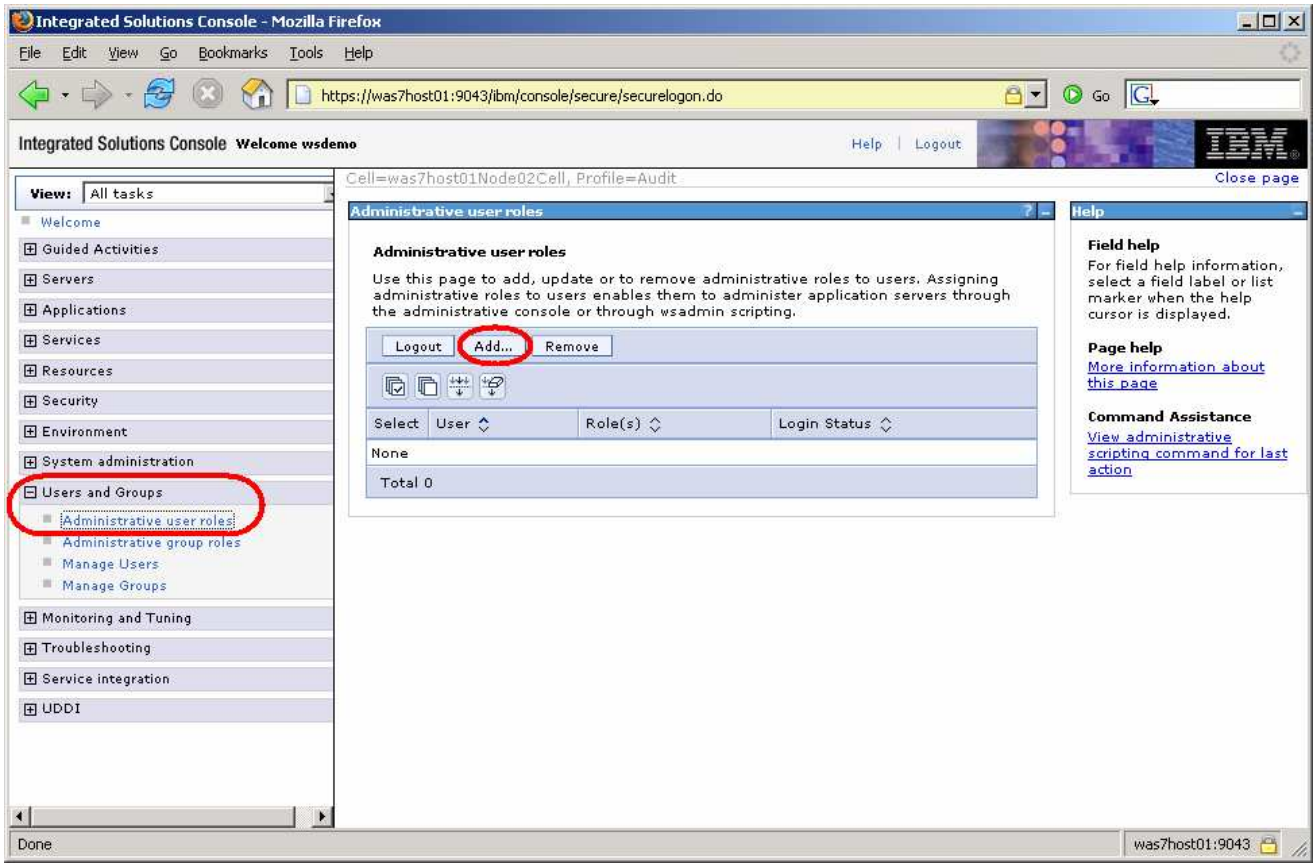


__ d. Click **Create** again and then **Close.**

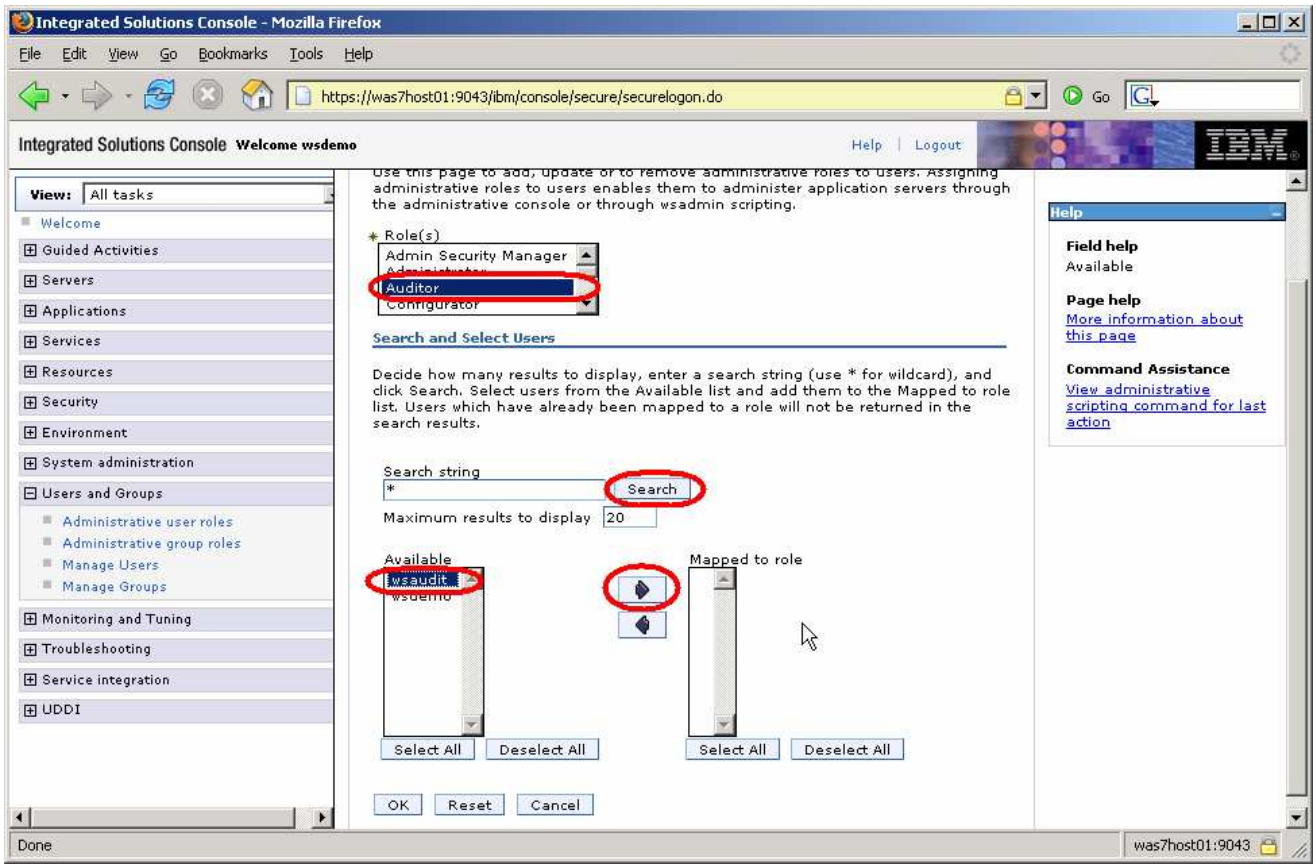
____ 4. Assign the **Auditor** role to **wsaudit.**

__ a. Using the administrative console, click **Administrative user roles** under **Users and Groups**.

__ b. Click **Add.**



__ c. Select the **Auditor** role under the **Roles** list. Then click the **Search** button to display the list of known users. From the list of users, select **wsaudit** in the **Available** box and click the **right arrow** to add them to the **Mapped to role**.



__ d. Click **OK** and **Save** the changes.

Part 2: Configure and enable security auditing

Now that an auditor user exists, this part of the exercise configures and enables WebSphere security auditing. Before auditing can be enabled, several configuration settings need to be set so that the audit service knows what to do with the audit events.

This initial part of the exercise turns on the basic auditing functions and sends the output to a log file.

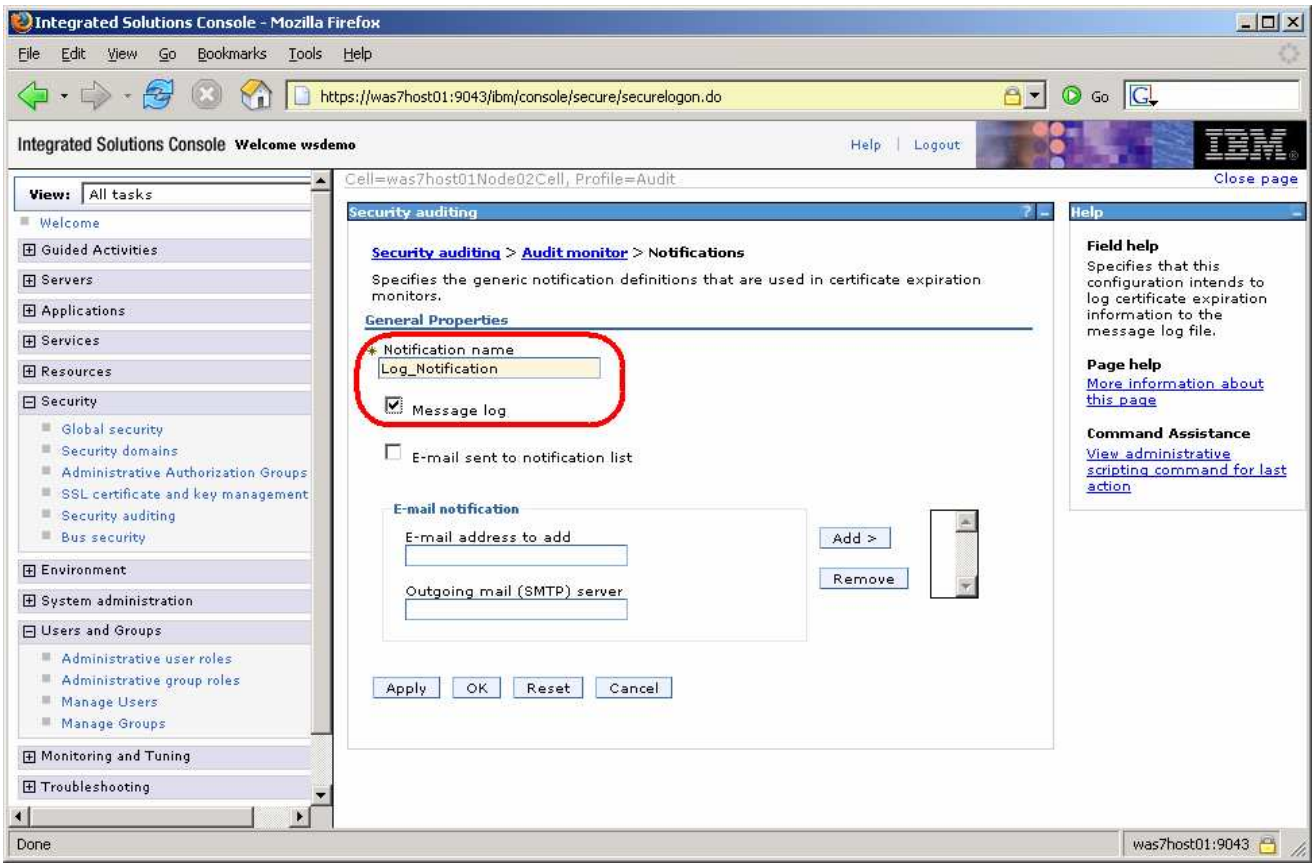
____ 1. Before enabling security auditing, there are some configuration setting that need to be set.

