ADOBE® LIVECYCLE® ES4 HARDENING AND SECURITY GUIDE

Legal notices

Contents

Chapter 1: About This Document

This document contains information about how to maximize the security of the Adobe® LiveCycle® Enterprise Suite (ES4) production environment.

Additional security information for LiveCycle is available at the LiveCycle Developer Center.

Security advisories and bulletins for LiveCycle are available at the Adobe Security Bulletins and Advisories Site.

1.1 Who should read this document?

This document is intended for consultants, security specialists, systems architects, and IT professionals who are responsible for planning application or infrastructure development and deployment of LiveCycle. These roles include the following common roles:

- IT and operations' engineers who must deploy secure web applications and servers in their own or customer organizations
- Architects and planners who are responsible for planning the architectural efforts for the clients in their organizations
- IT security specialists who focus on providing security across the platforms within their organizations
- Consultants from Adobe and partners who require detailed resources for customers and partners

1.2 Conventions used in this document

This document uses the following naming conventions for common file paths.

Name	Default value	Description
[LiveCycle root]	Windows: C:\Adobe\Adobe LiveCycle ES4 Linux and UNIX: opt/adobe/adobe_livecycle_ES4	The installation directory that is used for all LiveCycle modules. The installation directory contains subdirectories for the Adobe® LiveCycle® Configuration Manager. This directory also includes directories relating to third-party technologies.
[JBoss root]	C:\Adobe\Adobe LiveCycle ES4\jboss	(JBoss Turnkey) The home directory of the application server that runs LiveCycle.

1.3 Additional information

The resources in this table can help you learn about LiveCycle.

For information about	See
LiveCycle, the LiveCycle solutions, and development tools	LiveCycle Overview
Preparing your environment for installing or upgrading to LiveCycle	Preparing to Install LiveCycle (Single Server) Preparing to Install LiveCycle (Server Cluster) Preparing to Upgrade to LiveCycle ES4
Installing LiveCycle (Single Server)	Installing and Deploying LiveCycle Using JBoss Turnkey Installing and Deploying LiveCycle for JBoss Installing and Deploying LiveCycle for WebLogic Installing and Deploying LiveCycle for WebSphere
Configuring LiveCycle (Server Cluster)	Configuring LiveCycle Application Server Clusters Using JBoss Configuring LiveCycle Application Server Clusters Using WebLogic Configuring LiveCycle Application Server Clusters Using WebSphere
Upgrading to LiveCycle	Upgrading to LiveCycle ES4 for JBoss Turnkey Upgrading to LiveCycle ES4 for JBoss Upgrading to LiveCycle ES4 for WebLogic Upgrading to LiveCycle ES4 for WebSphere
Installing LiveCycle - Workbench 11	Installing LiveCycle Workbench
Performing general administrative tasks for LiveCycle	LiveCycle Administration Help
Other services and products that integrate with LiveCycle	http://www.adobe.com
LiveCycle documentation set	LiveCycle Documentation Set

Chapter 2: General Security Considerations

This section provides introductory information that helps you prepare for hardening your LiveCycle environment. It includes prerequisite information about LiveCycle, operating system, application server, and database security. You should review this information before you continue to lock down your environment.

2.1 Vendor-specific security information

This section contains security-related information about operating systems, application servers, and databases that are incorporated into your LiveCycle solution.

Use the links in this section to find vendor-specific security information for your operating system, database, and application server.

2.1.1 Operating system security information

When securing your operating system, carefully consider implementing the measures described by your operating system vendor, including these:

- Defining and controlling users, roles, and privileges
- Monitoring logs and audit trails
- Removing unnecessary services and applications
- Backing up files

For security information about operating systems that LiveCycle supports, see the resources in the this table.

Operating System	Security Resource
IBM® AIX® 5.3 and 6.1	IBM AIX Security Benefits
Microsoft® Windows® XP SP 2 (for non-production environments only)	Windows XP Security Guide
Microsoft Windows 7, 32-bit and 64-bit (for non-production environments only)	Windows 7 Security Guide
Microsoft Windows Server® 2003 Enterprise or Standard Edition	Search for "Windows Server 2003 Security Guide" at Microsoft.com
Microsoft Windows Server® 2008 Enterprise or Standard Edition	Search for "Windows Server 2008 Security Guide" at Microsoft.com
Microsoft Vista™ SP1, all flavors, 32-bit and 64-bit (for non-production environments only)	Search for "Windows Vista Security Guide" at Microsoft.com
Red Hat® Linux® AP or ES	Red Hat Enterprise Linux Security Guide
Sun Solaris 10	System Administration Guide: Security Services

2.1.2 Application server security information

When securing your application server, you should carefully consider implementing the measures described by your server vendor, including these:

- Using non-obvious administrator user name
- Disabling unnecessary services
- Securing the console manager
- Enabling secure cookies
- Closing unneeded ports
- Limiting clients by IP addresses or domains
- Using the Java™ Security Manager to programmatically restrict privileges

For security information about application servers that LiveCycle supports, see the resources in this table.

Application Server	Security Resource
Oracle WebLogic®	Search for Understanding WebLogic Security at http://download.oracle.com/docs/.
IBM WebSphere®	Securing applications and their environment
Red Hat® JBoss®	Security on JBoss

2.1.3 Database security information

When securing your database, you should consider implementing the measures described by your database vendor, including these:

- Restricting operations with access control lists (ACLs)
- Using non-standard ports
- Hiding the database behind a firewall
- Encrypting sensitive data before writing it to the database (see the database manufacturer's documentation)

For security information about databases that LiveCycle supports, see the resources in this table.

Database	Security Resource
IBM DB2® 9.1 or 9.5	DB2 Product Family Library
Microsoft SQL Server 2005 SP2 or 2008	Search the Web for "SQL Server 2005: Security" Search the Web for "SQL Server 2008: Security"
MySQL 5	MySQL 5.0 General Security Issues MySQL 5.1 General Security Issues
Oracle® 10g or 11g	See the Security chapter in the Oracle 11g documentation

This table describes the default ports that are required to be open during your LiveCycle configuration process. If you are connecting over https, adjust your port information and IP addresses accordingly. For more information about configuring ports, see the *Installing and Deploying LiveCycle* document for your application server.

Product or service	Port number
JBoss	8080
WebLogic	7001
WebLogic Managed Server	Set by administrator during configuration
WebSphere	9060, if Global Security is enabled the default SSL port value is 9043. 9080
BAM Server	7001
SOAP	8880
MySQL	3306
Oracle	1521
DB2	50000
SQL Server	1433
LDAP	The port on which the LDAP server is running. The default port is typically 389. However, if you select the SSL option, the default port is typically 636. You must confirm with your LDAP administrator which port to specify.

2.1.4 Configuring JBoss to use a non-default HTTP port

JBoss Application Server uses 8080 as the default HTTP port. JBoss also has pre-configured ports 8180, 8280, and 8380, which are commented out in the jboss-service.xml file. If you have an application on your computer that already uses this port, change the port that LiveCycle uses by following these steps:

1 Open the jboss-service.xml file in an editor.

JBoss turnkey install: *[JBoss root]*/server/lc_turnkey/conf/

JBoss manual install: *[appserver root]*/server/all/conf/

2 Locate and uncomment the following mbean:

```
<mbean code="org.jboss.services.binding.ServiceBindingManager"

name="jboss.system:service=ServiceBindingManager">

<attribute name="ServerName">ports-01</attribute>

<attribute name="StoreURL">${jboss.home.url}/docs/examples/binding-manager/sample-
bindings.xml</attribute>

<attribute name="StoreFactoryClassName">

org.jboss.services.binding.XMLServicesStoreFactory

</attribute>

</mbean>
```

3 Save and close the file.

4 Restart JBoss.

JBoss is now configured to use port 8180. If you need to use either 8280 or 8380, modify the `ServerName` attribute value to use one of the following alternative ports:

- For 8280: `ports-02`
- For 8380: `ports-03`

If you need to configure a port number other than those pre-configured for JBoss, perform the following steps:

1. Locate and open the deploy/jboss-web.deployer file in *[JBoss root]* (turnkey) or *[appserver root]* (JBoss manual install).
2. Locate and uncomment the mbean from step 2 above.
3. Modify the `ServerName` value to the port number to use.
4. Save and close the file.
5. Restart JBoss.

2.2 LiveCycle security considerations

This section describes some LiveCycle-specific security issues that you should know about.

2.2.1 Email credentials not encrypted in database

The email credentials stored by LiveCycle applications are not encrypted before they are stored in the LiveCycle database. When you configure a service endpoint to use email, any password information used as part of that endpoint configuration is not encrypted when it is stored in the database.

2.2.2 Sensitive content for Rights Management in the database

LiveCycle uses the LiveCycle database to store sensitive document key information as well as other cryptographic material that is used for policy documents. Securing the database against intrusion helps to protect this sensitive information.

2.2.3 Password in clear text format in adobe-ds.xml

The application server that is used to run LiveCycle requires its own configuration for access to your database through a data source that is configured on the application server. You should ensure that your application server does not expose your database password in clear text in its data source configuration file.

The adobe-ds.xml file contains passwords in clear text format. Consult your application server vendor about how to encrypt these passwords for your application server. For example, the JBoss® instructions are at Encrypting DataSource Passwords.

