# 1 Install and Setup

The recommendations below refers to the hardening guidelines when installing and setting up the Liberty server.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 1.1 | Ensure root does not have ownership of Websphere Liberty binaries (Manual) | Preventing the root user from owning files in the ${wlp.install.dir} directory prevents unauthorized commands or files from being run. | Set the ownership and group of ${wlp.install.dir} to something other than root: chown -R <non-root user>:<non-root group> ${wlp.install.dir} |
| 1.2 | Ensure extraneous files and directories are removed (Manual) | The installation might provide example applications, documentation, and other directories which may not serve a production use. | Remove extraneous resources for each: Extension: $ rm -rf ${wlp.user.dir}/extension/<non-production extension> Shared resource(s): $ rm -rf ${wlp.user.dir}/shared/<non-production shared resource(s)> Server: $ rm -rf ${wlp.user.dir}/server/<non-production server> Client: $ rm -rf ${wlp.user.dir}/client/<non-production client> |
| 1.3 | Ensure only defined users have access to the file system (Manual) | In Linux systems, there is a category of permission as others. These users are the ones who are not neither owner nor in a group associated with the file/folder. Since these users are anyone else than users defined, they should not have any access at all on Websphere Liberty file system. | Ensure that other has no access to the system. chmod -R o-rwx ${wlp.install.dir} |
| 1.4 | Ensure that only one user ID has write access to the WebSphere Liberty configuration files (Manual) | Ensure that only one user ID has write access to the WebSphere Liberty configuration files. If there are multiple administrators, they can use sudo and the /etc/sudoers file to elevate their privilege when write access is required. | Create a single, non-login, user ID that owns the server's configuration directory. Add any WebSphere administrators to the group that owns the server's configuration directory, which will automatically give them read access to the server's configuration, but not write access. Use sudo and the /etc/sudoers file to allow these administrators to elevate their privilege to the user ID that owns the server's configuration directory when write access is required. |
| 1.5 | Ensure Websphere Liberty Server Output is not set to the default value (Manual) | The user ID that the WebSphere Liberty server process runs under should not have write access to its own configuration files. The WebSphere Liberty server process requires write access to certain runtime files such as temporary caches and logs, but those files can be written to a different location than the configuration files, and permissions can be set separately for those two locations. | Create the server.env file if it does not exist. For more information about server.env please see here. Define WLP\_OUTPUT\_DIR in the server.env file: WLP\_OUTPUT\_DIR=/<server-writeable directory>/ Ensure that the WebSphere Liberty server process user ID has write access to the WLP\_OUTPUT\_DIR directory. Ensure that the WebSphere Liberty server process user ID does not have write access to the ${server.config.dir} directory. |
| 1.6 | Ensure automated configuration updates are disabled (Automated) | WebSphere Liberty provides the ability to automatically update the server runtime when the configuration changes, without requiring a server restart. | Add the updateTrigger attribute to the config element in ${server.config.dir}/configDropins/overrides/\*.xmland set to mbean or disabled. <config updateTrigger="mbean" /> |
| 1.7 | Ensure the WebSphere Liberty Installation is Validated (Manual) | Ensure that all WebSphere Liberty binaries were installed successfully and that you are running the latest fix pack version. | If the audit procedure fails, remove your current installation and install from a more secure location, ensuring that you are installing the latest fixpack from a trusted source. |
| 1.8 | Ensure Websphere Liberty file system access is Restricted (Manual) | The permissions of files and directories underneath ${wlp.install.dir} should follow the principle of least privilege. | Change WebSphere Liberty file system access to 750 (owner has read/write/execute, group has read/execute, other has no access): chmod –R 750 ${wlp.install.dir} |
| 1.9 | Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites (Automated) | WebSphere Liberty allows additional configuration files to be included in the main configuration file. Using included files in the main configuration file provides organization, separation, update controls and file access restriction. | Set the onConflict attribute to the IGNORE value in all include elements in the Liberty configuration. <include ... onConflict="IGNORE" /> Reduce file permission on all included files to essential users only. |

# 2 User Registries

A User registry in Liberty refers to a database of usernames and passwords and is used to validate a user's credential.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 2.1 | Ensure 'displayAuthenticationRealm' is set to 'false' (Automated) | Configuring displayAuthenticationRealm will ensure that the registry information is not displayed in the login prompt. This will restrict the potential leak of security realm information. | Set the displayAuthenticationRealm attribute in the webAppSecurity element to false in ${server.config.dir}/configDropins/overrides/<any file name>.xml. <webAppSecurity ... displayAuthenticationRealm="false" /> |
| 2.2 | Ensure Basic Registry and Quick Start security Registry are Removed (Automated) | The Basic Registry and Quick Start Security user registries are indented for developing and testing environments. | Use robust user registries, such as an LDAP registry or a Custom registry for production. |
| 2.3 | Ensure that the LDAP connection uses TLS (Automated) | TLS (Transport Layer Security) provides secure communication over a network. | Set the sslEnabled attribute on all ldapRegistry elements in ${server.config.dir}/configDropins/overrides/<any file name>.xml. Also set the sslRef attribute to a value that contains the correct keystore and truststore configuration for LDAPS communication. <ldapRegistry sslEnabled="true" sslRef="LDAPSSLSettings" > </ldapRegistry> <ssl id="LDAPSSLSettings" keyStoreRef="LDAPKeyStore" trustStoreRef="LDAPTrustStore" /> |