__ a. In the administrative console, click **Security auditing** under **Security**.

__ b. Before enabling the auditing, it is necessary to determine what happens with the audit records. Start by clicking **Audit monitor** under **Related Items**.

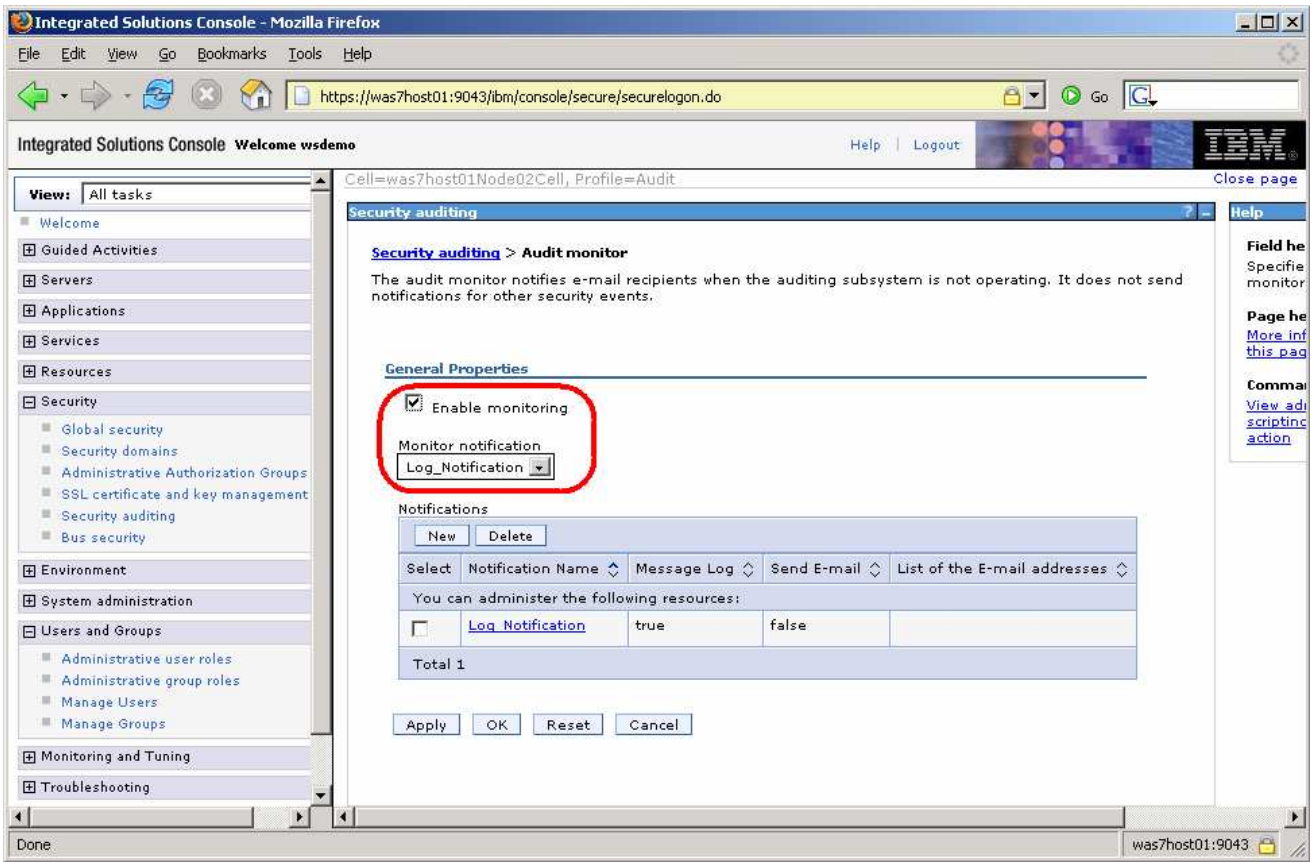
__ c. Under **Notifications**, click **New**.

__ d. This screen defined the notification specifics. Enter **Log_Notification** for the **Notification name** and check the **Message log** box. You can also configure e-mail notifications if needed.



__ e. Click **OK** and **Save** the changes.

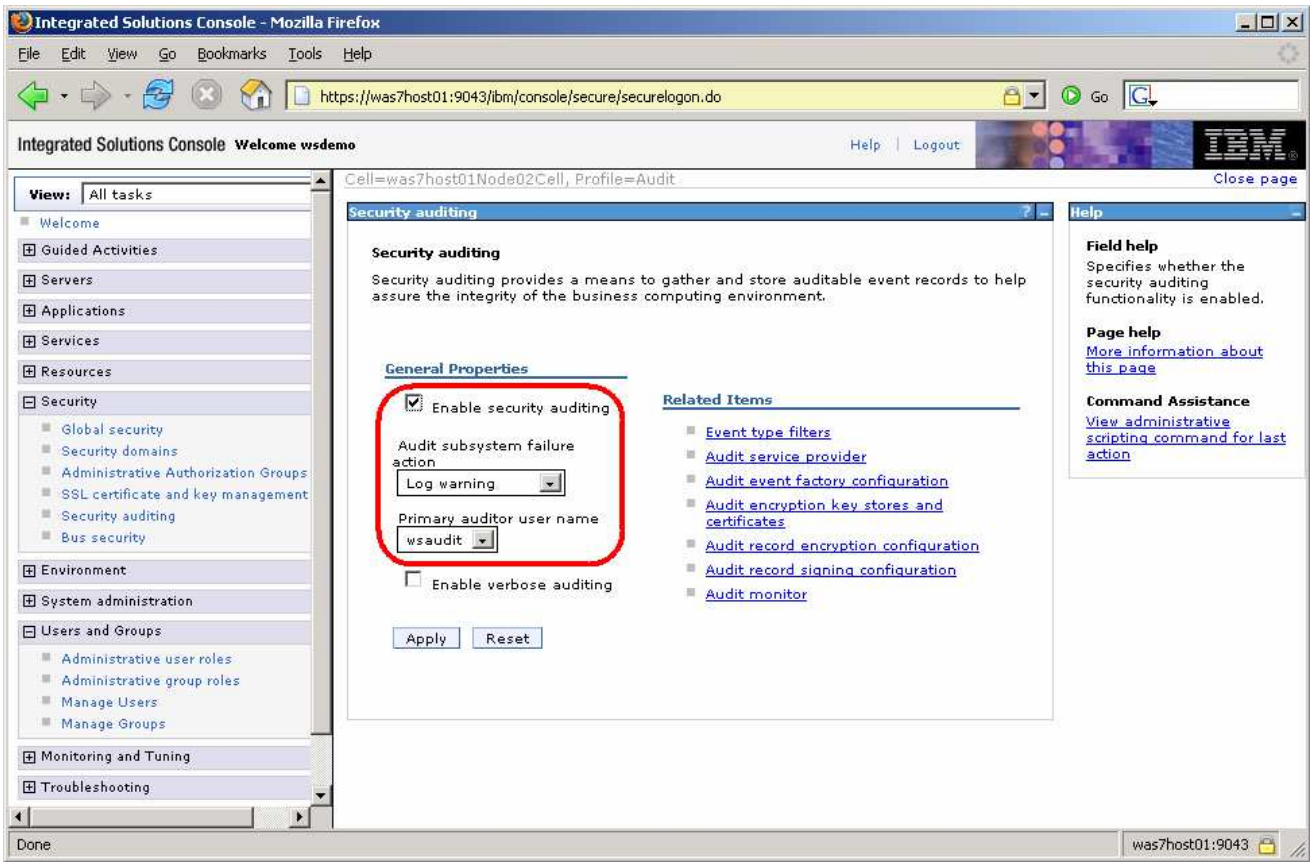
__ f. Now that a notification definition exists, it is possible to configure auditing to use that notification. On the same screen, check the **Enable monitoring** box and verify that **Log_Notification** has been selected in the **Monitor notification** pull-down list.



__ g. Click **OK** and **Save** the changes. This returns you to the main **Security auditing** page.

____ 2. Now that the configuration settings have been completed, it is possible to enable auditing.

__ a. At this point, check the **Enable security auditing** box. From the **Audit subsystem failure action** pull-down, select **Log warning**. And from the **Primary auditor user name**, select **wsaudit**.



NOTE: The Audit subsystem failure action dropdown menu has the following options:

No warning: The **No warning** action specifies that the auditor will not be notified of a failure in the audit subsystem. The product will continue processing but audit reporting will be disabled.

Log warning: The Log warning action specifies that the auditor will be notified of a failure in the audit subsystem. The product will continue processing but audit reporting will be disabled.

Terminate server: The Terminate server action specifies the application server to gracefully quiesce when an unrecoverable error occurs in the auditing subsystem. If e-mail notifications are configured, the auditor will be sent a notification that an error has occurred. If logging to the system log is configured, the notification of the failure will be logged to the system file.

__ b. Click **Apply** and **Save** the changes.

____ 3. **Restart the server** to have these security changes take effect.

__ a. In order for these changes to take effect, the server needs to be restarted. If this were running in a federated environment, the nodes would first be resynchronized, and then all processes in the cell would be restart.