Note: *The LiveCycle JBoss turnkey installer encrypts the database password.*

IBM WebSphere Application Server and Oracle WebLogic Server may encrypt data source passwords by default. However, you should confirm with your application server documentation to ensure that this is happening.

2.2.4 Protecting the private key stored in Trust Store

The private keys or credentials imported in Trust Store are stored in LiveCycle database. You must take appropriate precautions to secure the database and restrict the access only to designated administrators.

Chapter 3: Hardening Your Environment

This section describes recommendations and best practices for securing servers that run LiveCycle. This is not a comprehensive host-hardening document for your operating system and . Instead, this section describes a variety of security-hardening settings that you should implement to enhance the security of LiveCycle that is running within a corporate intranet. To ensure that the LiveCycle application servers stay secure, however, you should also implement security monitoring, detection, and response procedures.

This section describes hardening techniques that should be applied during the following stages during the installation and configuration life cycle:

- **Pre-installation:** Use these techniques before you install LiveCycle.
- **Installation:** Use these techniques during the LiveCycle installation process.
- **Post-installation:** Use these techniques after installation and periodically thereafter.

LiveCycle is highly customizable and can work in many different environments. Some of the recommendations may not fit your organization's needs.

3.1 Preinstallation

Before installing LiveCycle, you can apply security solutions to the network layer and operating system. This section describes some issues and makes recommendations for reducing security vulnerabilities in these areas.

Installation and configuration on UNIX and Linux

You should not install or configure LiveCycle using a root shell. By default, files are installed under the /opt directory, and the user who performs the installation needs all file permissions under /opt. Alternatively, an installation can be performed under an individual user's /user directory where they already have all file permissions.

Installation and configuration on Windows

You should perform the installation on Windows as an administrator if you are installing LiveCycle on JBoss by using the turnkey method or if you are installing PDF Generator. Also, when installing PDF Generator on Windows with native application support, you must run the installation as the same Windows user who installed Microsoft Office. For more information about installation privileges, see the *Installing and Deploying LiveCycle* document for your application server.

3.1.1 Network layer security

Network security vulnerabilities are among the first threats to any Internet-facing or intranet-facing application server. This section describes the process of hardening hosts on the network against these vulnerabilities. It addresses network segmentation, Transmission Control Protocol/Internet Protocol (TCP/IP) stack hardening, and the use of firewalls for host protection.

The following table describes common processes that reduce network security vulnerabilities.

Issue	Description
Demilitarized zones (DMZs)	Deploy LiveCycle servers within a demilitarized zone (DMZ). Segmentation should exist in at least two levels with the application server used to run LiveCycle placed behind the inner firewall. Separate the external network from the DMZ that contains the web servers, which in turn must be separated from the internal network. Use firewalls to implement the layers of separation. Categorize and control the traffic that passes through each network layer to ensure that only the absolute minimum of required data is allowed.
Private IP addresses	Use Network Address Translation (NAT) with RFC 1918 private IP addresses on LiveCycle application servers. Assign private IP addresses (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) to make it more difficult for an attacker to route traffic to and from a NAT'd internal host through the Internet.
Firewalls	Use the following criteria to select a firewall solution: • Implement firewalls that support proxy servers and/or *stateful inspection* instead of simple packet-filtering solutions. • Use a firewall that supports a *deny all services except those explicitly permitted* security paradigm. • Implement a firewall solution that is dual-homed or multihomed. This architecture provides the greatest level of security and helps to prevent unauthorized users from bypassing the firewall security.
Database ports	Do not use default listening ports for databases (MySQL - 3306, Oracle - 1521, MS SQL - 1433). For information about changing database ports, see your database documentation. Using a different database port affects the overall LiveCycle configuration. If you change default ports, you must make corresponding modifications in other areas of configuration, such as the data sources for LiveCycle. For information about configuring data sources in LiveCycle, see *Installing andDeploying LiveCycle* or *Upgrading to LiveCycle*for your application server, at LiveCycle Documentation Set.

3.1.2 Operating system security

The following table describes some potential approaches to minimizing security vulnerabilities found in the operating system.

Issue	Description
Security patches	There is an increased risk that an unauthorized user may gain access to the application server if vendor security patches and upgrades are not applied in a timely fashion. Test security patches before you apply them to production servers. Also, create policies and procedures to check for and install patches on a regular basis.
Virus protection software	Virus scanners can identify infected files by scanning for a signature or watching for unusual behavior. Scanners keep their virus signatures in a file, which is usually stored on the local hard drive. Because new viruses are discovered often, you should frequently update this file for the virus scanner to identify all current viruses.
Network Time Protocol (NTP)	For forensic analysis, keep accurate time on the LiveCycle servers. Use NTP to synchronize the time on all systems that are connected directly to the Internet.

For additional security information for your operating system, see "2.1.1 Operating system security information" on page 3 .

3.2 Installation

This section describes techniques you can use during the LiveCycle installation process to reduce security vulnerabilities. In some cases, these techniques use options that are part of the installation process. The following table describes these techniques.

Issue	Description
Privileges	Use the least amount of privileges necessary to install the software. Log in to your computer by using an account that is not in the Administrators group. On Windows, you can use the Run As command to run the LiveCycle installer as an administrative user. On UNIX and Linux systems, use a command such as `sudo` to install the software.
Software source	Do not download or run LiveCycle from untrusted sources. Malicious programs can contain code to violate security in several ways, including data theft, modification and deletion, and denial of service. Install LiveCycle from the Adobe DVD or only from a trusted source.
Disk partitions	Place LiveCycle on a dedicated disk partition. Disk segmentation is a process that keeps specific data on your server on separate physical disks for added security. Arranging data in this way reduces the risk of directory traversal attacks. Plan to create a partition that is separate from the system partition on which you can install the LiveCycle content directory. (On Windows, the system partition contains the system32 directory, or boot partition.)
Components	Evaluate existing services and disable or uninstall any that are not required. Do not install unnecessary components and services. The default installation of an application server might include services that are not necessary for your use. You should disable all unnecessary services prior to deployment to minimize points of entry for an attack. For example, on JBoss, you can comment out unnecessary services in the META-INF/jboss-service.xml descriptor file.
Cross-domain policy file	The presence of a `crossdomain.xml` file on the server may immediately weaken that server. It is recommended that you make the list of domains as restrictive as possible. Do not place the `crossdomain.xml` file that was used during development into production when using Guides *(deprecated)*. For a guide that uses web services, if the service is on the same server that served up the guide, a `crossdomain.xml` file is not needed at all. But if the service is on another server, or if clusters are involved, the presence of a `crossdomain.xml` file would be needed. Refer to http://kb2.adobe.com/cps/142/tn_14213.html, for more information on the crossdomain.xml file.
Operating System security settings	If you need to use 192-bit or 256-bit XML encryption on Solaris platforms, ensure that you install `pkcs11_softtoken_extra.so` instead of `pkcs11_softtoken.so`.

3.3 Post-installation steps

After you successfully install LiveCycle, it is important to periodically maintain the environment from a security perspective.

The following section describes in detail the different tasks that are recommended to secure the deployed LiveCycle server.

3.3.1 LiveCycle security

The following recommended settings apply to the LiveCycle server outside of the administrative web application. To reduce the security risks to the server, apply these settings immediately after installing LiveCycle.

Security patches

There is an increased risk that an unauthorized user might gain access to the application server if vendor security patches and upgrades are not applied in a timely fashion. Test security patches before you apply them to production servers to ensure compatibility and availability of LiveCycle applications. Also, create policies and procedures to check for and install patches on a regular basis. LiveCycle updates are on the Enterprise products download site.

Service accounts (JBoss turnkey on Windows only)

LiveCycle installs a service, by default, by using the LocalSystem account. The built-in LocalSystem user account has a high level of accessibility; it is part of the Administrators group. If a worker-process identity runs as the LocalSystem user account, that worker process has full access to the entire system.

To run the application server on which LiveCycle is deployed, using a specific non-administrative account, follow these instructions:

1. In the Microsoft Management Console (MMC), create a local user for the LiveCycle server service to log in as:
   - Select **User cannot change password**.
   - On the **Member Of** tab, ensure that the **Users** group is listed.

   Note: *You cannot change this setting for PDF Generator.*
2. Select **Start** > **Settings** > **Administrative Tools** > **Services**.
3. Double-click the JBoss for Adobe LiveCycle 10 and stop the service.
4. On the **Log On** tab, select **This Account**, browse for the user account you created, and enter the password for the account.
5. In the MMC, open **Local Security Settings** and select **Local Policies** > **User Rights Assignment**.
6. Assign the following rights to the user account that the LiveCycle server is running under:
   - Deny log on through Terminal Services
   - Deny log on locally
   - Log on as Service (should be already set)
7. Give the new user account the Read & Execute, List Folder Contents, and Read permissions for the LiveCycle web content directories item.
8. Start the application server.

Disabling the Configuration Manager bootstrap servlet

Configuration Manager made use of a servlet deployed on your application server to perform bootstrapping of the LiveCycle database. Because Configuration Manager accesses this servlet before configuration is complete, access to it has not been secured for authorized users, and it should be disabled after you have successfully used Configuration Manager to configure LiveCycle.