# 3 Application Deployment

By running Liberty with the Security Manager, applications are run in a sandbox which can prevent untrusted code from accessing files on the file system.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 3.1 | Ensure that automatic applications updates are disabled (Automated) | Automatic application updates will dynamically update the runtime behavior of the applications of any changes to the application. | Add the dropinsEnabled attribute and the updateTrigger attributes to the applicationMonitor element to ${server.config.dir}/configDropins/overrides/\*.xml. Set the dropinsEnabled to false to stop usage of dropins folder. Set the updateTrigger to mbean or disabled. <applicationMonitor updateTrigger="mbean" dropinsEnabled="false" /> |
| 3.2 | Ensure JDK Security Manager is Enabled (Automated) | The JDK's security manager allows applications to implement a security policy. It allows an application to permit or deny operations defined by the security policy. | Enable the websphere.java.security property in ${server.config.dir}/bootstrap.properties file websphere.java.security and also confirm that the appropriate permissions are granted in the application's permission.xml file and/or in the javaPermission element specified in ${server.config.dir}/configDropins/overrides/\*.xml. For example, <javaPermission className="java.security.PropertyPermission" name="os.name" actions="read" restriction="true" /> |

# 4 Web Applications

Security considerations that apply to Web applications like servlets, jsps etc.

### 4.1.1 Securing Session Cookies

JSESSIONID is a cookie generated by Servlet containers and used for session management in J2EE web applications for HTTP protocol

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.1.1.1 | Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies (Automated) | The SameSite attribute is used by web browsers to determine if a particular cookie should be sent with a request. Setting this attribute can help protect against Cross Site Request Forgery (CSRF) attacks. It is recommended to set the SameSite attribute to Strict. A Strict value for the SameSite attribute ensures the cookie is only sent by the web browser if the site for the cookie matches the site in the address bar, for example. | Add the cookieSameSite attribute to the httpSession element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. Set the cookieSameSite value to Strict. <httpSession cookieSameSite="Strict"/> |
| 4.1.1.2 | Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies (Manual) | The HttpOnly attribute on a cookie prevents the cookie from being accessed by the client side scripts. | Set the cookieHttpOnly attribute to true in the httpSession element in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <httpSession cookieHttpOnly="true" /> |
| 4.1.1.3 | Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies. (Automated) | The domain name attribute in a cookie specifies which hosts can receive the cookie. | Set the cookieDomain attribute in the httpSession element to the appropriate domain name in the ${server.config.dir}/configDropins/overrides/<any file name>.xml For Example, "mySubDomain.myCompany.com" <httpSession cookieDomain="mySubDomain.myCompany.com" /> |
| 4.1.1.4 | Ensure 'cookieSecure' secure attribute is set to 'true' (Automated) | The secure flag on a cookie will restrict the browser to send the cookies only on encrypted channels like HTTPS. | Set the cookieSecure attribute to true in the httpSession element in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <httpSession cookieSecure="true"/> |

### 4.1.2 Securing Authentication Cookies

Security cookies used for authentication including LTPA and JWT cookies

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.1.2.1 | Ensure 'sameSiteCookie' attribute is set to 'Strict' (Manual) | The SameSite attribute is used by web browsers to determine if a particular cookie should be sent with a request. Setting this attribute can help protect against Cross Site Request Forgery (CSRF) attacks. It is recommended to set the SameSite attribute to Strict. A Strict value for the SameSite attribute ensures the cookie is only sent by the web browser if the site for the cookie matches the site in the address bar, for example. | Add the sameSiteCookie attribute to the webAppSecurity element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. Set the sameSiteCookie value to Strict. <webAppSecurity sameSiteCookie="Strict"/> |
| 4.1.2.2 | Ensure 'ssoDomainNames' attribute is configured for the authentication cookies. (Automated) | The domain name attribute in a cookie specifies which hosts can receive the cookie. | Add the appropriate domain name to the ssoDomainNames attribute in the webAppSecurity element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. For example, to add mySubDomain.myCompany.com <webAppSecurity ssoDomainNames="mySubDomain.myCompany.com"/> |
| 4.1.2.3 | Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie. (Automated) | The secure flag on a cookie will restrict the browser to send the cookies only on encrypted channels like HTTPS. | Set the setCookieSecureFlag attribute to true in the jwtSso element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. <jwtSso setCookieSecureFlag="true"/> |
| 4.1.2.4 | Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies (Automated) | Cookies with the secure flag will only be sent over encrypted HTTPS requests. | Set the ssoRequiresSSL attribute is set to true in the webAppSecurity element on ${server.config.dir}/configDropins/overrides/\*.xml <webAppSecurity ssoRequiresSSL="true" /> |
| 4.1.2.5 | Ensure 'ssoCookieName' LTPA cookie name is set (Automated) | The server will authenticate only the LTPA cookie name that is configured. When the request contains other LTPA cookie names, they will be ignored. | Set the ssoCookieName attribute to something other than LtpaToken2 in the webAppSecurity in the ${server.config.dir}/configDropins/overrides/\*.xml For Example, obscureCookieName2 <webAppSecurity ssoCookieName="obscureCookieName2" /> Set the useOnlyCustomCookieName attribute to true in the webAppSecurity in the ${server.config.dir}/configDropins/overrides/\*.xml <webAppSecurity useOnlyCustomCookieName="true" /> |
| 4.1.2.7 | Ensure 'trackLoggedOutSSOCookies' is set to 'true' (Automated) | The trackLoggedOutSSOCookies attribute keeps track of the LTPA cookies that are logged out in a running server. | Set trackLoggedOutSSOCookies to true in the webAppSecurity element in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <webAppSecurity trackLoggedOutSSOCookies="true" /> |
| 4.1.2.8 | Ensure 'cookieName' JWT (JSON Web Token) cookie name is set (Automated) | The server will authenticate only the JWT cookie name that is configured. When the request contains other JWT cookie names, they will be ignored. | Set the cookieName attribute to any obscure value in jwtSso element in the ${server.config.dir}/configDropins/overrides/\*.xml For Example, "obscuredCookieName2" <jwtSso cookieName="obscuredCookieName2" /> Set the useOnlyCustomCookieName attribute to true in the webAppSecurity in the ${server.config.dir}/configDropins/overrides/\*.xml <webAppSecurity useOnlyCustomCookieName="true" /> |