__ b. For this exercise, **stop** the server and then **start** it again.

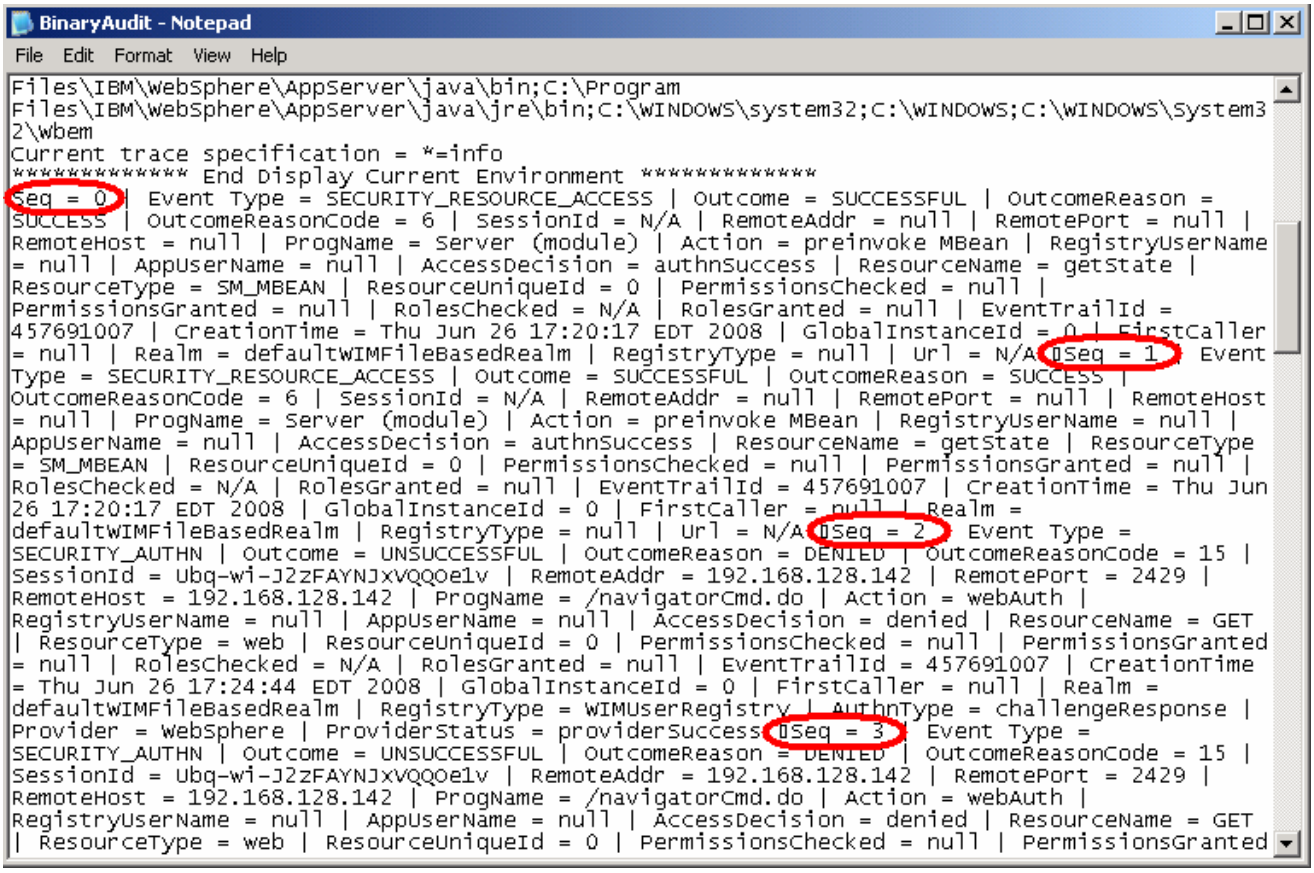
Part 3: View the audit logs

Security auditing is now enabled. This part of the exercise goes through the process of viewing the audit data.

The fastest way to view the data is to simply look at the log file that is generated, but that can be difficult to read. The other way to view the data is to use wsadmin to generate an html report. This part of the exercise goes through both of these options.

____ 1. View the log records with a text editor.

__ a. Using **Windows Explorer**, go to the logs directory for the server and open the file called **BinaryAudit_<cellName>_<nodeName>_server1.log** in a text editor.



__ b. Notice the sequence numbers. Those are the individual audit records, but this format certainly is not easy to read. If a better text editor is used, the output can be slightly more readable, but still not easy to read.

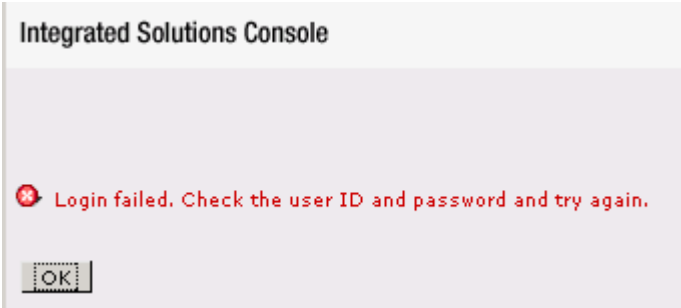
```
************ Start Display Current Environment ************
WebSphere Platform 7.0.0.0 [ND 7.0.0.0 h0823.03] running with process name was7host01Node02Cell\was7host01Nod
Detailed IFix information: No Interim Fixes applied to this build
Host Operating System is Windows XP, version 5.1 build 2600 Service Pack 3
Java version = J2RE 1.6.0 IBM J9 2.4 Windows XP x86-32 jvmwi3260-20080523_19691 (JIT enabled, AOT enabled)
J9VM - 20080523_019691_1HdSMr
JIT  - r9_20080522_1822
GC   - 20080521_AC, Java Compiler = j9jit24, Java VM name = IBM J9 VM
was.install.root = C:\Program Files\IBM\WebSphere\AppServer
user.install.root = C:\Program Files\IBM\WebSphere\AppServer\profiles\Audit
Java Home = C:\Program Files\IBM\WebSphere\AppServer\java\jre
ws.ext.dirs = C:\Program Files\IBM\WebSphere\AppServer/java/lib;C:\Program Files\IBM\WebSphere\AppServer\prof
Classpath = C:\Program Files\IBM\WebSphere\AppServer\profiles\Audit/properties;C:\Program Files\IBM\WebSphere
Java Library path = C:\Program Files\IBM\WebSphere\AppServer\java\jre\bin;.;C:\Program Files\IBM\WebSphere\Ap
Current trace specification = *=info
************* End Display Current Environment *************
Seq = 0 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeRea
Seq = 1 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeRea
Seq = 2 | Event Type = SECURITY_AUTHN | Outcome = UNSUCCESSFUL | OutcomeReason = DENIED | OutcomeReasonCode =
Seq = 3 | Event Type = SECURITY_AUTHN | Outcome = UNSUCCESSFUL | OutcomeReason = DENIED | OutcomeReasonCode =
Seq = 4 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = REDIRECT | OutcomeReasonCode =
Seq = 5 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeRea
Seq = 6 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeRea
Seq = 7 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 8 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 9 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 10 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 11 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 12 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
Seq = 13 | Event Type = SECURITY_AUTHN | Outcome = SUCCESSFUL | OutcomeReason = SUCCESS | OutcomeReasonCode =
```

__ c. It is also possible to use **tail –f** to track the entries added to the log file in real time.

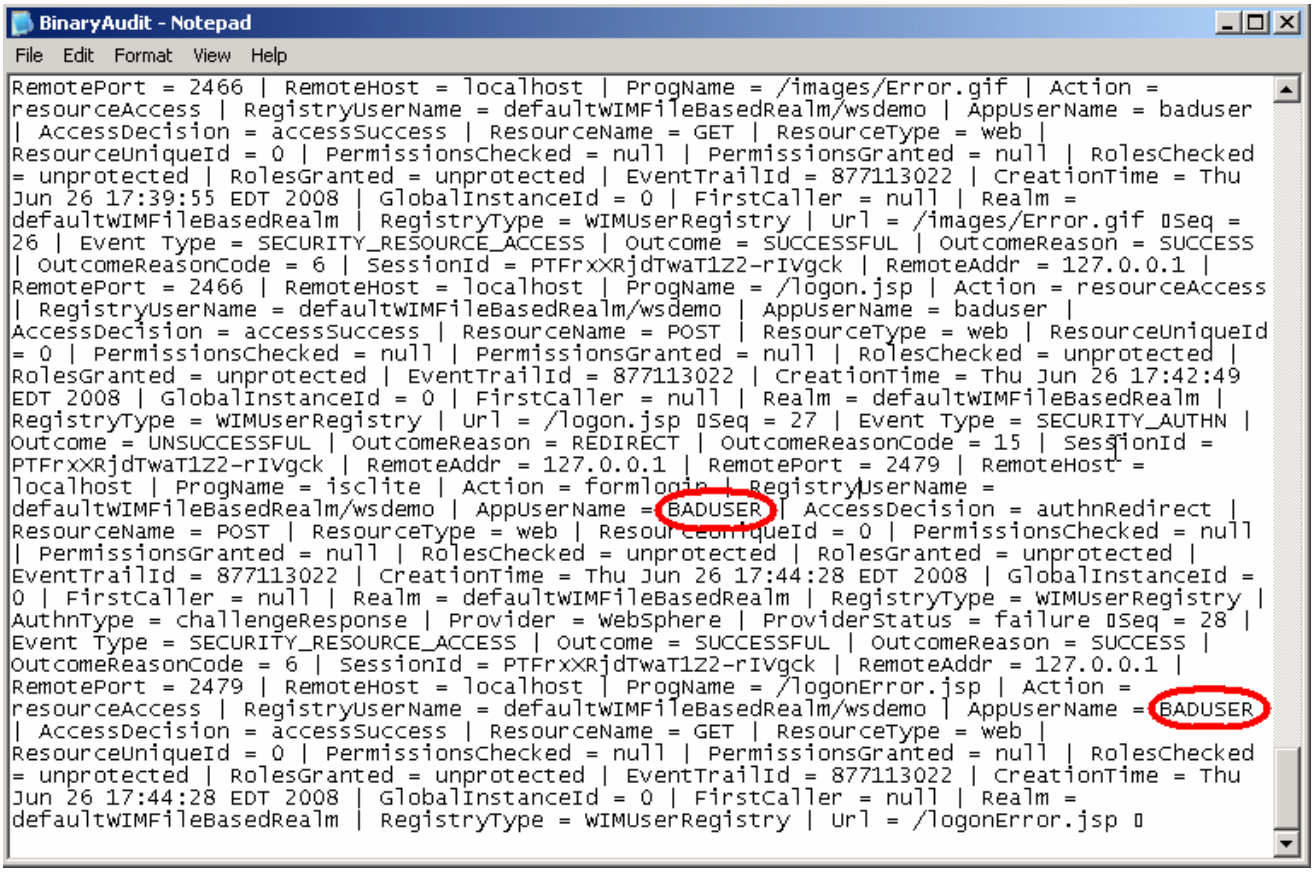
____ 2. Verify that auditing is actually logging events that need to be reported.

__ a. Open a new browser instance to the administrative console.

__ b. When prompted for a username and password, enter **BADUSER** and **wsdemo**



__ c. Reopen the **BinaryAudit_<cellName>_<nodeName>_server1.log** in a text editor and search for **BADUSER**. There will be several instances and it becomes clear that the login attempt failed.



____ 3. View the log entries using the Audit Log Reader. This is an interface available through wsadmin which will convert the audit log entries into an html report.