1. Unzip the adobe-livecycle-[appserver].ear file.
2. Open the META-INF/application.xml file.
3. Search for the adobe-bootstrapper.war section:

```
<!-- bootstrapper start -->
<module id="WebApp_adobe_bootstrapper">
    <web>
        <web-uri>adobe-bootstrapper.war</web-uri>
        <context-root>/adobe-bootstrapper</context-root>
    </web>
</module>
<module id="WebApp_adobe_lcm_bootstrapper_redirector">
    <web>
        <web-uri>adobe-lcm-bootstrapper-redirector.war</web-uri>
        <context-root>/adobe-lcm-bootstrapper</context-root>
    </web>
</module>
<!-- bootstrapper end-->
```

4 Comment out the adobe-bootstrapper.war and the adobe-lcm-bootstrapper-redirectory. war modules as follows:

```
<!-- bootstrapper start -->
<!--
<module id="WebApp_adobe_bootstrapper">
    <web>
        <web-uri>adobe-bootstrapper.war</web-uri>
        <context-root>/adobe-bootstrapper</context-root>
    </web>
</module>
<module id="WebApp_adobe_lcm_bootstrapper_redirector">
    <web>
        <web-uri>adobe-lcm-bootstrapper-redirector.war</web-uri>
        <context-root>/adobe-lcm-bootstrapper</context-root>
    </web>
</module>
-->
<!-- bootstrapper end-->
```

5 Save and close the META-INF/application.xml file.

6 Zip the EAR file and redeploy it to the application server.

7 Type the URL into a browser to test the change and ensure that it no longer works.

Lockdown remote access to the Trust Store

Configuration Manager lets you upload a Reader Extensions 10 credential to the LiveCycle trust store. This means that access to the Trust Store Credential Service over remote protocols (SOAP and EJB) has been enabled by default. This access is no longer necessary after you have uploaded the Rights credential using Configuration Manager or if you decide to use the Administration Console later to manage credentials.

You can disable remote access to all of the Trust Store services by following the steps in the section "4.1 Disabling non-essential remote access to services" on page 34.

Disable all non-essential anonymous access

Some LiveCycle server services have operations that may be invoked by an anonymous caller. If anonymous access to these services is not required, disable it by following the steps in "4.2 Disabling non-essential anonymous access to services" on page 35.

3.3.1.1 Change the default administrator password

When LiveCycle is installed, a single default user account is configured for user Super Administrator/ login-id Administrator with a default password of *password*. You should immediately change this password using the Configuration Manager.

1 Type the following URL in a web browser:

`http://[host name]:[port]/adminui`

The default port number is one of these:

JBoss: 8080

WebLogic Server: 7001

WebSphere: 9080.

2 In the **User Name** field, type `administrator` and, in the **Password** field, type `password`.

3 Click **Settings** > **User Management** > **Users and Groups**.

4 Type `administrator` in the **Find** field, and click **Find**.

5 Click **Super Administrator** from the list of users.

6 Click **Change Password** on the Edit User page.

7 Specify the new password and click **Save**.

In addition, it is recommended to change the default password for CRX Administrator by performing the following steps:

1 Log into http://[*host name*]:[*port*]/lc/libs/granite/security/content/admin.html using the default username/password.

2 Type Administrator in the search field and click **Go**.

3 Select **Administrator** from the search result and click the **Edit** icon at the bottom-right of the user interface.

4 Specify the new password in the **New Password** field and the old password in the **Your Password** field.

5 Click the Save icon at the bottom-right of the user interface.

3.3.1.2 Disable WSDL generation

Web Service Definition Language (WSDL) generation should be enabled only for development environments, where WSDL generation is used by developers to build their client applications. You may choose to disable WSDL generation in a production environment to avoid exposing a service's internal details.

1 Type the following URL in a web browser:

`http://[host name]:[port]/adminui`

2 Click **Settings > Core System Settings > Configurations**.

3 Uncheck **Enable WSDL** and click **OK**.

3.3.1.3 Restricting LiveCycle Content Services (deprecated) user data check-in quotas

Note: *Adobe is migrating Adobe® LiveCycle® Content Services ES customers to the Content Repository built on the modern, modular CRX architecture, acquired during the Adobe acquisition of Day Software. The Content Repository is provided with LiveCycle Foundation and is available as of the LiveCycle ES4 release.*

By default, Content Services does not restrict on the amount of data a user can check in to the server at any one time. Large amounts of data are potentially threatening to the system as they leave the system without the resources to perform other operations. This situation can cause a denial of service to other incoming processes. Use JVM arguments to enable quota management in Content Services.

Important: *These JVM arguments must be passed prior to synchronizing the users. This user quota cannot be modified once the users have been synchronized.*

3.3.1.3.1 Enable quota management on Content Services:

On JBoss

1. Navigate to the *[jboss root]*/bin directory and open the startup script in a text editor:
   - **(Windows)** run.bat
   - **(Linux and UNIX)** run.sh
2. Add the following properties below the `Set JAVA_OPTS` argument:

   `-Dsystem.usages.enableQuotaSize=true -Dsystem.usages.quota=`**[size in KB]**
3. Save and close the file.
4. Restart the JBoss server before synchronizing the users.

On WebLogic

1. Access the WebLogic Server Administration Console, type http://*[host name]*:*[port]*/console in the URL line of a web browser, where *[port]* is the non-secure listening port. By default, this port value is 7001.
2. On the login screen, type your WebLogic user name and password and click **Log In**.
3. Under Change Center, click **Lock & Edit**.
4. Under Domain Structure, click **Environment** > **Servers** and, in the right pane, click the managed server name.
5. In the Settings for Server pane, click the **Configurationtab** > **Server Start tab**.
6. In the Arguments box, add the following arguments separated by a space delimiter:

   ```
   -Dsystem.usages.enableQuotaSize=true
   -Dsystem.usages.quota=[size in KB]
   ```
7. Click **Save** and then click **Activate Changes**.
8. Restart the WebLogic server before synchronizing the users.

On WebSphere

1. In the WebSphere Administrative Console navigation tree, do the following task for your application server:

   (WebSphere 6.x) Click **Servers** > **Application servers**

   (WebSphere 7.x) Click **Servers** > **ServerTypes** > **WebSphere application servers**
2. Click the server name in the right pane.
3. Under Server Infrastructure, click **Java and Process Management** > **Process Definition**.
4. Under Additional Properties, click **Java Virtual Machine**.
5. In the **Generic JVM arguments** box, add `-Dsystem.usages.enableQuotaSize=true` and `-Dsystem.usages.quota=`<*size in KB*>, separated by commas, to the existing properties.
6. Click **OK** or **Apply**, and then click **Save directly to the master configuration**.

7 Restart the WebSphere server before synchronizing the users.

3.3.2 Application server security

The following table describes some techniques for securing your application server after the LiveCycle application is installed.

Issue	Description
Application server administrative console	After you install, configure, and deploy LiveCycle on your application server, you should disable access to the application server administrative consoles. See your application server documentation for details.
Application server cookie settings	Application cookies are controlled by the application server. When deploying the application, the application server administrator can specify cookie preferences on a server-wide or application-specific basis. By default, the server settings take preference. All session cookies generated by your application server should include the `HttpOnly` attribute. For example, when using the JBoss Application Server, you can modify the SessionCookie element to `httpOnly="true"` in the `deploy/jbossweb.sar/context.xml` file. You can restrict cookies to be sent using HTTPS-only. As a result, they are not sent unencrypted over HTTP. Application server administrators should enable secure cookies for the server on a global basis. For example, when using the JBoss Application Server, you can modify the connector element to `secure=true` in the `server.xml` file. See your application server documentation for more details on cookie settings.
Directory browsing	When someone requests a page that does not exist or requests the name of a director (the request string ends with a forward slash (/)), the application server should not return the contents of that directory. To prevent this, you can disable directory browsing on your application server. You should do this for the Administration Console application and for other applications running on your server. For JBoss, set the value of the listings initialization parameter of the `DefaultServlet` property to `false` in the web.xml file, as shown by this example: <servlet> <servlet-name>default</servlet-name> <servlet-class> org.apache.catalina.servlets.DefaultServlet </servlet-class> <init-param> <param-name>listings</param-name> <param-value>false</param-value> </init-param> <load-on-startup>1</load-on-startup> </servlet> For WebSphere, set the `directoryBrowsingEnabled` property in the ibm-web-ext.xmi file to `false`. For WebLogic, set the index-directories properties in the weblogic.xml file to `false`, as shown by this example: <container-descriptor> <index-directory-enabled>false </index-directory-enabled> </container-descriptor>

3.3.3 Using JMX Console on JBoss

When the Java Management Extensions (JMX) console is installed with JBoss, URLs can be constructed for use as cross-site scripting (XSS) exploits that can reveal sensitive information about your system.

If you installed LiveCycle by using the turnkey method and are using the version of JBoss that was included with the turnkey installation, the JBoss JMX Console is removed by default to ensure that security risks are minimized. However, if you need to use the JBoss JMX Console, reinstall it by following this procedures.