### 4.1.3 Securing Other Cookies

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.1.3.1 | Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies (Automated) | The SameSite attribute is used by web browsers to determine if a particular cookie should be sent with a request. Setting this attribute can help protect against Cross Site Request Forgery (CSRF) attacks. It is recommended to set the SameSite attribute to Strict. A Strict value for the SameSite attribute ensures the cookie is only sent by the web browser if the site for the cookie matches the site in the address bar for example. | Set the strict attribute to \* in the samesite element in the httpEndpoint element in ${server.config.dir}/configDropins/overrides/\*.xml. <httpEndpoint ... <samesite strict="\*"/> </httpEndpoint> |

## 4.2 Secure Transport

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.2.1 | Ensure 'trustDefaultCerts' is set to 'false' (Automated) | The default certificates from the Java runtime are trusted by the server in addition to the certificates configured in the SSL/TLS configurations in the server. | Add the trustDefaultCerts attribute to all ssl elements to ${server.config.dir}/configDropins/overrides/\*.xml and set the value to false. <ssl trustDefaultCerts="false" /> |
| 4.2.2 | Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security) (Automated) | The latest versions of TLS provide drop support for less secure cryptographic features and add support for more advanced cryptographic algorithms. | Set the sslProtocol attribute version to the latest supported level in all ssl elements to ${server.config.dir}/configDropins/overrides/\*.xml. <ssl ... sslProtocol="TLSv1.2" /> |
| 4.2.3 | Ensure HSTS (HTTP Strict Transport Security) is enabled (Automated) | The HSTS response header informs browsers that the site should only be accessed using HTTPS, and that any future attempts to access it using HTTP should automatically be converted to HTTPS. | Set the addstricttransportsecurityheader attribute in the webContainer element in the ${server.config.dir}/configDropins/overrides/\*.xml as follows: <webContainer addstricttransportsecurityheader="maxage=31536000;includeSubDomains" /> |
| 4.2.4 | Ensure that outbound TLS configurations are specified (Automated) | If there are no specific SSL/TLS configurations configured for outbound TLS connections the common TLS configurations are used by both the inbound and outbound connections. | Add the outboundSSLRef attribute to the sslDefault elements to ${server.config.dir}/configDropins/overrides/\*.xml and set the value to a valid ssl configuration id. <sslDefault outboundSSLRef="alternateSSLSettings" /> Also add host and port attributes on the outboundConnection elements for all ssl elements used for outbound requests to ${server.config.dir}/configDropins/overrides/\*.xml and set the values to hosts and ports used by the application or server. <ssl id="alternateSSLSettings" ... <outboundConnection host="hostname1" port="020" /> <outboundConnection host="hostname2" port="9020" /> </ssl> |
| 4.2.5 | Ensure that secure ciphers suites are configured (Automated) | A cipher suite includes a set of algorithms used when making secure TLS connections. Strong cipher suites contain more secure algorithms. | Add the securityLevel attribute to all ssl elements to ${server.config.dir}/configDropins/overrides/\*.xml and set the value to HIGH. Also add the enforceCipherOrder attribute and set the value to true. <ssl ... securityLevel="HIGH" enforceCipherOrder="true" /> Or add the enabledCiphers attribute to all ssl elements to ${server.config.dir}/configDropins/overrides/\*.xml and set the value to space separated list of appropriate strong ciphers in the preferred order. For example, <ssl ... enabledCiphers="TLS\_AES\_256\_GCM\_SHA384 SSL\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 SSL\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384" enforceCipherOrder="true" /> |
| 4.2.6 | Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications (Automated) | The transport-guarantee setting of CONFIDENTIAL will enforce that the application can only be accessed through HTTPS secure connection. HTTPS protocol protects the integrity and confidentiality of data between the client and the server. | Add the transport-guarantee attribute to user-data-constraint element under the security-constraint in ${wlp.user.dir}/shared/apps/WEB-INF/web.xml and set the value to CONFIDENTIAL. <security-constraint> ... <user-data-constraint> <transport-guarantee>CONFIDENTIAL</transport-guarantee> </user-data-constraint> Note: The security-constraints can also be set in the code using annotations like @ServletSecurity.TransportGuarantee |
| 4.2.7 | Ensure Hostname verification for TLS communication is enabled (Automated) | Hostname verification is a server identity check that is used to ensure that a client is talking to the correct server. The check is performed on the client side of an SSL communication and involves looking at the server’s certificate Subject Alternative Name (or the SubjectDN) to see if it matches the host part of the URL that was used to make the outbound request. | Enable hostname verification in the SSL configuration by adding the verifyHostname attribute to the ssl configuration element in ${server.config.dir}/configDropins/overrides/<any file name>.xml <ssl ... verifyHostname="true" /> For JAX-RS client, enable hostname verification in the webTarget element by adding the disableCNCheck attribute in ${server.config.dir}/configDropins/overrides/<any file name>.xml <webTarget ... disableCNCheck="false" /> |
| 4.2.8 | Ensure that CA (Certificate Authority) certificates are used (Automated) | SSL/TLS certificates are used to establish trust during the secure communications. Certificates can be a simple self-signed cert or can be from a well established or known CA authority. | Add non self signed CA certificates as described here. |
| 4.2.9 | Ensure 'ocsp.enable' certificate revocation is set to 'true' (Automated) | Certificate revocation is the process of canceling the digital certificate of the revoked user and keeping track of them. | Add ocsp.enable=true in $JAVA\_HOME/jre/lib/security/java.security file. ocsp.enable=true |
| 4.2.10 | Ensure mutual TLS authentication is enabled (Automated) | Mutual TLS authentication requires that both the server and the client authenticate to the other during SSL/TLS handshake. | Add these settings to ${server.config.dir}/configDropins/overrides/<any file name>.xml for direct login to WebSphere Liberty. <httpDispatcher ... trustedHeaderOrigin=“none”/> <ssl ... clientAuthentication="true" /> For login in conjunction with a proxy, add the following. Replace the ip addresses with your values. <httpDispatcher ... trustedHeaderOrigin=“10.20.30.40, 10.20.50.60”/> <ssl ... clientAuthentication="true" /> |
| 4.2.11 | Ensure that strong algorithms are used for TLS certificates. (Manual) | Certificates created with stronger algorithms use stronger hashes which is more secure. Also, some browsers warn when using weak certificates. | Create certificates with SHA256 or higher algorithm and 2048 or higher key bit size. |
| 4.2.12 | Ensure `httpPort` attribute set to `-1` (Automated) | Disabling the http port in the server configuration ensures that only the secure https protocol will be used to access the web applications. HTTPS protocol protects the integrity and confidentiality of data between the client and the server. | Set the httpPort attribute to -1 in the httpEndpoint element in ${server.config.dir}/configDropins/overrides/\*.xml. <httpEndpoint ... httpPort="-1"/> |
| 4.2.13 | Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates (Manual) |  |  |