__ a. Using a command window, go to the bin directory for your profile. Enter the command:

```
wsadmin –lang jython –username wsaudit –password wsdemo
```

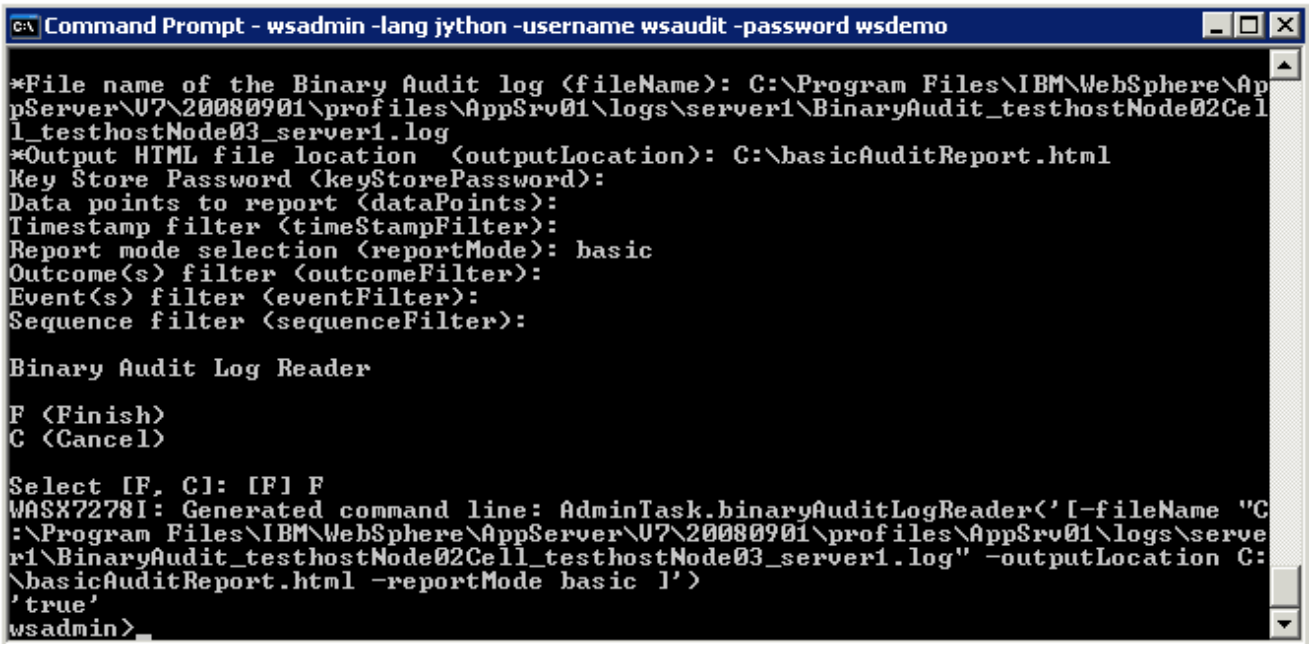
__ b. Once the wsadmin shell has started, enter the following command to generate an html report

```
AdminTask.binaryAuditLogReader('-interactive')
```

__ c. The interactive mode will prompt for input for the following questions. Enter the following:

- **filename:**

  <profile_root>\logs\server1\BinaryAudit_<cellName>_<nodeName>_server1.log
- **outputLocation:** **C:\basicAuditReport.html**
- **Key Store Password:** **<blank>**
- **Data points:** **<blank>**
- **Timestamp filter:** **<blank>**
- **Report mode selection:** **basic**
- **Events filter:** **<blank>**
- **Outcomes filter:** **<blank>**
- **Sequence filter:** **<blank>**
- **Select [F, C]:** **F**



__ d. At this point an html file by the name of **basicAuditReport.html** is generated. With a Windows Explorer window, browse to the C:\ directory and double click **basicAuditReport.html**.

Audit Records

Hostname was7host01 . ReportTime Jun 30, 2008, 15:05:50

Record Number	Event Type	Outcome
0	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Thu Jun 26 17:20:17 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
1	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Thu Jun 26 17:20:17 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
2	SECURITY_AUTHN	DENIED
CreationTime=Thu Jun 26 17:24:44 EDT 2008	Action=webAuth	ProgName=/navigatorCmd.do
RemoteAddr=192.168.128.142	RemotePort=2429	RemoteHost=192.168.128.142
ResourceName=GET	ResourceType=web	ResourceUniqueId=0
3	SECURITY_AUTHN	DENIED
CreationTime=Thu Jun 26 17:24:44 EDT 2008	Action=webAuth	ProgName=/navigatorCmd.do
RemoteAddr=192.168.128.142	RemotePort=2429	RemoteHost=192.168.128.142
ResourceName=GET	ResourceType=web	ResourceUniqueId=0

Part 4: (Optional) Create a new event filter

At this point, security auditing is configured and enabled and the logs have been viewed both through a text interface and an HTML report. Those are the most basic steps for getting started with auditing.

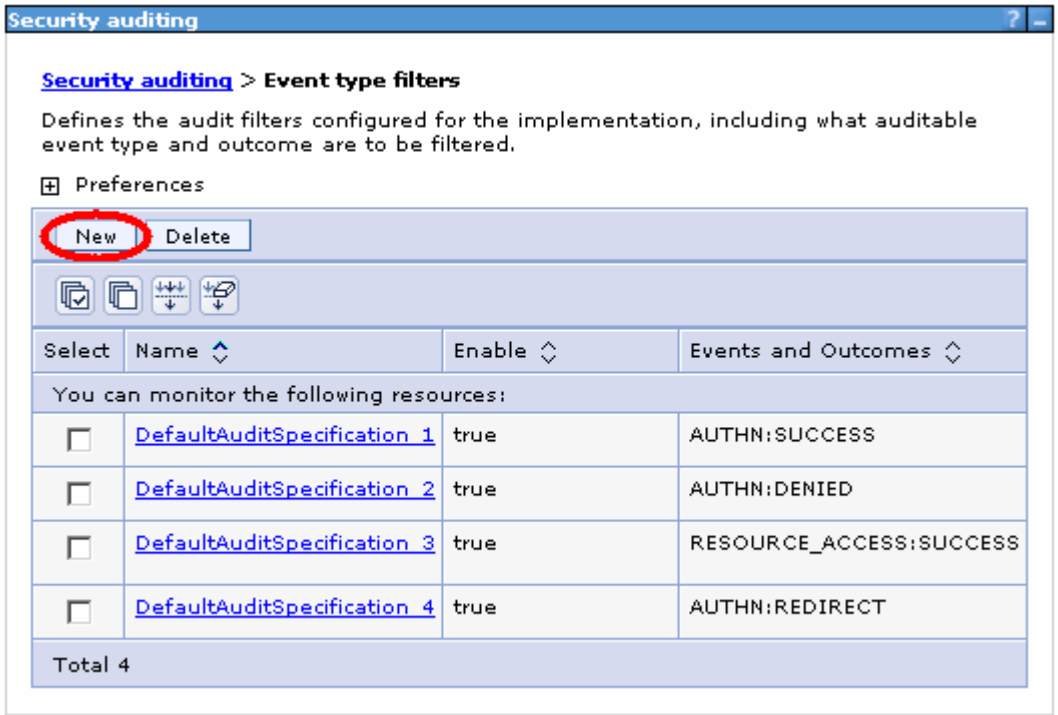
The rest of the exercise goes through some additional features including configuring additional filters and encryption of the audit data. Since these features might not be of interest to all students, these parts have been marked as optional.

In this part of the exercise, an additional event filter is created. This filter tells the audit service to audit any authorization failures.

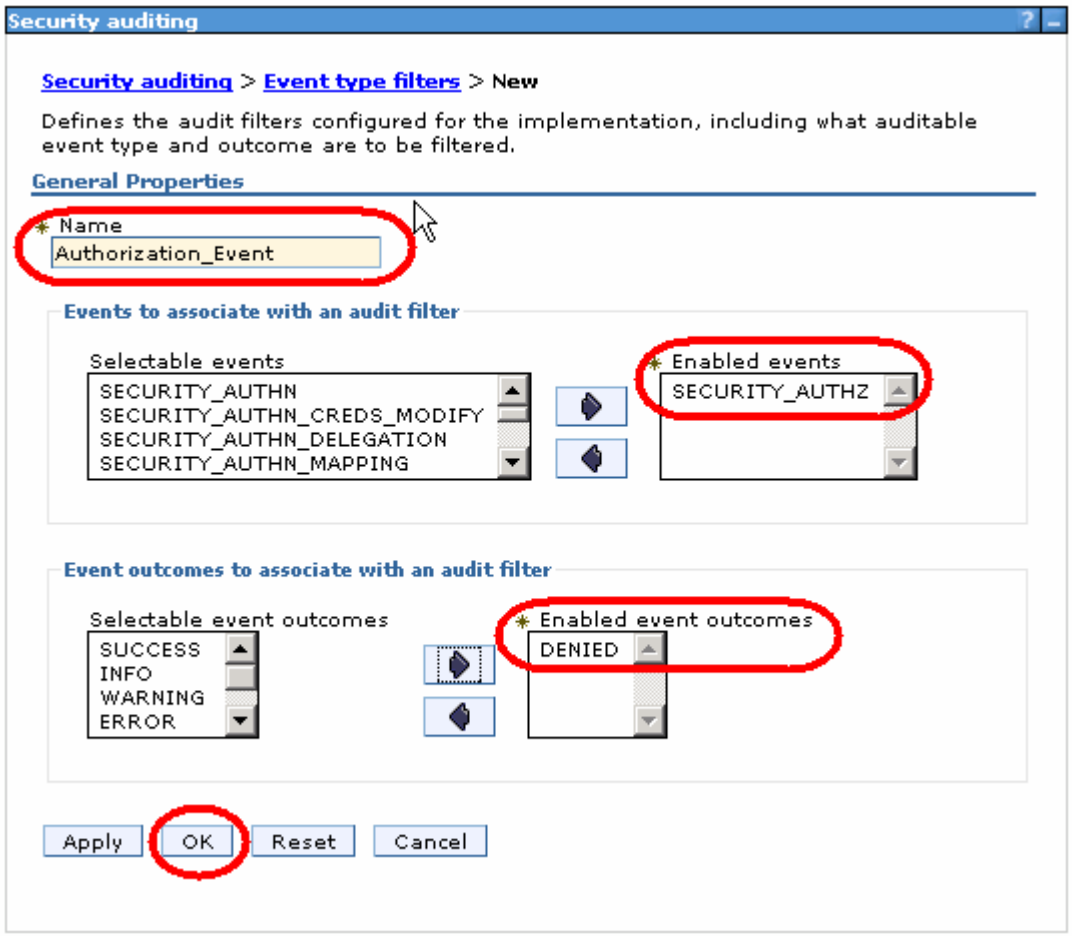
____ 1. The first step will be to add and configure the new event filter.

__ a. Using the administrative console, log in as **wsaudit**. Go to the **Security auditing** page, and click **Event type filters** under **Related Items**.

__ b. There are four default filters, including authentication success, denied and redirect. There is also one resource_access filter. To create a new filter, click **New**.



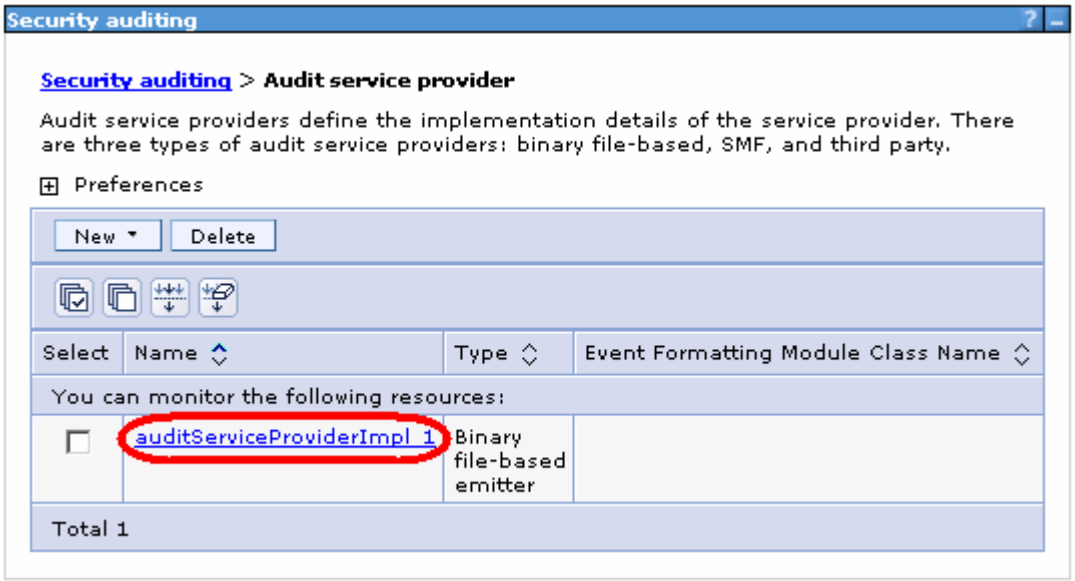
__ c. Enter **Authorization_Event** for the Name. Select **SECURITY_AUTHZ** from the **Selectable events** region and click the **right arrow** to move it into the **Enabled events**. Then select **DENIED** from the **Selectable events outcomes** and click the **right arrow** to move it into the **Enabled event outcomes**.