1. Download a copy of JBoss 4.2.0 (or later) from JBoss.org.
2. Stop the JBoss Application Server.
3. From the zipped archive file you downloaded, extract the files from *[JBoss root]*/deploy/jmx-console.war/.
4. Place the jmx-console.war/... files in the deploy directory of the JBoss installation directory.
5. Restart JBoss.
6. Go to the following URL to ensure that the JBoss JMX Console is available:

   http://localhost:8080/jmx-console

3.3.4 Database security

When securing your database, you should implement the measures described by your database vendor. You should allocate a database user with the minimum required database permissions granted for use by LiveCycle. For example, do not use an account with database administrator privileges.

On Oracle, the database account that you use needs only the CONNECT, RESOURCE, and CREATE VIEW privileges. For similar requirements on other databases, see Preparing to Install LiveCycle (Single Server).

3.3.4.1 Configuring integrated security for SQL Server on Windows for JBoss

1. Modify *[JBOSS_HOME]*\server\all\deploy\adobe-ds.xml to add `integratedSecurity=true` to the connection URL, as shown in this example:

   ```
   jdbc:sqlserver://<serverhost>:<port>;databaseName=<dbname>;integratedSecurity=true
   ```

2. Add the sqljdbc_auth.dll file to the Windows systems path on the computer that is running the application server. The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.2 driver installation (the default is *[InstallDir]*/sqljdbc_1.2/enu/auth/x86).
3. Modify JBoss Windows service (JBoss for LiveCycle) property for Log On As from Local System to a login account that has LiveCycle database and a minimum set of privileges. If you are running JBoss from the command line instead of as a Windows service, you do not need to perform this step.
4. Set Security for SQL Server from **Mixed** mode to **Windows Authentication only**.

3.3.4.2 Configuring integrated security for SQL Server on Windows for WebLogic

1. Start the WebLogic Server Administration Console by typing the following URL in the URL line of a web browser:

   ```
   http://[host name]:7001/console
   ```

2. Under Change Center, click **Lock & Edit**.
3. Under Domain Structure, click *[base_domain]*> **Services** > **JDBC** > **Data Sources** and, in the right pane, click **IDP_DS**.
4. On the next screen, on the **Configuration** tab, click the **ConnectionPool** tab and, in the **Properties** box, type `integratedSecurity=true`.

5 Under Domain Structure, click **[base_domain]** > **Services** > **JDBC** > **Data Sources** and, in the right pane, click **RM_DS**.

6 On the next screen, on the **Configuration** tab, click the **Connection Pool** tab and, in the **Properties** box, type `integratedSecurity=true`.

7 Add the sqljdbc_auth.dll file to the Windows systems path on the computer that is running the application server. The sqljdbc_auth.dll file is located with the Microsoft SQL JDBC 1.2 driver installation (the default is *[InstallDir]*/sqljdbc_1.2/enu/auth/x86).

8 Set Security for SQL Server from **Mixed** mode to **Windows Authentication only**.

3.3.4.3 Configuring integrated security for SQL Server on Windows for WebSphere

On WebSphere, you can configure integrated security only when you use an external SQL Server JDBC driver, not the SQL Server JDBC driver that is embedded with WebSphere.

1 Log in to the WebSphere Administrative Console.

2 In the navigation tree, click **Resources** > **JDBC** > **Data Sources** and, in the right pane, click **IDP_DS**.

3 In the right pane, under Additional Properties, click **Custom Properties**, and then click **New**.

4 In the **Name** box, type `integratedSecurity` and, in the **Value** box, type `true`.

5 In the navigation tree, click **Resources** > **JDBC** > **Data Sources** and, in the right pane, click **RM_DS**.

6 In the right pane, under Additional Properties, click **Custom Properties**, and then click **New**.

7 In the **Name** box, type `integratedSecurity` and, in the **Value** box, type `true`.

8 On the computer where WebSphere is installed, add the sqljdbc_auth.dll file to the Windows systems path (C:\Windows). The sqljdbc_auth.dll file is in the same location as the Microsoft SQL JDBC 1.2 driver installation (default is *[InstallDir]*/sqljdbc_1.2/enu/auth/x86).

9 Select **Start** > **Control Panel** > **Services**, right-click the Windows service for WebSphere (IBM WebSphere Application Server <version> - <node>) and select **Properties**.

10 In the Properties dialog box, click the **Log On** tab.

11 Select **This Account** and provide the information required to set the login account you want to use.

12 Set Security on SQL Server from **Mixed** mode to **Windows Authentication only**.

3.3.5 Protecting access to sensitive content in the database

The LiveCycle database schema contains sensitive information about system configuration and business processes and should be hidden behind the firewall. The database should be considered within the same trust boundary as the LiveCycle server. To guard against information disclosure and theft of business data, the database must be configured by the database administrator (DBA) to allow access only by authorized administrators.

As an added precaution, you should consider using database vendor-specific tools to encrypt columns in tables that contain the following data:

- Rights Management Document Keys
- Trust Store HSM PIN encryption key
- Local User Password Hashes

For information about vendor-specific tools, see "2.1.3 Database security information" on page 4 .

3.3.6 LDAP security

A Lightweight Directory Access Protocol (LDAP) directory is typically used by LiveCycle as a source for enterprise user and group information, and a means to perform password authentication. You should ensure that your LDAP directory is configured to use Secure Socket Layer (SSL) and that LiveCycle is configured to access your LDAP directory using its SSL port.

3.3.6.1 LDAP denial of service

A common attack using LDAP involves an attacker deliberately failing to authenticate multiple times. This forces the LDAP Directory Server to lock out a user from all LDAP-reliant services.

You can set the number of failure attempts and subsequent lock-out time that LiveCycle implements when a user repeatedly fails to authenticate to LiveCycle. In Administration Console, choose low values. When selecting the number of failure attempts, it is important to understand that after all attempts are made, LiveCycle locks out the user before the LDAP Directory Server does.

3.3.6.2 Set automatic account locking

1 Log in to Administration Console.

2 Click **Settings** > **User Management** > **Domain Management**.

3 Under Automatic Account Locking Settings, set **Maximum Consecutive Authentication Failures** to a low number, such as 3.

4 Click **Save**.

3.3.7 Auditing and logging

The proper and secure use of application auditing and logging can help ensure that security and other anomalous events are tracked and detected as quickly as possible. Effective use of auditing and logging within an application includes such items as tracking successful and failed logins, as well as key application events such as the creation or deletion of key records.

You can use auditing to detect many types of attacks, including these:

- Brute force password attacks
- Denial of service attacks
- Injection of hostile input and related classes of scripting attacks

This table describes auditing and logging techniques you can use to reduce your server's vulnerabilities.

Issue	Description
Log file ACLs	Set appropriate LiveCycle log file access control lists (ACLs). Setting the appropriate credentials helps prevent attackers from deleting the files. The security permissions on the log file directory should be Full Control for Administrators and SYSTEM groups. The LiveCycle user account should have Read and Write permissions only.
Log file redundancy	If resources permit, send logs to another server in real time that is not accessible by the attacker (write only) by using Syslog, Tivoli, Microsoft Operations Manager (MOM) Server, or another mechanism. Protecting logs this way helps prevent tampering. Also, storing logs in a central repository aids in correlation and monitoring (for example, if multiple LiveCycle servers are in use and a password-guessing attack is taking place across multiple computers where each computer is queried for a password).

3.3.8 LiveCycle Unix system library dependencies

The following information is intended to help you plan for a LiveCycle deployment on a UNIX environment.

3.3.8.1 Convert PDF service

The Convert PDF service that is part of LiveCycle requires the following minimum system libraries:

Linux

```
/lib/
    libdl.so.2 (0x00964000)
    ld-linux.so.2 (0x007f6000)
/lib/tls/
    libc.so.6 (0x00813000)
    libm.so.6 (0x0093f000)
    libpthread.so.0 (0x00a5d000)
/usr/lib/libz.so.1 (0x0096a000)
/gcc410/lib/
    libgcc_s.so.1 (0x00fc0000)
    libstdc++.so.6 (0x00111000)
```

Solaris

```
/usr/platform/SUNW,Sun-Fire-V210/lib/libc_psr.so.1
/usr/lib/
    libc.so.1
    libdl.so.1
    libintl.so.1
    libm.so.1
    libmp.so.2
    libnsl.so.1
    libpthread.so.1
    libsocket.so.1
    libstdc++.so.6
    libthread.so.1
```

AIX

```
/usr/lib/
    libpthread.a(shr_comm.o)
    libpthread.a(shr_xpg5.o)
    libc.a(shr.o)
    librtl.a(shr.o)
    libpthreads.a(shr_comm.o)
    libcrypt.a(shr.o)
/aix5.2/lib/gcc/powerpc-ibm-aix5.2.0.0/4.1.0/libstdc++.a(libstdc++.so.6)
/aix5.2/lib/gcc/powerpc-ibm-aix5.2.0.0/4.1.0/libgcc_s.a(shr.o)
```

3.3.8.2 XMLForms

XMLForms requires the following minimum system libraries:

Linux

```
/lib/
    libdl.so.2
    libpthread.so.0
    libm.so.6
    libgcc_s.so.1
    libc.so.6
    librt.so.1
    ld-linux.so.2
/usr/X11R6/lib/
    libX11.so.6
```