## 4.3 Ensure the Use of Dedicated Administrative Accounts

Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.  
● ● ●

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.3.3 | Ensure 'tokenReuse' is set to 'false' (Automated) | The "jti" claim in the JWT token can be made unique to prevent the JWT tokens from being replayed. | Add the tokenReuse attribute to the openidConnectClient element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to false to prevent token reuse on JSON Web Tokens. <openidConnectClient tokenReuse="false" /> |
| 4.3.4 | Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party) (Automated) | The issuer claim in the JWT token is used by the RP (relying party) to verify the OP (OIDC provider) token issuer. | Add the disableIssChecking attribute to the openidConnectClient element to ${server.config.dir}/configDropins/overrides/\*.xml. Set the disableIssChecking attribute value to false to ensure that issuer claim checking for JSON Web Tokens occurs. <openidConnectClient disableIssChecking="false" /> |
| 4.3.5 | Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP) (Automated) | Hostname verification is a server identity check that is used to ensure that a client is talking to the correct server. The check is performed on the client side of an SSL communication and involves looking at the server’s certificate Subject Alternative Name (or the SubjectDN) to see if it matches the host part of the URL that was used to make the outbound request. | Add the hostNameVerificationEnabled attribute to the openidConnectClient element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to true to do hostname verification for JSON Web Tokens. <openidConnectClient hostNameVerificationEnabled="true" /> |
| 4.3.6 | Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP) (Automated) | The signatureAlgorithm is used by the RP to verify the signed ID tokens sent by the OP. | Add the signatureAlgorithm attribute to the openidConnectClient element to ${server.config.dir}/configDropins/overrides/\*.xml and set it a valid signature algorithm type, for example RS256, to enable token signing for JSON Web Tokens. <openidConnectClient signatureAlgorithm="RS256" /> |
| 4.3.7 | Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP) (Automated) | The signatureAlgorithm is used by the OP to sign the ID tokens. | Add the signatureAlgorithm attribute to the openidConnectProvider element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to a valid algorithm, such as RS256, to ensure tokens are signed. <openidConnectProvider signatureAlgorithm="RS256" /> |
| 4.3.8 | Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP) (Automated) | HTTPS protocol protects the integrity and confidentiality of data between the client and the server. | Add the httpsRequired attribute to the openidConnectClient element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to true to ensure that security transport is used for JSON Web Tokens. <openidConnectClient httpsRequired="true" /> |
| 4.3.10 | Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0 (Automated) | The OAuth access token contains sensitive data and should be hashed to protect it. | Add the accessTokenEncoding attribute to the oauthProvider element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to a valid encoding type, for example PBKDF2WithHmacSHA512, to enable stored access token encoding. Do not use the plain value as it does not encode. <oauthProvider accessTokenEncoding="PBKDF2WithHmacSHA512" /> |
| 4.3.11 | Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0 (Automated) | Public clients can be blocked to access OAuth applications for better control. | Add the allowPublicClients attribute to the oauthProvider element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to false to block public clients. <oauthProvider allowPublicClients="false" /> |
| 4.3.12 | Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0 (Automated) | The OAuth client secret is encoded using the 'clientSecretEncoding' attribute. | Add the clientSecretEncoding attribute to the oauthProvider element to ${server.config.dir}/configDropins/overrides/\*.xml is set to a valid encoding type, for example PBKDF2WithHmacSHA512, to enable stored access token encoding. <oauthProvider clientSecretEncoding="PBKDF2WithHmacSHA512" /> |
| 4.3.13 | Ensure 'httpsRequired' is set to 'true' in OAuth 2.0 (Automated) | HTTPS protocol protects the integrity and confidentiality of data between the client and the server. | Add the httpsRequired attribute to the oauthProvider element to ${server.config.dir}/configDropins/overrides/\*.xml and set to true to ensure secure transport with a client. <oauthProvider httpsRequired="true" /> |
| 4.3.14 | Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0 (Automated) | Resource owner validation check validates the resource owner credentials. | Add the skipResourceOwnerValidation attribute to the oauthProvider element to ${server.config.dir}/configDropins/overrides/\*.xml and set to false to ensure resource owner validation is completed. <oauthProvider skipResourceOwnerValidation="false" /> |
| 4.3.15 | Ensure 'httpsRequired' is set to 'true' in SAML (Automated) | HTTPS protocol protects the integrity and confidentiality of data between the client and the server. | Add the httpsRequired attribute to all samlWebSso20 elements to ${server.config.dir}/configDropins/overrides/\*.xml and set it to true. <samlWebSso20 ... httpsRequired="true" /> |
| 4.3.16 | Enforce 'wantAssertionsSigned' to 'true' in SAML (Automated) | A SAML (Security Assertions Markup Language) authentication assertion is issued as proof of an authentication event and can be signed. | Add the wantAssertionsSigned attribute to all samlWebSso20 elements to ${server.config.dir}/configDropins/overrides/\*.xml and set it to true. <samlWebSso20 ... wantAssertionsSigned="true" /> |
| 4.3.17 | Ensure 'authnRequestsSigned' is set to 'true' in SAML (Automated) | SAML providers can sign the request messages for authenticity. | Add the authnRequestsSigned attribute to all samlWebSso20 elements to ${server.config.dir}/configDropins/overrides/\*.xml and set it to true. <samlWebSso20 ... authnRequestsSigned="true" /> |