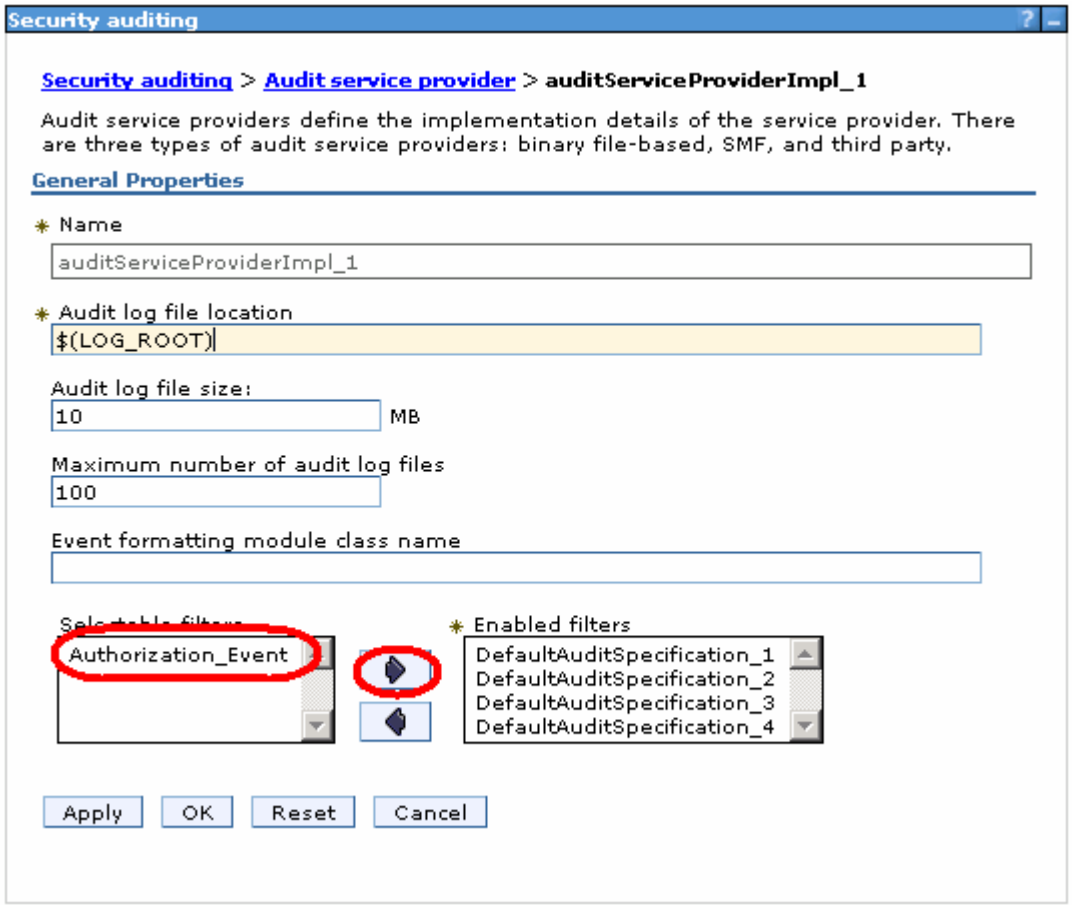
__ d. Click **OK** and **Save** the changes.

____ 2. Notice that there is a new event defined. But this event will not be audited until further configuration is complete. The next step is to configure the service provider.

__ a. Go back to the **Security auditing** page and click **Audit service provider**. There will be only one defined at this point, click **auditServiceProviderImpl_1**.



__ b. Notice that the **Authorization_Event** that was just created is listed under the **Selectable filters**, but is not part of the **Enabled filters** list. Select the new filter and click the **right arrow** to move it to the **Enabled filters** list.

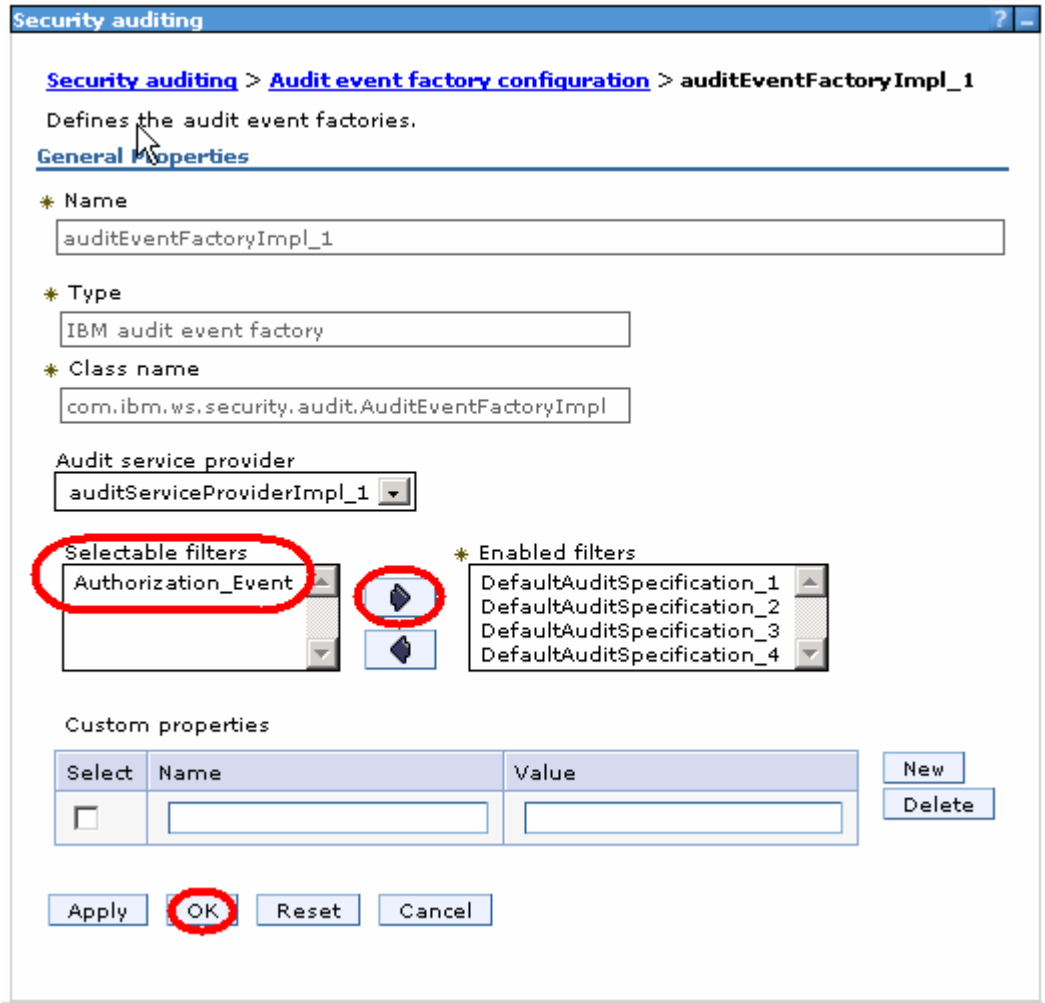


__ c. Click **OK** and **Save** the changes.

____ 3. Update the event factory configuration.

__ a. Return to the **Security auditing** page and click on **Audit event factory configuration**. There will be only one defined at this point, click **auditEventFactoryImpl_1**.

__ b. Like in the service provider screen, move the **Authorization_Event** to the **Enabled filters** for the event factory.



__ c. Click **OK** and **Save** the changes.

Note: The event factory is where the configuration is done to define what events are gathered. The service **provider** is where the configuration occurs to define which events are reported. See the Information Center for details on the numerous other event types that can be configure.

____ 4. Restart the application server and verify that these updates are doing what is expected.

__ a. **Restart the application server** in order for the changes to take effect.

__ b. Once the application server has been restart, look at the **BinaryAudit.log** file in the server's log directory. Take note of the latest sequence number.

__ c. Now, attempt to stop the application server using **wsaudit** as the username. Since the **wsaudit** user is not a console administrator, this should fail.

```
Command Prompt
C:\Program Files\IBM\WebSphere\AppServer\profiles\Audit\bin>stopServer.bat serve
r1 -username wsaudit -password wsdemo
ADMU0116I: Tool information is being logged in file C:\Program
           Files\IBM\WebSphere\AppServer\profiles\Audit\logs\server1\stopServer.
log
ADMU0128I: Starting tool with the Audit profile
ADMU3100I: Reading configuration for server: server1
ADMU0111E: Program exiting with error: javax.management.JMRuntimeException:
           ADMN0022E: Access is denied for the stop operation on Server MBean
           because of insufficient or empty credentials.
ADMU4113E: Verify that username and password information is correct.  If
           running tool from the command line, pass in the correct -username
           and -password.  Alternatively, update the <conntype>.client.props
           file.
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
ADMU0211I: Error details may be seen in the file: C:\Program
           Files\IBM\WebSphere\AppServer\profiles\Audit\logs\server1\stopServer.
log

C:\Program Files\IBM\WebSphere\AppServer\profiles\Audit\bin>
```

__ d. Once the stopServer command has failed, look at the **BinaryAudit.log** file again. Look for the **SECURITY_AUTHZ** entry that shows the denial.

```
Command Prompt
Seq = 1 | Event Type = SECURITY_RESOURCE_ACCESS | Outcome = SUCCESSFUL | Outcome
Reason = SUCCESS | OutcomeReasonCode = 6 | SessionId = N/A | RemoteAddr = null |
 RemotePort = null | RemoteHost = null | ProgName = Server (module) | Action = p
reinvoke MBean | RegistryUserName = null | AppUserName = null | AccessDecision =
 authnSuccess | ResourceName = getState | ResourceType = SM_MBEAN | ResourceUniq
ueId = 0 | PermissionsChecked = null | PermissionsGranted = null | RolesChecked
= N/A | RolesGranted = null | EventTrailId = 457691007 | CreationTime = Mon Jun
30 17:34:56 EDT 2008 | GlobalInstanceId = 0 | FirstCaller = null | Realm = defau
ltWIMFileBasedRealm | RegistryType = null | Url = N/A
Seq = 2 | Event Type = SECURITY_AUTHZ | Outcome = UNSUCCESSFUL | OutcomeReason =
 DENIED | OutcomeReasonCode = 16 | SessionId = N/A | RemoteAddr = null | RemoteP
ort = null | RemoteHost = null | ProgName = Server.stop:java.lang.Boolean:java.l
ang.Integer | Action = authz | RegistryUserName = null | AppUserName = defaultWI
MFileBasedRealm/wsaudit | AccessDecision = authzDenied | ResourceName = Server |
 ResourceType = WAS | ResourceUniqueId = 0 | PermissionsChecked = null | Permiss
ionsGranted = null | RolesChecked = operator , administrator | RolesGranted = nu
ll | EventTrailId = 1079617506 | CreationTime = Mon Jun 30 17:36:29 EDT 2008 | G
lobalInstanceId = 0 | FirstCaller = null | Realm = defaultWIMFileBasedRealm | Re
gistryType = WIMUserRegistry | Provider = WebSphere | ProviderStatus = providerS
uccess | PolicyName = null | PolicyType = null

C:\Program Files\IBM\WebSphere\AppServer\profiles\Audit\bin>_
```

Part 5: (Optional) Digitally sign the audit log entries

By default, the auditing data is stored in clear text. Although this provides useful information, it could potentially be tampered with. To help deal with this issue, the data can be digitally signed, encrypted or both. This part of the exercise turns on digital signatures for the audit data ensuring the integrity of the data.