Solaris

```
/usr/lib/
    libdl.so.1
    libpthread.so.1
    libintl.so.1
    libsocket.so.1
    libnsl.so.1
    libm.so.1
    libc.so.1
    librt.so.1
    libX11.so.4
    libmp.so.2
    libmd5.so.1
    libscf.so.1
    libaio.so.1
    libXext.so.0
    libdoor.so.1
    libuutil.so.1
    libm.so.2
usr/platform/SUNW,Sun-Fire-V210/lib/libc_psr.so.1
usr/platform/SUNW,Sun-Fire-V210/lib/libmd5_psr.so.1
```

AIX 6.1

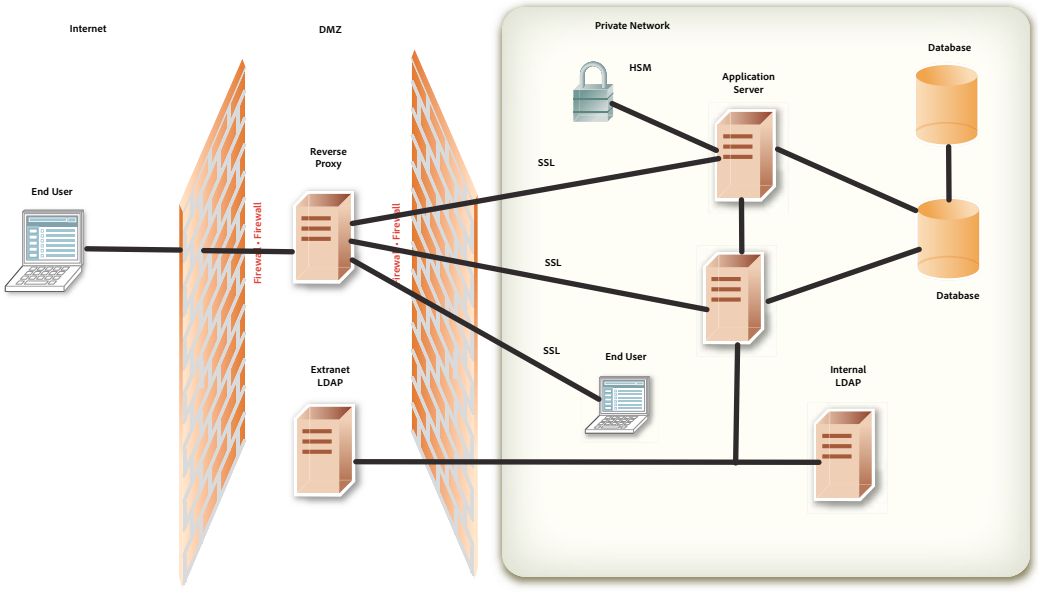
```
/usr/lib/
    libpthread.a(shr_comm.o)
    libpthread.a(shr_xpg5.o)
    libc.a(shr.o)
    librtl.a(shr.o)
    libdl.a(shr.o)
    libX11.a(shr4.o)
    libiconv.a(shr4.o)
    libpthreads.a(shr_comm.o)
/unix
    /usr/lib/libcrypt.a(shr.o)
    /usr/lib/libIM.a(shr.o)
    /usr/lib/libpthreads.a(shr_xpg5.o)
```

3.4 Configuring LiveCycle for access beyond the enterprise

After you successfully install LiveCycle, it is important to periodically maintain the security of your environment. This section describes the tasks that are recommended to maintain the security of your LiveCycle production server.

3.4.1 Setting up a reverse proxy for web access

A *reverse proxy* can be used to ensure that one set of URLs for LiveCycle web applications are available to both external and internal users. This configuration is more secure than allowing users to connect directly to the application server that LiveCycle is running on. The reverse proxy performs all HTTP requests for the application server that is running LiveCycle. Users have only network access to the reverse proxy and can only attempt URL connections that are supported by the reverse proxy.



LiveCycle root URLs for use with reverse proxy server

The following application root URLs for each LiveCycle web application. You should configure your reverse proxy only to expose URLs for web application functionality that you want to provide to end users.

Certain URLs are highlighted as end-user-facing web applications. You should avoid exposing other URLs for Configuration Manager for access to external users through the reverse proxy.

Root URL	Purpose and/or associated web application	Web-based interface	End-user access
/ReaderExtensions/*	Reader Extensions end-user web application for applying usage rights to PDF documents	Yes	Yes
/edc/*	Rights Management end-user web application	Yes	Yes
/edcws/*	Web service URL for Rights Management	No	Yes
/pdfgui/*	PDF Generator administration web application	Yes	Yes

Root URL	Purpose and/or associated web application	Web-based interface	End-user access
/workspace/*	Workspace end-user web application	Yes	Yes
/workspace-server/*	Workspace servlets and data services that the Workspace client application requires	Yes	Yes
/contentspace/*	LiveCycle Contentspace (deprecated) end-user web application	Yes	Yes
/adobe-bootstrapper/*	Servlet for bootstrapping the LiveCycle repository	No	No
/soap/*	Information page for LiveCycle server web services	No	No
/soap/services/*	Web service URL for all LiveCycle server services	No	No
/edc/admin/*	Rights Management administration web application	Yes	No
/adminui/*	Administration Console home page	Yes	No
/TruststoreComponent/ secured/*	Trust Store Management administration pages	Yes	No
/FormsIVS/*	Forms IVS application for testing and debugging form rendering	Yes	No
/OutputIVS/*	Output IVS application for testing and debugging output service	Yes	No
/rmws/*	REST URL for Rights Management	No	Yes
/OutputAdmin/*	Output administration pages	Yes	No
/FormServer/*	Forms web application files	Yes	No
/FormServer/GetImage Servlet	Used for fetching JavaScript during HTML transformation	No	No
/FormServerAdmin/*	Forms administration pages	Yes	No
/repository/*	URL for WebDAV (debugging) access	Yes	No
/AACComponent/*	Applications and Services user interface	Yes	No
/WorkspaceAdmin/*	Workspace administration pages	Yes	No
/rest/*	Rest support pages	Yes	No
/CoreSystemConfig/*	LiveCycle Core Configuration settings page	Yes	No
/um/	User Management authentication	No	Yes
/um/*	User Management administration interface	Yes	No
/DoumentManager/*	Uploading and downloading of documents that are to be processed when accessing (Deprecated for AEM forms) remoting endpoints, SOAP WSDL endpoints, and the Java SDK over SOAP transport or EJB transport with HTTP documents enabled.	Yes	Yes
/remoting/*	Adding a (Deprecated for AEM forms) Remoting endpoint enables a Flex application to invoke the service using (Deprecated for AEM forms) LiveCycle Remoting.	Yes	Yes

3.5 Protecting from Cross-Site Request Forgery attacks

A Cross-Site Request Forgery (CSRF) attack exploits the trust that a web site has for the user, to transmit commands that are unauthorized and unintended by the user. The attack is set up by including a link or a script in a web page, or a URL in an e-mail message, to access another site to which the user has already been authenticated.

For example, you may be logged in to Administration Console while simultaneously browsing another website. One of the web pages may include an HTML image tag with a `src` attribute that targets a server-side script on the victim website. By leveraging the cookie-based session-authentication mechanism provided by web browsers, the attacking website can send malicious requests to this victim server-side script, masquerading as the legitimate user. For more examples, see https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)#Examples.

The following characteristics are common to CSRF:

- Involve sites that rely on a user's identity.
- Exploit the site's trust in that identity.
- Trick the user's browser into sending HTTP requests to a target site.
- Involve HTTP requests that have side effects.

LiveCycle uses the Referer Filter feature to block CSRF attacks. The following terms are used in this section to describe the Referer Filtering mechanism:

- **Allowed Referer:** A Referer is the address of the source page that sends a request to the server. For JSP pages or forms, the Referer is usually the previous page in the browsing history. Referers for images are usually the pages on which the images are displayed. You can identify the Referers that are allowed access to your server resources by adding them to the Allowed Referer list.
- **Allowed Referer Exceptions:** You may want to restrict the scope of access for a particular Referer in your Allowed Referer list. To enforce this restriction you can add individual paths of that Referer to the Allowed Referer Exceptions list. Requests originating from paths in the Allowed Referer Exceptions list are prevented from invoking any resource on the LiveCycle server. You can define Allowed Referer Exceptions for a specific application and also use a global list of exceptions that apply to all applications.
- **Allowed URIs:** This is a list of resources that are to be served without checking the Referer Header. Resources, for example, help pages, that do not result in state changes on the server, can be added to this list. The resources in the Allowed URIs list are never blocked by the Referer Filter irrespective of who the Referer is.
- **Null Referer:** A server request that is not associated with or does not originate from a parent web page is considered to be a request from a Null Referer. For example, when you open a new browser window, type an address, and press enter, the Referer sent to the server is null. A desktop application (.NET or SWING) making an HTTP request to a web server, also sends a Null Referer to the server.

3.5.1 Referer Filtering

The Referer Filtering process can be described as follows:

1. The LiveCycle server checks the HTTP method used for invocation:
   a. If it is POST, the LiveCycle server performs the Referer header check.
   b. If it is GET, the LiveCycle server bypasses the Referer check, unless *CSRF_CHECK_GETS* is set to true, in which case it performs the Referer header check. *CSRF_CHECK_GETS* is specified in the *web.xml* file for your application.