## 4.4 General

General recommendations related to web resources including servlets, JSPs, JAX-RS applications.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 4.4.1 | Ensure 'disableXPoweredBy' is set to 'true' (Automated) | The disableXPoweredBy setting can reveal the server's identity. | Add the disableXPoweredBy attribute to webContainer element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. Set the disableXPoweredBy attributes value to true. <webContainer disableXPoweredBy="true" /> |
| 4.4.2 | Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false' (Automated) | Setting the preserveFullyQualifiedReferrerUrl attribute to false ensures that the host for the referrer URL is removed, and that the redirect is to localhost. | Set the preserveFullyQualifiedReferrerUrl attribute to false in the webAppSecurity element on ${server.config.dir}/configDropins/overrides/\*.xml <webAppSecurity preserveFullyQualifiedReferrerUrl="false" /> |
| 4.4.4 | Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic (Manual) | Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the hostNameExcludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set to comma-separated list of host names. <tcpOptions hostNameExcludeList="\*.abc.com,sample.all.com" /> |
| 4.4.5 | Ensure 'logoutOnHttpSessionExpire' is set to 'true' (Automated) | Logout users after the HTTP session timer expires. | Set the logoutOnHttpSessionExpire attribute to true in the webAppSecurity element on ${server.config.dir}/configDropins/overrides/\*.xml <webAppSecurity logoutOnHttpSessionExpire="true" /> |
| 4.4.6 | Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic (Manual) |  |  |
| 4.4.7 | Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic (Automated) |  |  |
| 4.4.8 | Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic (Manual) | IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the addressExcludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set it to a commaseparated list of IP addresses. <tcpOptions addressExcludeList="254.1.0.9,\*.1.255.0" /> |
| 4.4.10 | Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses (Automated) | The web server plug-in uses private headers to provide information about the original request. These headers take precedence over the HTTP host header and are used to select a virtual host to service a request. To restrict private header processing to specific trusted sources, specify a comma-separated list of IP addresses and hostnames. | Add the settings below to ${server.config.dir}/configDropins/overrides/<any file name>.xml. <httpDispatcher ... trustedHeaderOrigin="localhost, 127.0.0.1, 192.168.\*.\*, 0:0:0:0:0:ffff:\*:\*, \*.ibm.com"/> |
| 4.4.12 | Ensure security constraints are specified to protect web applications (Automated) | Protect web applications by configuring security constraints for all web resources using either deployment descriptor and/or annotations. | Set <security-constraint> elements in the web.xml deployment descriptor file of each application or use annotations in the code. Example using security-constraint: <security-constraint> <web-resource-collection> <url-pattern>/\*</url-pattern> </web-resource-collection> <auth-constraint> <role-name>testing</role-name> </auth-constraint> </security-constraint> Example using annotations: @WebServlet("/myServlet") @ServletSecurity( httpMethodConstraints = { @HttpMethodConstraint(value = "GET", rolesAllowed = "user"), @HttpMethodConstraint(value = "POST", rolesAllowed = "manager", transportGuarantee = TransportGuarantee.CONFIDENTIAL), } ) public class myServlet extends HttpServlet { // servlet code... } |
| 4.4.13 | Ensure application security feature is enabled (Automated) | The app security features (any of the appSecurity-\* versions) enable basic features like authentication, authorization and transport security. Depending on the resources in the servers and the security configuration needed additional security features also need to be enabled. | Add the appSecurity-2.0 under the featureManager element in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <featureManager> <feature>appSecurity-2.0</feature> </featureManager> |
| 4.4.14 | Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false' (Automated) | When a user tries to access a session owned by another user, the UnauthorizedSessionRequestException is raised so the user cannot continue. | Set the invalidateOnUnauthorizedSessionRequestException attribute to true in the httpSession element in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <httpSession invalidateOnUnauthorizedSessionRequestException="true" /> |
| 4.4.16 | Ensure HTTP session overflow is 'disabled' (Manual) | Applications that use in-memory HTTP sessions can configure if overflow is allowed and the number of sessions that can be created. | Set the allowOverflow attribute on the httpSession element to the value of false in ${server.config.dir}/configDropins/overrides/\*.xml. Also set the maxInMemorySessionCount attribute on the httpSession element to a maximum number of sessions the JVM is able to support for each web module. <httpSession allowOverflow="false" maxInMemorySessionCount="1000" /> |
| 4.4.17 | Ensure uncovered http methods are denied (Automated) | Servlets are secured by URL and each URL that is to be secured must be specified in the web.xml file describing the application. | Create aliases for each servlet. Assign a security constraint for each URL defined in alias. Add additional security by adding <deny-uncovered-http-methods /> to WEBINF/web.xml to block all undeclared methods. <servlet-mapping id="ServletMapping\_1"> <servlet-name>MyServlet</servlet-name> <url-pattern>/MyURLPattern</url-pattern> </servlet-mapping> <deny-uncovered-http-methods /> <!-- SECURITY CONSTRAINTS --> <security-constraint id="SecurityConstraint\_1"> <web-resource-collection id="WebResourceCollection\_1"> <web-resource-name>Protected with Employee or Manager roles</web-resource-name> <url-pattern>/MyURLPattern</url-pattern> <http-method>GET</http-method> <http-method>POST</http-method> </web-resource-collection> <auth-constraint id="AuthConstraint\_1"> <role-name>Employee</role-name> <role-name>Manager</role-name> </auth-constraint> </security-constraint> Note: The security constraints can also be set in the code using annotations like @ServletSecurity |