The administrator is able to choose which certificate's private key is used to digitally sign the log entries. This then means that only the corresponding public key is needed to validate the signature. For an additional level of security, turning on digital signing also has the side effect of having the log entries 64-bit encoded.

____ 1. For this part of the exercise, administrative access is required for the console (not just auditor access).

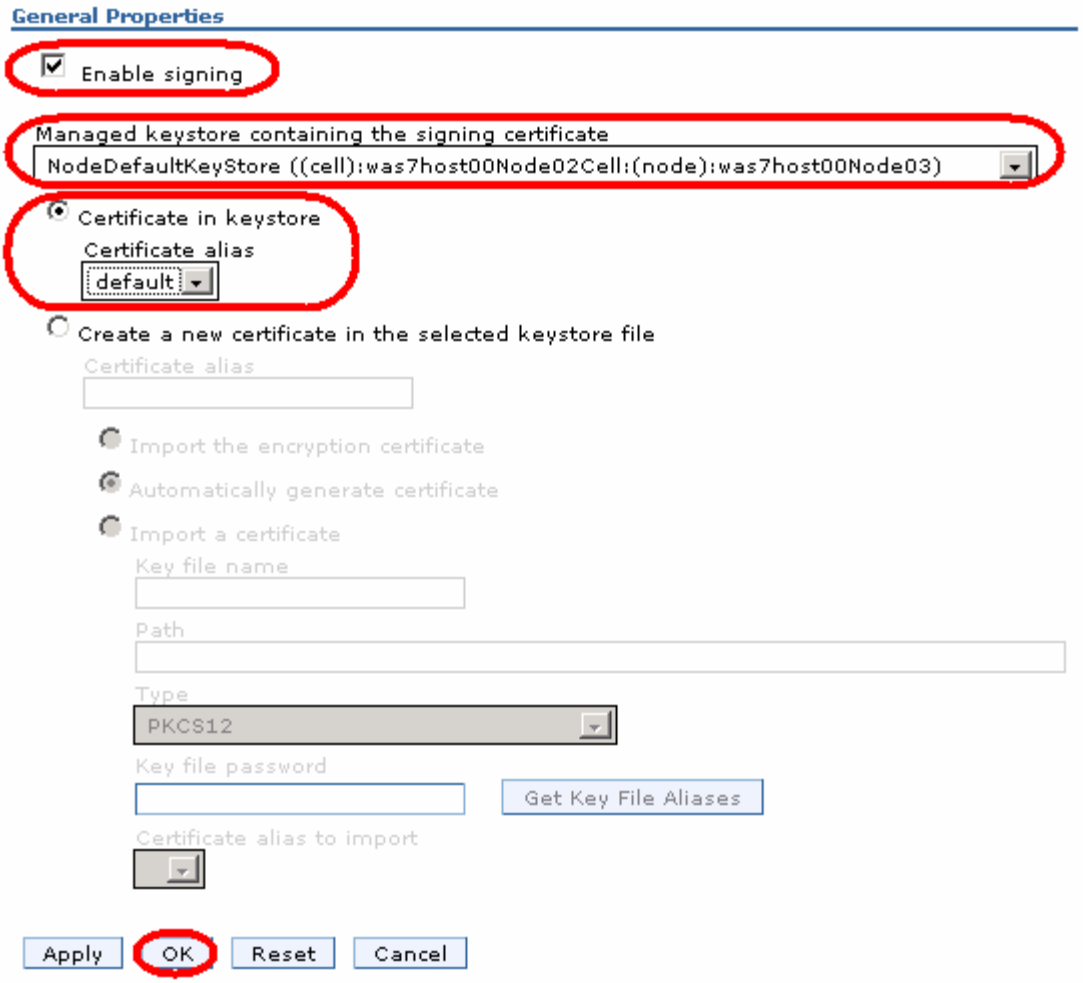
__ a. In the administrative console window, **logout** as the **wsaudit** user.

__ b. Log in again as **wsdemo**, which has implicit access as an administrator.

____ 2. Turn on digital signing for the audit logs.

__ a. Return to the **Security auditing** page of the administrative console and click **Audit record signing configuration**.

__ b. Check the **Enable signing** box. Accept the default for the **Managed keystore containing the signing certificate**, which should be the **NodeDefaultKeyStore**. For the **Certificate alias** under **Certificate in keystore**, select **default** from the pulldown.