2 The LiveCycle server checks whether the requested URI is whitelisted:

   a If the URI is whitelisted, the server accepts the request.

   b If the requested URI is not whitelisted, the server retrieves the Referer of the request.

3 If there is a Referer in the request, the server checks whether it is an Allowed Referer. If it is allowed, the server checks for a Referer Exception:

   a If it is an exception, the request is blocked.

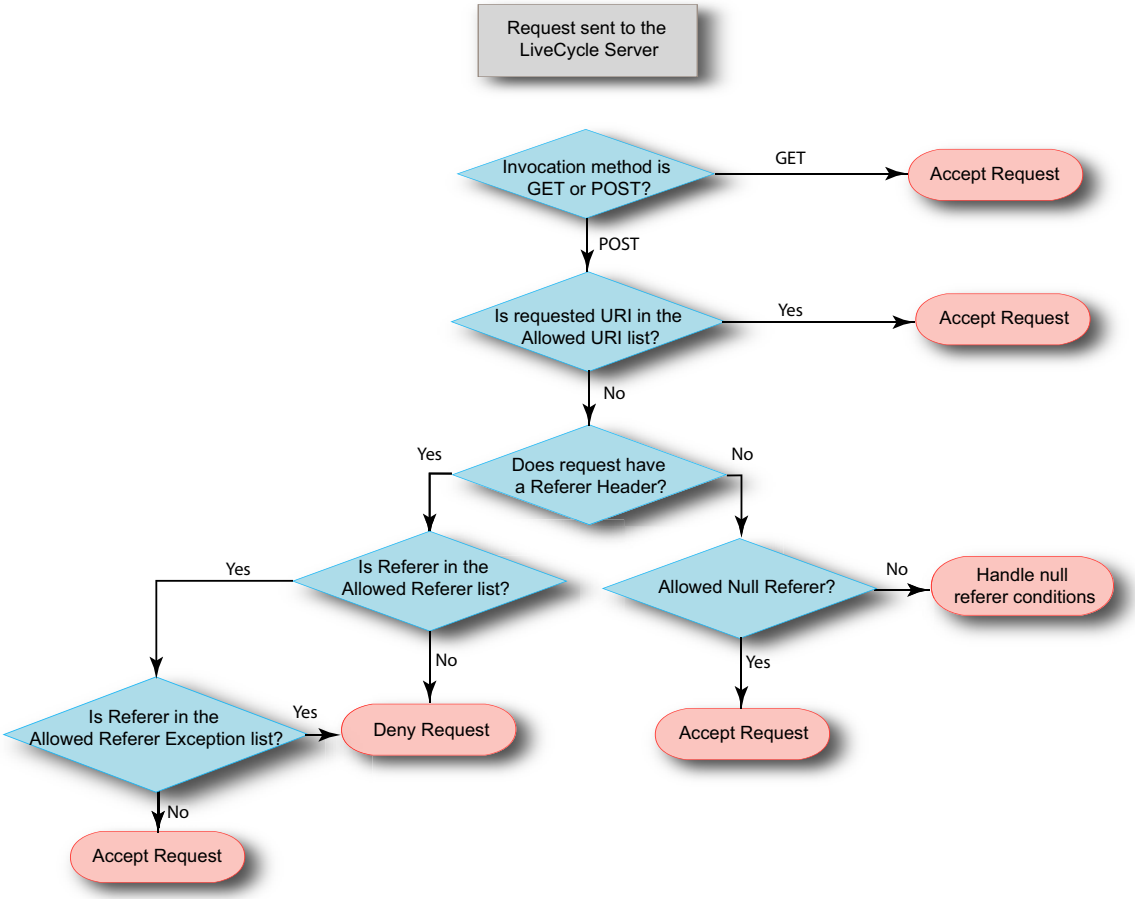
   b If it is not an exception, the request is passed.

4 If there is no Referer in the request, the server checks whether a Null Referer is allowed:

   a If a Null Referer is allowed, the request is passed.

   b If a Null Referer is not allowed, the server checks whether the requested URI is an exception for the Null Referer and handles the request accordingly.

The following depicts the CSRF check that LiveCycle performs when a request is sent to the server.



3.5.2 Managing Referer Filtering

LiveCycle provides a Referer Filter to specify Referers that are allowed access to your server resources. By default, the Referer filter does not filter requests that use a safe HTTP method, e.g. GET, unless *CSRF_CHECK_GETS* is set to true. If the port number for an Allowed Referer entry is set to 0, LiveCycle will allow all requests with Referers from that host regardless of the port number. If no port number is specified, only requests from the default port 80 (HTTP) or port 443 (HTTPS) are allowed. Referer Filtering is disabled if all the entries in the Allowed Referer list are deleted.

When you first install Document Services, the Allowed Referer list is updated with the address of the server on which Document Services is installed. The entries for the server include the server name, the IPv4 address, the IPv6 address if IPv6 is enabled, the loopback address, and a localhost entry. The names added to the Allowed Referer list are returned by Host operating system.For example a server with an IP address of 10.40.54.187 will include the following entries: `http://server-name:0, https://10.40.54.187:0, http://127.0.0.1:0, http://localhost:0`. For any unqualified name retuned by Host operating system (names that do not have IPv4 address, IPv6 address or qualified domain name) white list is not updated. Modify the Allowed Referer list to suit your business environment. Do not deploy the LiveCycle server in the production environment with the default Allowed Referer list. After modifying any of the Allowed Referers, Referer Exceptions, or URIs, ensure that you restart the server for the changes to take effect.

Managing Allowed Referer list

You can manage the Allowed Referer list from the User Management Interface of Administration Console. The User Management Interface provides you with the functionality to create, edit, or delete the list. Refer to the *Preventing CSRF attacks* section of the *Administration Help* for more information on working with the Allowed Referer list.

Managing Allowed Referer Exception and Allowed URI lists

LiveCycle provides APIs to manage the Allowed Referer Exception list and the Allowed URI list. You can use these APIs to retrieve, create, edit, or delete the list. Following is a list of available APIs:

- createAllowedURIsList
- getAllowedURIsList
- updateAllowedURIsList
- deleteAllowedURIsList
- addAllowedRefererExceptions
- getAllowedRefererExceptions
- updateAllowedRefererExceptions
- deleteAllowedRefererExceptions

Refer to the *LiveCycle API Reference* for more information on the APIs.

Use the ***LC_GLOBAL_ALLOWED_REFERER_EXCEPTION*** list for Allowed Referer Exceptions at the global level i.e. to define exceptions that are applicable to all applications. This list contains only URIs with either an absolute path (e.g. `/index.html`) or a relative path (e.g. `/sample/`). You can also append a regular expression to the end of a relative URI, e.g. `/sample/(.)*`.

The ***LC_GLOBAL_ALLOWED_REFERER_EXCEPTION*** list ID is defined as a constant in the `UMConstants` class of the `com.adobe.idp.um.api` namespace, found in `adobe-usermanager-client.jar`. You can use the LiveCycle APIs to create, modify, or edit this list. For example, to create the Global Allowed Referer Exceptions list use:

```
addAllowedRefererExceptions(UMConstants.LC_GLOBAL_ALLOWED_REFERER_EXCEPTION,
Arrays.asList("/index.html", "/sample/(.)*"))
```

Use the ***CSRF_ALLOWED_REFERER_EXCEPTIONS*** list for application-specific exceptions.

Disabling the Referer Filter

In the event that the Referer Filter completely blocks access to the LiveCycle server and you cannot edit the Allowed Referer list, you can update the server startup script and disable Referer Filtering.

Include the `-Dlc.um.csrffilter.disabled=true` JAVA argument in the startup script and restart the server. Ensure that you delete the JAVA argument after you have appropriately reconfigured the Allowed Referer list.

Referer Filtering for Custom WAR files

You may have created custom WAR files to work with LiveCycle in order to meet your business requirements. To enable Referer Filtering for your custom WAR files, include ***adobe-usermanager-client.jar*** in the class path for the WAR and include a filter entry in the *web.xml* file with the following parameters:

CSRF_CHECK_GETS controls the Referer check on GET requests. If this parameter is not defined, the default value is set to false. Include this parameter only if you want to filter your GET requests.

CSRF_ALLOWED_REFERER_EXCEPTIONS is the ID of the Allowed Referer Exceptions list. The Referer Filter prevents requests originating from Referers in the list identified by the list ID, from invoking any resource on the LiveCycle server.

CSRF_ALLOWED_URIS_LIST_NAME is the ID of the Allowed URIs list. The Referer Filter does not block requests for any of the resources in the list identified by the list ID, regardless of the value of the Referer header in the request.

CSRF_ALLOW_NULL_REFERER controls the Referer Filter behavior when the Referer is null or not present. If this parameter is not defined, the default value is set to false. Include this parameter only if you want to allow Null Referers. Allowing null referers may allow some types of Cross Site Request Forgery attacks.

CSRF_NULL_REFERER_EXCEPTIONS is a list of the URIs for which a Referer check is not performed when the Referer is null. This parameter is enabled only when *CSRF_ALLOW_NULL_REFERER* is set to false. Separate multiple URIs in the list with a comma.