| 4.4.18 | Ensure 'disallowServeServletsByClassName' is 'disabled' (Automated) | Servlets can be served by class name or via a normal URL alias. | Add the disallowServeServletsByClassName attribute on the webContainer element in ${server.config.dir}/configDropins/overrides/\*.xml and set it to true. <webContainer disallowServeServletsByClassName="true"/> If serving servlets by class name is set at the application level, update the webApplication elements or ibm-web-ext files. |
| 4.4.19 | Ensure server headers on requests are removed (Automated) | A server header contains information about the software used by the server to handle the request. This information can be returned to browsers or web clients in certain situations. | Add the removeServerHeader attribute to httpOptions element specified in ${server.config.dir}/configDropins/overrides/\*.xml. Set the removeServerHeader attribute value to true. <httpEndpoint id="defaultHttpEndpoint" ... <httpOptions removeServerHeader=`true`/> </httpEndpoint> |
| 4.4.20 | Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications (Automated) | Disable directory browsing for your web applications. Directory browsing automatically list the contents of directories that do not have an index page or welcome page present. | Add the directoryBrowsingEnabled attribute on the webContainer element in ${server.config.dir}/configDropins/overrides/\*.xml and set it to false. <webContainer directoryBrowsingEnabled="false"/> If directory browsing is set at the application level, update the webApplication elements or ibm-web-ext files. |
| 4.4.21 | Ensure 'default-error-page' is set for web applications (Manual) | When errors occur in a Web application or before the application dispatch, an error message is displayed to the user. By default, the app server displays an exception stack dump of the error. A default error handler or page should be defined for all applications. | Add the default-error-page attribute on the web-ext element for all webApplication elements in ${server.config.dir}/configDropins/overrides/\*.xml and set it to a valid error page. <webApplication ... > <web-ext default-error-page="errorPageName.jsp"/> |
| 4.4.22 | Ensure virtual hosts are defined to isolate applications (Automated) | Isolate applications by configuring separate virtual hosts. | Ensure the following to enable virtual hosting. 1. Configure the application with virtual-host name in the enterpriseApplication or webApplication elements in the [Liberty configuration]${server.config.dir}/configDropins/overrides/\*.xml <webApplication ... > <web-bnd virtual-host name="myApplication1"/> </webApplication> Note: The virtual-host name can also be set in the ibm-web.bnd.xml file in the application. 2. Configure the application to use the specific virtual host in ${server.config.dir}/configDropins/overrides/\*.xml. <virtualHost id="myApplication1"> <hostAlias>your\_host\_name:9080</hostAlias> </virtualHost> |
| 4.4.23 | Ensure virtual hosts are Defined to isolate JMX communication and application traffic (Automated) |  |  |

## 4.5 Implement and Manage a Firewall on End-User

Devices  
Implement and manage a host-based firewall or port-filtering tool on end-user  
devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.  
● ● ●

## 4.8 Uninstall or Disable Unnecessary Services on

Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.  
● ●

## 5.1 The CSIv2 (Common Secure Interoperability version 2) serverPolicy

CSIv2 is a protocol used by the EJB (Enterprise Java Beans) clients to communicate using RMI/IIOP with the EJB applications. This protocol supports different authentication mechanisms. The following recommendations discuss all the 3 mechanisms for the CSIv2 policy on the server side. One can configure one or more of these to protect the EJB applications as per the CSIv2 specification.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 5.1.1 | Ensure 'sslEnabled' is set to 'true' within the CSIv2 Transport Layer (Automated) | The CSIv2 Transport policy configures security at the transport layer when accessing EJB applications using RMI/IIOP. | Set the sslEnabled attribute in ORB > serverPolicy.csiv2 > layers > transportLayer to true in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <orb id="defaultOrb"> <serverPolicy.csiv2> <layers> <transportLayer sslEnabled="true"/> </layers> </serverPolicy.csiv2> </orb> |
| 5.1.2 | Ensure 'establishTrustInClient' is set to 'required' within the CSIv2 Authentication Layer (Automated) |  |  |
| 5.1.3 | Ensure 'identityAssertionEnabled' is set to 'true' within the CSIv2 Attribute Layer (Automated) | The CSIv2 Attribute policy configures security at the Attribute layer when accessing EJB applications using RMI/IIOP. | Set the identityAssertionEnabled attribute to true and identityAssertionTypes to ITTX509CertChain, ITTDistinguishedName in ORB > serverPolicy.csiv2 > layers > authenticationLayer in the ${server.config.dir}/configDropins/overrides/<any file name>.xml <orb id="defaultOrb"> <serverPolicy.csiv2> <layers> <attributeLayer identityAssertionEnabled="true" identityAssertionTypes="ITTX509CertChain, ITTDistinguishedName"/> </layers> </serverPolicy.csiv2> </orb> |