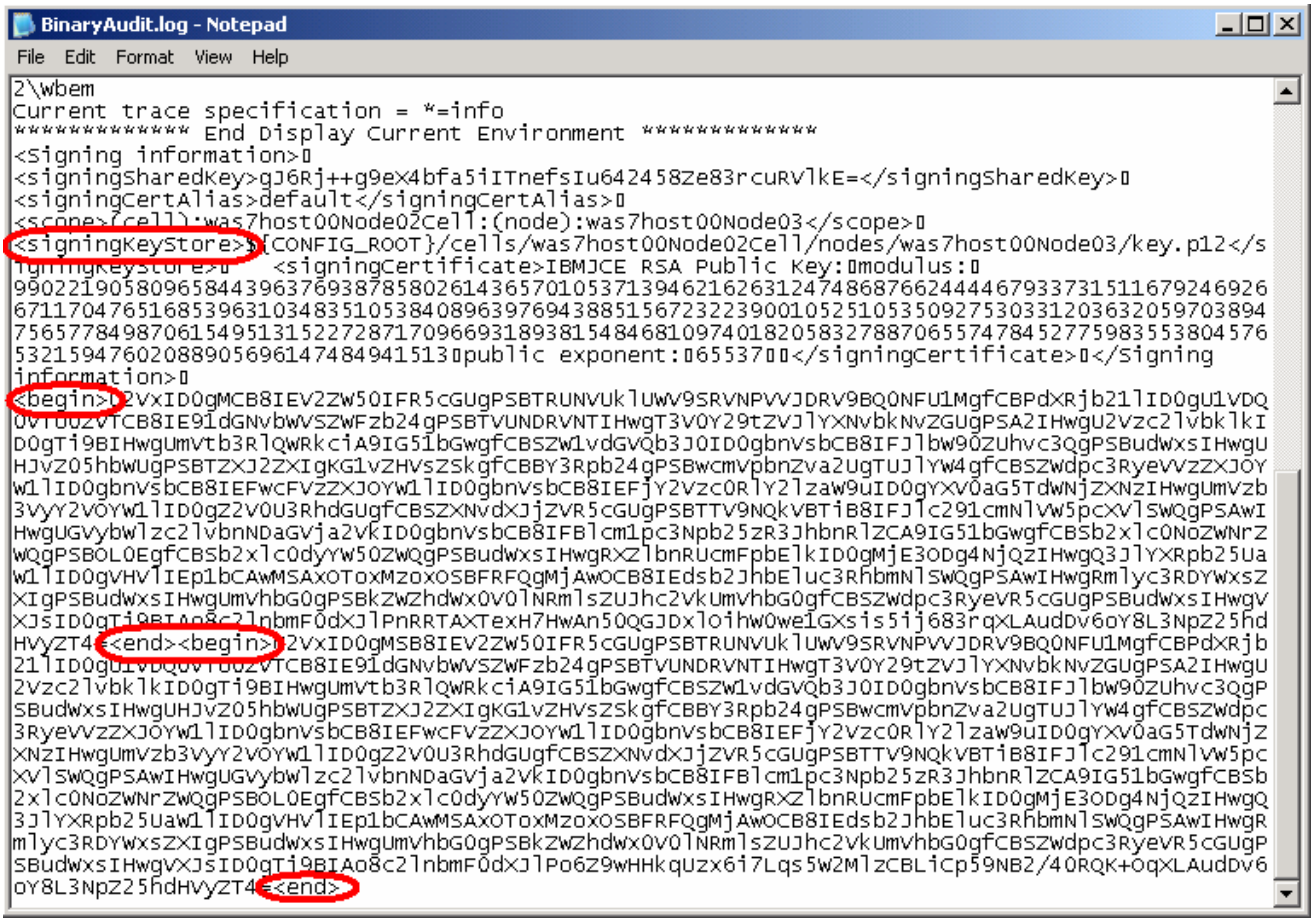


__ c. Click **OK** and **Save** the changes.

__ d. **Restart the application server** to have the changes take affect.

____ 3. View the audit log and take note that the log entries are now encoded.

__ a. Using a text editor, open the new BinaryAudit.log file. Notice that the records are now encoded. The file header also includes specific information on the keys used for digitally signing the records.



```
2\wbem
Current trace specification = *=info
************** End Display Current Environment **************
<Signing information>▯
<signingSharedKey>gJ6Rj++g9eX4bfa5iITnefsIu642458Ze83rcuRVlkE=</signingSharedKey>▯
<signingCertAlias>default</signingCertAlias>▯
<scope>(cell):was7host00Node02Cell:(node):was7host00Node03</scope>▯
<signingKeyStore>${CONFIG_ROOT}/cells/was7host00Node02Cell/nodes/was7host00Node03/key.p12</s
igningKeyStore>▯   <signingCertificate>IBMJCE RSA Public Key:▯modulus:▯
99022190580965844396376938785802614365701053713946216263124748687662444467933731511679246926
67117047651685396310348351053840896397694388515672322390010525105350927530331203632059703894
75657784987061549513152272871709669318938154846810974018205832788706557478452775983553804576
53215947602088905696147484941513▯public exponent:▯65537▯▯</signingCertificate>▯</Signing
information>▯
<begin>U2VxID0gMCB8IEV2ZW50IFR5cGUgPSBTRUNVUklUWV9SRVNPVVJDRV9BQ0NFU1MgfCBPdXRjb21lID0gU1VDQ
0VTU0ZVTCB8IE91dGNvbWVSZWFzb24gPSBTVUNDRVNTIHwgT3V0Y29tZVJlYXNvbkNvZGUgPSA2IHwgU2Vzc2lvbklkI
D0gTi9BIHwgUmVtb3RlQWRkciA9IG51bGwgfCBSZW1vdGVQb3J0ID0gbnVsbCB8IFJlbW90ZUhvc3QgPSBudWxsIHwgU
HJvZ05hbWUgPSBTZXJ2ZXIgKG1vZHVsZSkgfCBBY3Rpb24gPSBwcmVpbnZva2UgTUJlYW4gfCBSZWdpc3RyeVVzZXJOY
W1lID0gbnVsbCB8IEFwcFVzZXJOYW1lID0gbnVsbCB8IEFjY2Vzc0RlY2lzaW9uID0gYXV0aG5TdWNjZXNzIHwgUmVzb
3VyY2VOYW1lID0gZ2V0U3RhdGUgfCBSZXNvdXJjZVR5cGUgPSBTTV9NQkVBTiB8IFJlc291cmNlVW5pcXVlSWQgPSAwI
HwgUGVybWlzc2lvbnNDaGVja2VkID0gbnVsbCB8IFBlcm1pc3Npb25zR3JhbnRlZCA9IG51bGwgfCBSb2xlc0NoZWNrZ
WQgPSBOL0EgfCBSb2xlc0dyYW50ZWQgPSBudWxsIHwgRXZlbnRUcmFpbElkID0gMjE3ODg4NjQzIHwgQ3JlYXRpb25Ua
W1lID0gVHVlIEp1bCAwMSAxOToxMzoxOSBFRFQgMjAwOCB8IEdsb2JhbEluc3RhbmNlSWQgPSAwIHwgRmlyc3RDYWxsZ
XIgPSBudWxsIHwgUmVhbG0gPSBkZWZhdWx0V0lNRmlsZUJhc2VkUmVhbG0gfCBSZWdpc3RyeVR5cGUgPSBudWxsIHwgV
XJsID0gTi9BIAo8c2lnbmF0dXJlPnRRTAXTexH7HwAn5OQGJDxloihW0we1Gxsis5ij683rqXLAudDv6oY8L3NpZ25hd
HVyZT4=<end><begin>U2VxID0gMSB8IEV2ZW50IFR5cGUgPSBTRUNVUklUWV9SRVNPVVJDRV9BQ0NFU1MgfCBPdXRjb
21lID0gU1VDQ0VTU0ZVTCB8IE91dGNvbWVSZWFzb24gPSBTVUNDRVNTIHwgT3V0Y29tZVJlYXNvbkNvZGUgPSA2IHwgU
2Vzc2lvbklkID0gTi9BIHwgUmVtb3RlQWRkciA9IG51bGwgfCBSZW1vdGVQb3J0ID0gbnVsbCB8IFJlbW90ZUhvc3QgP
SBudWxsIHwgUHJvZ05hbWUgPSBTZXJ2ZXIgKG1vZHVsZSkgfCBBY3Rpb24gPSBwcmVpbnZva2UgTUJlYW4gfCBSZWdpc
3RyeVVzZXJOYW1lID0gbnVsbCB8IEFwcFVzZXJOYW1lID0gbnVsbCB8IEFjY2Vzc0RlY2lzaW9uID0gYXV0aG5TdWNjZ
XNzIHwgUmVzb3VyY2VOYW1lID0gZ2V0U3RhdGUgfCBSZXNvdXJjZVR5cGUgPSBTTV9NQkVBTiB8IFJlc291cmNlVW5pc
XVlSWQgPSAwIHwgUGVybWlzc2lvbnNDaGVja2VkID0gbnVsbCB8IFBlcm1pc3Npb25zR3JhbnRlZCA9IG51bGwgfCBSb
2xlc0NoZWNrZWQgPSBOL0EgfCBSb2xlc0dyYW50ZWQgPSBudWxsIHwgRXZlbnRUcmFpbElkID0gMjE3ODg4NjQzIHwgQ
3JlYXRpb25UaW1lID0gVHVlIEp1bCAwMSAxOToxMzoxOSBFRFQgMjAwOCB8IEdsb2JhbEluc3RhbmNlSWQgPSAwIHwgR
mlyc3RDYWxsZXIgPSBudWxsIHwgUmVhbG0gPSBkZWZhdWx0V0lNRmlsZUJhc2VkUmVhbG0gfCBSZWdpc3RyeVR5cGUgP
SBudWxsIHwgVXJsID0gTi9BIAo8c2lnbmF0dXJlPo6Z9wHHkqUzx6i7Lqs5W2MlzCBLiCp59NB2/40RQK+0qXLAudDv6
oY8L3NpZ25hdHVyZT4=<end>
```

__ b. Now verify that the html reports can still be generated correctly. In a command window, start wsadmin from the profile’s bin directory with the following command:

```
wsadmin –lang jython –username wsaudit –password wsdemo
```

__ c. Once the wsadmin shell has started, enter the following command to generate an html report

```
AdminTask.binaryAuditLogReader('-interactive’)
```

__ d. The interactive mode will prompt for input for the following questions. Enter the following:

- **filename:**
  <profile_root>\logs\server1\BinaryAudit<cellName>_<nodeName>_server1.log
- **outputLocation:** **C:\signedAuditReport.html**
- **Key Store Password:** **<blank>**
- **Data points:** **<blank>**
- **Timestamp filter:** **<blank>**
- **Report mode selection:** **basic**
- **Events filter:** **<blank>**
- **Outcomes filter:** **<blank>**
- **Sequence filter:** **<blank>**
- **Select [F, C]:** **F**

__ e. Using Windows Explorer, go to C:\ and double click on signedAuditReport.html. This will open the HTML report in a browser. Notice that the entries in this report look exactly like they did before the signing was turned on.

Audit Records

Hostname was7host00 . ReportTime Jul 1, 2008, 23:26:51

Record Number	Event Type	Outcome
0	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Tue Jul 01 19:13:19 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
1	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Tue Jul 01 19:13:19 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
2	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Tue Jul 01 23:24:20 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getProcessType	ResourceType=SM_MBEAN	ResourceUniqueId=0
3	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Tue Jul 01 23:24:44 EDT 2008	Action=execute command	ProgName=com.ibm.websphere.management.cmdframe
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getNodeBaseProductVersion	ResourceType=SM_COMMAND	ResourceUniqueId=0
4	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Tue Jul 01 23:26:49 EDT 2008	Action=execute command	ProgName=com.ibm.ws.security.audit.tools.binaryAudit
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=binaryAuditLogReader	ResourceType=SM_COMMAND	ResourceUniqueId=0

Part 6: (Optional) Encrypt the audit logs

If the intention is to not just protect the integrity of the data, but actually encrypt it, that is possible as well. In this part of the exercise, the log entries will be both encrypted and signed, but it certainly is possible to encrypt them and not sign them.

The first step toward encrypting the log entries is to create a new key store and certificate specifically for audit encryption.

____ 1. Log into the console as **wsaudit**.

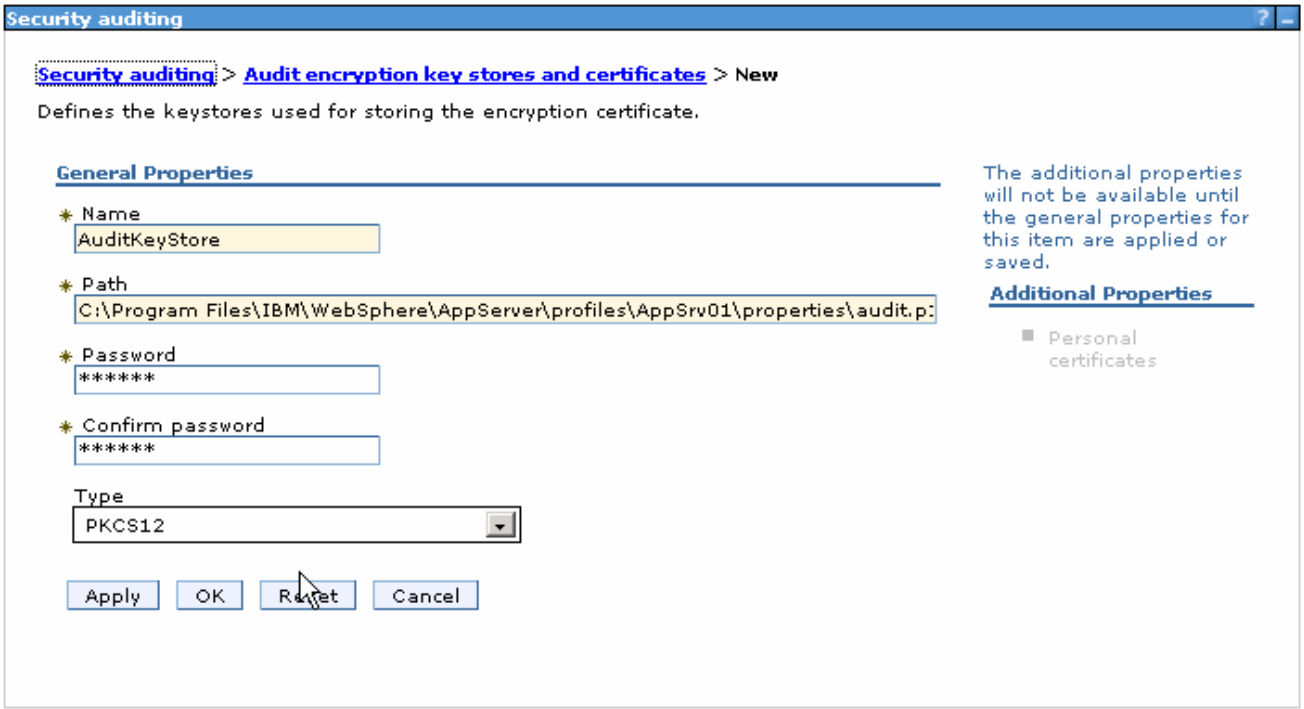
__ a. This section requires being logged in as the **wsaudit** console user since it has auditor access.

____ 2. Create a key store and certificate for audit encryption.

__ a. Using the **administrative console**, logged in as **wsaudit**, go to the **Security auditing** page. Click **Audit encryption key stores and certificates** under **Related Items**.

__ b. Click **New** to create a new key store and certificate.

__ c. For the **name**, enter **AuditKeyStore** and fore the Path enter **C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\properties\audit.p12**. Enter **wsdemo** in the **Password** fields and accept the default **Type** of **PKCS12**.

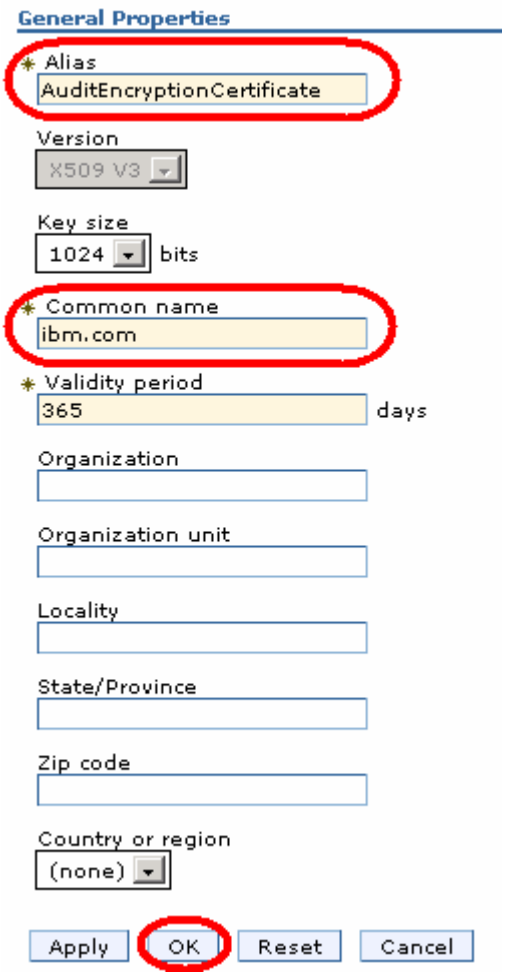


__ d. Click **OK** and **Save** the changes.

__ e. Next the actual certificate needs to be created. Click the **AuditKeyStore** in the **Audit encryption key stores and certificates** page. On the right side, click **Personal certificates** under **Additional Properties**.