Following is an example of the filter entry in the *web.xml* file for a ***SAMPLE*** WAR file:

```
<filter>
       <filter-name> filter-name </filter-name>
       <filter-class> com.adobe.idp.um.auth.filter.RemoteCSRFFilter </filter-class>
     <!-- default is false -->
     <init-param>
      <param-name> CSRF_ALLOW_NULL_REFERER </param-name>
      <param-value> false </param-value>
     </init-param>
     <!-- default is false -->
     <init-param>
      <param-name> CSRF_CHECK_GETS </param-name>
      <param-value> true </param-value>
     </init-param>
     <!-- Optional -->
     <init-param>
       <param-name> CSRF_NULL_REFERER_EXCEPTIONS </param-name>
       <param-value> /SAMPLE/login, /SAMPLE/logout  </param-value>
     </init-param>
     <!-- Optional -->
     <init-param>
      <param-name> CSRF_ALLOWED_REFERER_EXCEPTIONS </param-name>
      <param-value> SAMPLE_ALLOWED_REF_EXP_ID </param-value>
     </init-param>
     <!-- Optional -->
     <init-param>
      <param-name> CSRF_ALLOWED_URIS_LIST_NAME </param-name>
      <param-value> SAMPLE_ALLOWED_URI_LIST_ID </param-value>
     </init-param>
</filter>
    ........
    <filter-mapping>
      <filter-name> filter-name </filter-name>
      <url-pattern>/*</url-pattern>
    </filter-mapping>
```

Troubleshooting

If legitimate server requests are being blocked by the CSRF filter, try one of the following:

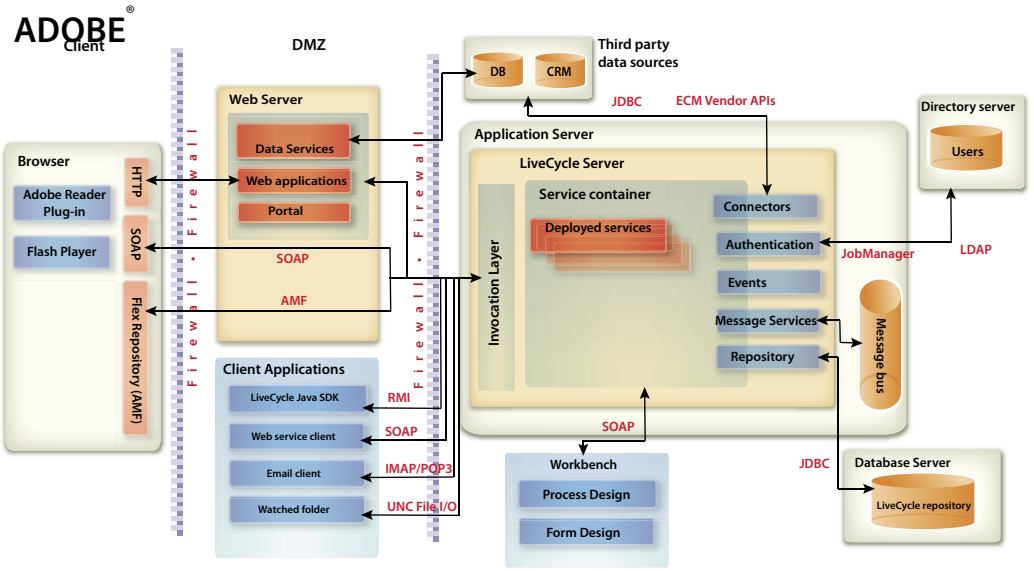
- If the rejected request has a Referer header, carefully consider adding it to the Allowed Referer list. Add only Referers that you trust.
- If the rejected request does not have a Referer header, modify your client application to include a Referer header.
- If the client can work in a browser, try that deployment model.
- As a last resort you can add the resource to the Allowed URIs list. This is not a recommended setting.

3.6 Secure network configuration

This section describes the protocols and ports that are required by LiveCycle and provides recommendations for deploying LiveCycle in a secure network configuration.

3.6.1 LiveCycle physical architecture

This image shows the components and protocols that are used in a typical LiveCycle deployment, including the appropriate firewall topology.



3.6.2 Network protocols used by LiveCycle

When you configure a secure network architecture as described in the previous section, the following network protocols are required for interaction between LiveCycle and other systems in your enterprise network.

Protocol	Use
HTTP	• Browser displays Configuration Manager and end-user web applications • All SOAP connections
SOAP	• Web service client applications, such as .NET applications • Adobe Reader® uses SOAP for LiveCycle server web services • Adobe Flash® applications uses SOAP for LiveCycle server web services • LiveCycle SDK calls when used in SOAP mode • Workbench design environment
RMI	LiveCycle SDK calls when used in Enterprise JavaBeans (EJB) mode
IMAP / POP3	• Email-based input to a service (Email endpoint) • User task notifications over email
UNC File IO	LiveCycle monitoring of watched folders for input to a service (watched folder endpoint)
LDAP	• Synchronizations of organizational user and group information in a directory • LDAP authentication for interactive users

Protocol	Use
JDBC	• Query and procedure calls made to an external database during execution of a process using the JDBC service • Internal access LiveCycle repository
WebDAV	Enables remote browsing of the LiveCycle design-time repository (forms, fragments, and so on) by any WebDAV client
AMF	Adobe Flash applications, where LiveCycle server services are configured with a (Deprecated for AEM forms) Remoting endpoint
JMX	LiveCycle exposes MBeans for monitoring using JMX

3.6.3 Ports for application servers

This section describes the default ports (and alternate configuration ranges) for each type of application server supported. These ports must be enabled or disabled on the inner firewall, depending on the network functionality you want to allow for clients that connect to the application server running LiveCycle.

Note: *By default, the server exposes several JMX MBeans under the adobe.com namespace. Only information that is useful for server health monitoring is exposed. However, to prevent information disclosure, you should prevent callers in an untrusted network from looking up JMX MBeans and accessing health metrics.*

JBoss ports

Purpose	Port
Access to web applications	[JBoss root]/server/all/deploy/jbossweb-tomcat50.sar/server.xml HTTP/1.1 Connector port 8080 AJP 1.3 Connector port 8009 SSL/TLS Connector port 8443
Access to LiveCycle server services	[JBoss root]/server/all/conf/jboss-service.xml WebService port 8083 NamingService Port 1099 RMIport from 1098 RMIObjectPort from 4444 PooledInvoker ServerBindPort 4445

Purpose	Port
J2EE cluster support	[JBoss root]/server/all/deploy/cluster-service.xml ha.jndi.HANamingService port from 1100 RmiPort 1101 RMIObjectPort 4447 (clusters only) ServerBindPort 4446
CORBA support	[JBoss root]/server/all/conf/jacorb.properties OAPort 3528 OASSLPort 3529
SNMP support	[JBoss root]/server/all/deploy/snmp-adaptor.sar/META-INF/jbossservice.xml ports 1161, 1162 [JBoss root]/server/all/deploy/snmp-adaptor.sar/managers.xml port 1162

WebLogic ports

Purpose	Port
Access to web applications	• Admin Server listen port: default is 7001 • Admin Server SSL listen port: default is 7002 • Port configured for Managed Server, for example 8001
WebLogic administration ports not required for access to LiveCycle	• Managed Server listen port: Configurable from 1 to 65534 • Managed Server SSL listen port: Configurable from 1 to 65534 • Node Manager listen port: default is 5556

WebSphere 6.1 ports

For information about WebSphere 6.1ports that LiveCycle requires, go to Port number settings in WebSphere Application Server versions.

WebSphere 7.0 ports

For information about WebSphere 7.0 ports that LiveCycle requires, go to http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.migration.express.doc/info/exp/ae/rmig_portnumber.html.

3.6.4 Configuring SSL

Referring to the physical architecture that is described in the section "3.6.1 LiveCycle physical architecture" on page 28, you should configure SSL for all of the connections that you plan to use. Specifically, all SOAP connections must be conducted over SSL to prevent exposure of user credentials on a network.

For instructions on how to configure SSL on JBoss, WebLogic, and WebSphere, see "Configuring SSL" in the LiveCycle Administration Help.

3.6.5 Configuring SSL redirect

After you configure your application server to support SSL, you must ensure that all HTTP traffic to LiveCycle applications and services are enforced to use the SSL port.

To configure SSL redirect for WebSphere or WebLogic, see your application server documentation.

1 Navigate to the adobe-livecycle-jboss.ear and unzip it.

2 Extract the adminui.war file and open the web.xml file for editing.

3 Add the following code to the web.xml file:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>app or resource name</web-resource-name>
        <url-pattern>/*</url-pattern>
        <!-- define all url patterns that need to be protected-->
        <http-method>GET</http-method>
        <http-method>POST</http-method>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

3.7 Windows-specific security recommendations

This section contains security recommendations that are specific to Windows when used to run LiveCycle.

3.7.1 JBoss Service accounts

The LiveCycle turnkey installation sets up a service account, by default, using the Local System account. The built-in Local System user account has a high level of accessibility; it is part of the Administrators group. If a worker process identity runs as the Local System user account, that worker process has full access to the entire system.

3.7.1.1 Run the application server using a non-administrative account

1 In the Microsoft Management Console (MMC), create a local user for the LiveCycle server service to log in as:

- Select **User cannot change password**.
- On the **Member Of** tab, ensure that the Users group is listed.