## 5.2 The CSIv2 (Common Secure Interoperability version 2) Client Policy

CSIv2 is a protocol used by the EJB (Enterprise Java Beans) clients to communicate using RMI/IIOP with the EJB applications. This protocol supports different authentication mechanisms. The following recommendations discuss all the 3 mechanisms for the CSIv2 policy on the client side. One can configure one or more of these to protect the EJB applications as per the CSIv2 specification.

## 5.3 Java Serialization

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 5.3.1 | Ensure filters are configured for Java serialization (JEP 290) (Manual) | Java provides configuration to allow incoming streams of object-serialization data to be filtered in order to improve both security and robustness. This allows | Set the System property jdk.serialFilter in the ${server.config.dir}/bootstrap.properties file to the correct filters to restrict classes to be deserialized. For example, jdk.serialFilter=!com.myCompany.restrictClass;com.myCompany.allowClass |

## 5.4 Restrict Administrator Privileges to Dedicated

Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user’s primary, non-privileged account.  
● ● ●

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 5.4.1 | Ensure that all appropriate EJB methods are protected (Automated) | Access to the EJB methods should be protected appropriately using roles in deployment descriptor or annotations. | Set <method-permission> elements in the ejb-jar.xml deployment descriptor file of each application or use annotations. Example using method-permission: <method-permission> <role-name>teller</role-name> <method> <ejb-name>myEJB1</ejb-name> <method-name>getBalance</method-name> </method> </method-permission> Example using annotations: @RolesAllowed("teller") public class myEJB1 { public void getBalance () {...} ... } |

# 6 Web Services

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 6.1 | Ensure 'HttpsToken' is set in WS-Security policy (Automated) | Protect JAX-WS Web services applications by enabling HTTPS secure transport in WSSecurity policy. Enabling HTTPS secure transport in WS-Security policy protects JAXWS web services. Enable HTTPS for secure communications. | Add HttpsToken as seen in the example to wsdl or policy attachment files for each web service. <wsp:Policy ...> ... <sp:TransportBinding> <wsp:Policy> <sp:TransportToken> <wsp:Policy> <sp:HttpsToken /> </wsp:Policy> </sp:TransportToken> ... </wsp:Policy> </sp:TransportBinding> </wsp:Policy> |
| 6.2 | Ensured 'HashPassword' is set in UsernameToken WSSecurity policy (Automated) | Hashing passwords in the Username token of WS-Security policy obfuscates the password which is more secure. | Add HashPassword as seen in the example to wsdl or policy attachment files for each web service. <sp:UsernameToken sp:IncludeToken="..."> ... <wsp:Policy> <sp:WssUsernameToken11 /> <sp:HashPassword /> </wsp:Policy> .. </sp:UsernameToken> |
| 6.3 | Ensure CallbackHandler is used to access private keys in keystore files (Manual) | Use CallbackHandler to retrieve user password and for accessing private keys in keystore files. Using the CallbackHandler method to retrieve a user's password and access private keys in keystore files \_\_\_ (state benefit). Do not use plain text passwords in user and keystore fields for WebServices security. | Ensure that the passwords are not configured in the wsSecurityClient and wsSecurityProvider elements in ${server.config.dir}/configDropins/overrides/<any file name>.xml. For more information, see the References section. "Implement the callbackhandler method. For more information, see the References section." <wsSecurityClient id="default" ws-security.callbackhandler="com.myCompany.myExample.myCBH" ...> ... </wsSecurityClient> <wsSecurityProvider id="default" ws-security.callbackhandler="com.myCompany.myExample.myCBH" ...> ... </wsSecurityProvider> |
| 6.4 | Ensure SOAP messages are Signed and encrypted with WSSecurity policy (Manual) | Signing and encrypting SOAP messages protects JAX-WS web services. | Add SignedParts and EncryptedParts assertions to sign and encrypt SOAP Body wsdl or policy attachment files for each web service. <wsp:Policy> ... <sp:SignedParts> <sp:Body /> </sp:SignedParts> ... <sp:EncryptedParts> <sp:Body /> </sp:EncryptedParts> ... </wsp:Policy> |
| 6.6 | Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy (Automated) | Using strong signature and encryption algorithms when signing and encrypting SOAP messages using WS-Security policy increases security. Using strong signature and encryption algorithms when signing and encrypting SOAP messages using WS-Security policy is more secure. | Use Basic256Sha256 for Algorithm suite in WS-Security policy as seen in the example to wsdl or policy attachment files for each web service. <sp:AlgorithmSuite> <wsp:Policy> <sp:Basic256Sha256/> </wsp:Policy> </sp:AlgorithmSuite> |