__ f. Click **Create self-signed Certificate** to create the new certificate.

__ g. Enter **AuditEncryptionCertificate** for the **Alias** and **ibm.com** for the **Common name**.

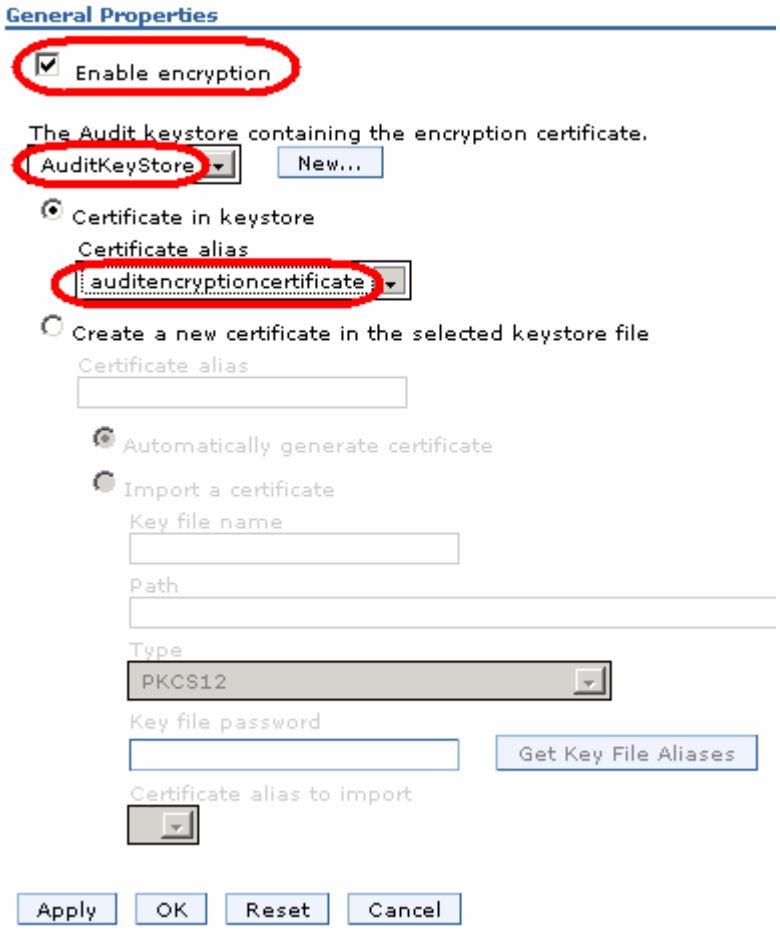


__ h. Click **OK**.

____ 3. Turn on encryption for the audit logs.

__ a. Return to the **Security auditing** page and click **Audit record encryption configuration**.

__ b. Check the **Enable encryption** box. Accept the default **keystore** of **AuditKeyStore** and the default **Certificate alias** of **auditencryptioncertificate**.



__ c. Click OK and Save the changes.

__ d. **Restart the application server** to have the changes take effect.

____ 4. View the audit log and take note that the log entries are now encrypted.

__ a. Using a text editor, open the new BinaryAudit.log file. Notice that the file header now includes encryption certificate information; otherwise the individual entries look much the same as they did when the records were merely signed.

__ b. Now verify that the html reports can still be generated correctly. In a command window, start wsadmin from the profile's bin directory with the following command:

```
wsadmin –lang jython –username wsaudit –password wsdemo
```

__ c. Once the wsadmin shell has started, enter the following command to generate an HTML report

```
AdminTask.binaryAuditLogReader('-interactive')
```

__ d. The interactive mode will prompt for input for the following questions. Enter the following (note – this time the key Store Password is required):

- **filename:**

  <profile_root>\logs\server1\BinaryAudit_<cellName>_<nodeName>_server1.log
- **outputLocation:** **C:\encryptedAuditReport.html**
- **Key Store Password:** **wsdemo**
- **Data points:** **<blank>**
- **Timestamp filter:** **<blank>**
- **Report mode selection:** **basic**
- **Events filter:** **<blank>**
- **Outcomes filter:** **<blank>**
- **Sequence filter:** **<blank>**
- **Select [F, C]:** **F**

__ e. Using Windows Explorer, go to C:\ and double click encryptedAuditReport.html. This will open the HTML report in a browser. Notice that the entries in this report look exactly like they did before the signing and encryption was turned on.

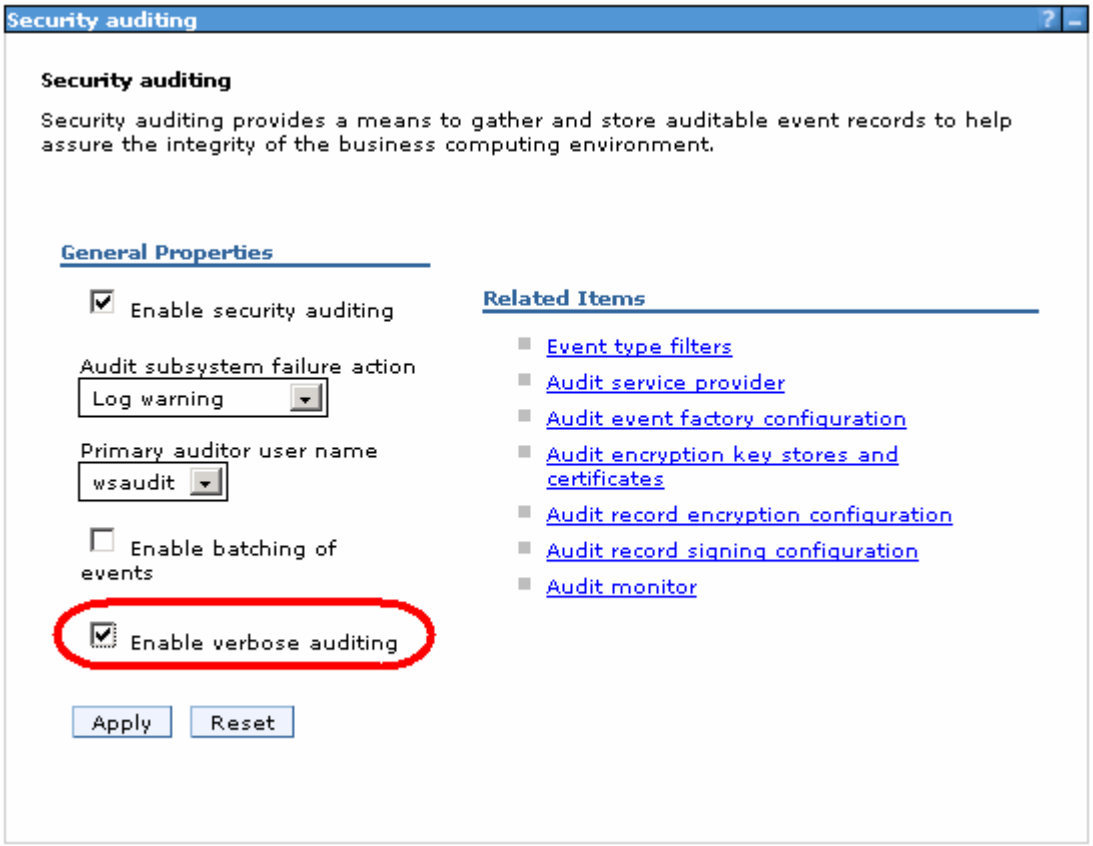
Part 7: (Optional) Verbose logging and reporting

Finally, for comparison, this section of the exercise turns on verbose audit logging and generates a report with the complete mode.

____ 1. Turn on verbose logging for security auditing.

__ a. In the administrative console, return to the **Security auditing** page.

__ b. Check the **Enable verbose auditing** box.



__ c. Click **Apply** and **Save** the changes.

____ 2. In order to read the log files in clear text, disable both signing and encryption.

__ a. In the **Security auditing** page, click **Audit record encryption configuration**.

__ b. Uncheck **Enable encryption** and click **OK**.

__ c. **Save** the changes.

__ d. In order to turn off signing, you will need to be logged into the console as an administrator user. **Logout** of the **wsaudit** session and **login** as **wsdemo**.

__ e. Return to the **Security auditing** page and click **Audit record signing configuration**.

__ f. Uncheck **Enable signing** and click **OK**.

__ g. **Save** the change.

____ 3. **Restart the application server** to have these changes take effect.

____ 4. Open the **BinaryAudit.log** in a text editor. Notice that the entries have additional information in them.

____ 5. Next, using wsadmin, generate an html report using the same process as before, but enter **complete** for the **reportMode** and **C:\completeAuditReport.**html for the **outputLocation**.

____ 6. Open the new audit report and notice that it also has more information than was available with the basic reportMode.

Audit Records

Hostname was7host00 . ReportTime Jul 2, 2008, 00:45:06

Record Number	Event Type	Outcome
0	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Wed Jul 02 00:33:54 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RegistryType=null	Domain=global	Realm=defaultWIMFileBasedRealm
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
LastEventTrailId=217888643	EventTrailId=217888643	GlobalInstanceId=0
AuthnType=null	Provider=null	ProviderStatus=null
MappedSecurityDomain=null	MappedRealm=null	MappedUserName=null
DelegationType=null	RoleName=null	IdentityName=null
FirstCaller=null	CallerList=null	TerminateReason=null
AccessDecision=authnSuccess	PolicyName=null	PolicyType=null
PermissionsChecked=null	PermissionsGranted=null	RolesChecked=N/A
RolesGranted=null	MgmtType=null	MgmtCommand=null
TargetInfoName=null	TargetInfoUniqueId=null	Url=N/A
OutcomeReasonCode=6		
1	SECURITY_RESOURCE_ACCESS	SUCCESS
CreationTime=Wed Jul 02 00:33:54 EDT 2008	Action=preinvoke MBean	ProgName=Server (module)
RegistryType=null	Domain=global	Realm=defaultWIMFileBasedRealm
RemoteAddr=null	RemotePort=null	RemoteHost=null
ResourceName=getState	ResourceType=SM_MBEAN	ResourceUniqueId=0
LastEventTrailId=217888643	EventTrailId=217888643	GlobalInstanceId=0
AuthnType=null	Provider=null	ProviderStatus=null
MappedSecurityDomain=null	MappedRealm=null	MappedUserName=null
DelegationType=null	RoleName=null	IdentityName=null
FirstCaller=null	CallerList=null	TerminateReason=null
AccessDecision=authnSuccess	PolicyName=null	PolicyType=null
PermissionsChecked=null	PermissionsGranted=null	RolesChecked=N/A
RolesGranted=null	MgmtType=null	MgmtCommand=null
TargetInfoName=null	TargetInfoUniqueId=null	Url=N/A
OutcomeReasonCode=6		

What you did in this exercise

In this lab you learned how to enable security auditing for WebSphere Application Server Network Deployment V7. You created an auditor user, configured and enabled auditing, and viewed the text based log files and the generated html report. In the optional parts of this exercise, you created a new event filter, digitally signed the audit log entries and the encrypted them. Finally, you switched the auditing level to verbose and generated a “complete” audit report.

This page is left intentionally blank.