2 Select **Settings** > **Administrative Tools** > **Services**.

3 Double-click the application server service and stop the service.

4 On the **Log On** tab, select **This Account**, browse for the user account you created, and enter the password for the account.

5 In the Local Security Settings window, under User Rights Assignment, give the following rights to the user account that the LiveCycle server is running under:

- Deny log on through Terminal Services
- Deny log on locally
- Log on as Service (should be already set)

6 Give the new user account Read & Execute, List Folder Contents, and Read permissions to LiveCycle web content directories.

7 Start the application server service.

3.7.2 File system security

LiveCycle uses the file system in the following ways:

- Stores temporary files that are used while processing document input and output
- Stores files in the global archive store that are used to support the solution components that are installed
- Watched folders store dropped files that are used as input to a service from a file system folder location

When using watched folders as a way to send and receive documents with a LiveCycle server service, take extra precautions with file system security. When a user drops content in the watched folder, that content is exposed through the watched folder. In this case, the service does not authenticate the actual end user. Instead, it relies on ACL and Share level security to be set at the folder level to determine who can effectively invoke the service.

3.8 JBoss-specific security recommendations

This section contains application server configuration recommendations that are specific to JBoss 4.2.x when used to run LiveCycle.

3.8.1 Disable JBoss Management Console and JMX Console

Access to the JBoss Management Console and JMX Console is already configured (JMX monitoring is disabled) when you install LiveCycle on JBoss by using the turnkey installation method. If you are using your own JBoss Application Server, ensure that access to the JBoss Management Console and JMX monitoring console are secured. Access to the JMX monitoring console is set in the JBoss configuration file called jmx-invoker-service.xml.

3.8.2 Disable directory browsing

After logging into Administration Console, it is possible to browse the console's directory listing by modifying the URL. For example, if you change the URL to one of the following URLs, a directory listing may appear:

```
http://<servername>:8080/adminui/secured/
http://<servername>:8080/um/
```

To disable the directory listing, set the value of the listings initialization parameter of the `DefaultServlet` property to `false` as shown in bold in the *[JBoss root]* \server\default\deploy\jbossweb-tomcatxxx.sar\conf\web.xml file, as shown in this example:

```
<servlet>
    <servlet-name>default</servlet-name>
    <servlet-class>
        org.apache.catalina.servlets.DefaultServlet
    </servlet-class>
    <init-param>
        <param-name>listings</param-name><param-value>false</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>]
```

3.9 WebLogic-specific security recommendations

This section contains application server configuration recommendations for securing WebLogic 9.1 when running LiveCycle.

3.9.1 Disable directory browsing

Set the index-directories properties in the weblogic.xml file to `false`, as shown by this example:

```
<container-descriptor>
    <index-directory-enabled>false
    </index-directory-enabled>
</container-descriptor>
```

3.9.2 Enable WebLogic SSL Port

By default, WebLogic does not enable the default SSL Listen Port, 7002. Enable this port in the WebLogic Server Administration Console before you configure SSL.

3.10 WebSphere-specific security recommendations

This section contains application server configuration recommendations for securing WebSphere running LiveCycle.

3.10.1 Disable directory browsing

Set the `directoryBrowsingEnabled` property in the ibm-web-ext.xml file to `false`.

3.10.2 Enable WebSphere administrative security

1. Log in to the WebSphere Administrative Console.
2. In the navigation tree, go to one of the following links:

   (WebSphere 6.1) **Security** > **Secure administration, applications, andinfrastructure**

   (WebSphere 7.0) **Security** > **Global Security**
3. Select **Enable administrative security**.
4. Deselect both **Enable application security** and **Use Java 2 security**.
5. Click **OK** or **Apply**.
6. In the **Messages** box, click **Save directly to the master configuration**.

Chapter 4: Configuring Secure Administration Settings

Generally, developers do not use the LiveCycle production environment to build and test their applications. Therefore, you must administer user accounts and services that, although required in a private development environment, are not required in a production environment.

This section describes methods for reducing the overall attack surface through administration options that LiveCycle provides.

4.1 Disabling non-essential remote access to services

After LiveCycle is installed and configured, many services are available for remote invocation over SOAP, Enterprise JavaBeans™ (EJB), and (Deprecated for AEM forms) LiveCycle Remoting. The term *remote*, in this case, refers to any caller that has network access to the SOAP, EJB, or Action Message Format (AMF) ports for the application server.

Although the LiveCycle server services require valid credentials to be passed for an authorized caller, you should allow only remote access to the services that you need to be remotely accessible. To achieve limited accessibility, you should reduce the set of remotely accessible services to the minimum possible for a functioning system and then enable remote invocation for the additional services that you need.

LiveCycle server services always need at least SOAP access. These services are typically required for use by Workbench but also include services that are called by the Workspace web application.

Complete this procedure using the Applications and Services web page in Administration Console:

1 Log in to Administration Console by typing the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

2 Click **Services** > **Applications and Services** > **Preferences**.

3 Set the Preferences to view up to 200 services and endpoints on the same page.

4 Click **Services** > **Applications and Services** > **Endpoint Management**.

5 Select **EJB** from the **Provider** list and then click **Filter**.

6 To disable all EJB endpoints, select the check box beside each one in the list and click **Disable**.

7 Click **Next** and repeat the previous step for all EJB endpoints. Ensure that EJB is listed in the Provider column before you disable endpoints.

8 Select **SOAP** from the **Provider** list and then click **Filter**.

9 To remove SOAP endpoints, select the check box beside each one in the list and click **Remove**. Do not remove the following endpoints:

- AuthenticationManagerService
- DirectoryManagerService
- JobManager
- event_management_service

- event_configuration_service
- ProcessManager
- TemplateManager
- RepositoryService
- TaskManagerService
- TaskQueueManager
- TaskManagerQueryService
- WorkspaceSingleSignOn
- EventGenerationandReceipt
- ApplicationManager

10 Click **Next** and repeat the previous step for SOAP endpoints that are not in the above list. Ensure that SOAP is listed in the Provider column before you remove endpoints.

4.2 Disabling non-essential anonymous access to services

Some LiveCycle server services permit unauthenticated (anonymous) invocation for some operations. This means that one or more operations exposed by the service may be invoked as any authenticated user or as no authenticated user at all.

1 Log in to Administration Console by typing the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

2 Click **Services** > **Applications and Services** > **Service Management**.

3 Click the name of the service that you want to disable (for example, AuthenticationManagerService).

4 Click the **Security** tab,deselect **Anonymous Access Allowed**, and click **Save**.

5 Complete steps 3 and 4 for the following services:

- AuthenticationManagerService
- EJB
- Email
- JobManager
- WatchedFolder
- UsermanagerUtilService
- Remoting
- RemoteEvents
- RepositoryProviderService
- EMCDocumentumRepositoryProvider
- IBMFilenetRepositoryProvider
- FormAugmenter

- TaskManagerService
- TaskManagerConnector
- TaskManagerQueryService
- TaskQueueManager
- TaskEndpointManager
- LCMTMInvoker
- UserService
- WorkspaceSearchTemplateService
- WorkspaceSignleSignOn
- WorkspacePropertyService
- OutputService
- FormsService

If you intend to expose any of these services for remote invocation, you should also consider disabling anonymous access for these services. Otherwise, any caller with network access to this service may invoke the service without passing valid credentials.

Anonymous access should be disabled for any services that are not needed. Many internal services require anonymous authentication to be enabled because they need to be invoked by potentially any user in the system without being preauthorized.

4.3 Remove sample user and role assignments

You may have included sample users and roles when you installed LiveCycle (for example, Kel Varsen and the Finance Corp User Domain. Using the User Management administration pages, you should remove the sample user domain and sample roles.

4.3.1 Remove sample users

1 Log in to Administration Console by typing the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

2 Click **Settings > User Management > Users and Groups**.

3 Select the Sample Organization from the **and domain** list and click **Find**.

4 To disable all sample users, select the check box beside each one in the list and click **Delete**.

4.3.2 Remove sample domains

1 Log in to Administration Console by typing the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

2 Click **Settings > User Management > Domain Management**.

3 To delete all sample domains, select the check box beside each one in the list and click **Delete**.

4 Click **Save**.

4.4 Changing the default global time-out

End users can authenticate to LiveCycle through Workbench, LiveCycle web applications, or custom applications that invoke LiveCycle server services. One global time-out setting is used to specify how long such users can interact with LiveCycle (using a SAML-based Assertion) before they are forced to reauthenticate. The default setting is two hours. On a production environment, the amount of time needs to be reduced to the minimum number of minutes acceptable.

4.4.1 Minimize reauthentication time limit

1 Log in to Administration Console by typing the following URL in a web browser:

```
http://[host name]:[port]/adminui
```

2 Click **Settings > User Management > Configuration > Import And Export Configuration Files**.

3 Click **Export** to produce a config.xml file with the existing LiveCycle settings.

4 Open the XML file in an editor and locate the following entry:

```
<entry key="assertionValidityInMinutes" value="120"/>
```

5 Change the value to any number greater than 5 (in minutes) and save the file.

6 In Administration Console, navigate to the Import And Export Configuration Files page.

7 Enter the path to the modified config.xml file or click Browse to navigate to it.

8 Click **Import** to upload the modified config.xml file and then click **OK**.