## 6.8 Define and Maintain Role-Based Access Control

Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.  
●

# 7 Messaging

This section covers the hardening guidelines for the messaging features.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 7.1 | Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names (Manual) | Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the hostNameExcludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set to comma-separated list of host names. <tcpOptions hostNameExcludeList="\*.abc.com,sample.all.com" /> |
| 7.2 | Ensure the 'hostNameIncludeList attribute' is set to a whitelist of host names (Manual) | Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the hostNameIncludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set to comma-separated list of host names. <tcpOptions hostNameIncludeList="\*.def.com,sample.here.com" /> |
| 7.3 | Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames (Manual) | IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the addressExcludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set it to a commaseparated list of IP addresses. <tcpOptions addressExcludeList="254.1.0.9,\*.1.255.0" /> |
| 7.4 | Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses (Manual) | IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints. | Add the addressIncludeList attribute to all tcpOptions elements to ${server.config.dir}/configDropins/overrides/\*.xml and set to comma-separated list of IP Address. <tcpOptions addressIncludeList="254.\*.\*.9,255.0.0.2" /> |
| 7.5 | Ensure the `useSSL` attribute is set to `true` for TLS Transport (Automated) | Configuring TLS provides secure communication for JmsOutbound connections. | Add the useSSL attribute to wasJmsOutbound element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. Set the useSSL attributes value to true. <wasJmsOutbound ... useSSL ="true" /> |

## 7.7 Remediate Detected Vulnerabilities

Remediate detected vulnerabilities in software through processes and tooling on a monthly, or more frequent, basis, based on the remediation process.  
● ●

# 8 MicroProfile Metrics

Recommendations related to MicroProfile feature configuration. This section discusses only the recommendations specific to the MicroProfile features. Any common features like JWTs (JSON Web Tokens) are listed in the SSO sub-section under the Web Application Security section.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 8.1 | Ensure 'authentication' is set to 'true' to protect the metrics end point (Automated) | Protect access to the metrics end point in the MicroProfile feature so that only valid users are allowed to access it. | Ensure that the authentication attribute of the mpMetrics element is set to true in the ${server.config.dir}/configDropins/overrides/<any file name>.xml. <mpMetrics authentication="true"/> |

# 9 z/OS

This section covers the hardening guidelines for WebSphere Liberty on the z/OS platform.

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 9.1 | Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization (Automated) | The SAF role mapper should be used to perform SAF authorization checks when accessing applications. | Configure the zosSecurity-1.0 feature and set the safAuthorization element in ${server.config.dir}/configDropins/overrides/<any file name>.xml. <feature>zosSecurity-1.0</feature> <safAuthorization id="saf" /> |
| 9.3 | Ensure 'safkeyringhw:' is set to use a hardware crypto card (Manual) | Use hardware crypto card to store to store cryptographic keys and certificates. | For the crypto card configuration, configure the location attribute in the keyStore elements referenced by the SSL configurations to point to a valid hardware crypto keyring configuration and set the type attribute to JCECCARACFKS in ${server.config.dir}/configDropins/overrides/<any file name>.xml. <keyStore id="defaultKeyStore" location="safkeyringhw:///myHWKeyring" type="JCECCARACFKS" ... /> In addition, make sure that the com.ibm.crypto.hdwrCCA.provider.IBMJCECCA provider is configured in the $JAVA\_HOME/jre/lib/security/java.security file. ... security.provider.3=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA ... For the hybrid crypto card configuration, configure the location attribute in the keyStore elements referenced by the SSL configurations to point to a valid hybrid hardware crypto keyring configuration and set the type attribute to JCEHYBRIDRACFKS in ${server.config.dir}/configDropins/overrides/<any file name>.xml. <keyStore id="defaultKeyStore" location="safkeyringhybrid:///myHybridKeyring" type="JCEHYBRIDRACFKS" .../> In addition, configure the com.ibm.crypto.ibmjcehybrid.provider.IBMJCEHYBRID and the com.ibm.crypto.hdwrCCA.provider.IBMJCECCA providers in the $JAVA\_HOME/jre/lib/security/java.security file. ... security.provider.2=com.ibm.crypto.ibmjcehybrid.provider.IBMJCEHYBRID security.provider.3=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA ... |

## 9.4 Apply Host-based Firewalls or Port Filtering

Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.  
● ● ●  
● ● ●

## 9.5 Implement Application Firewalls

Place application firewalls in front of any critical servers to verify and validate the traffic going to the server. Any unauthorized traffic should be blocked and logged.  
●  
●

# 10 Miscellaneous

Includes miscellaneous recommendations

|  |  |  |  |
| --- | --- | --- | --- |
| Control Reference ID | Control Name & Description | Description | Remediation |
| 10.1 | Ensure Unused Features are Removed (Automated) | The Websphere Liberty architecture provides developers with the option to tune their server to only the features that they need. For example, the JDBC feature only needs to be added if database access is required. If you did not want any remote administrative access to the REST interface, you would remove the REST feature. | Removed any unneeded features listed under the featureManager element in the Liberty configuration |
| 10.2 | Ensure Passwords are Encrypted (Automated) | Sensitive information is stored in the Liberty configuration and can be in clear text unless encrypted or hashed. | Use the Liberty provided securityUtility tool to encrypt the password. To encrypt a password with Advanced Encryption Standard (AES) encryption: securityUtility encode --encoding=aes --key=myKey passW0rd returns: {aes}AE/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3ev396oYRw== Replace the clear text password in the Liberty configuration with the encrypted password created by the securityUtility tool. In this example, the encrypted password {aes}AE/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3ev396oYRw== was returned by the securityUtility and used to update the bindPassword attribute in the ldapRegistry element. <ldapRegistry ... bindPassword="{aes}AE/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3e v396oYRw==" </ldapRegistry> In WebSphere Liberty, the default key that is used for encrypting and decrypting can be overridden by setting the wlp.password.encryption.key property. Make sure that you do not set this property in main Liberty configuration. Otherwise, the file that contains the key might be included when you run the server dump or server package commands. Instead, set the wlp.password.encryption.key property in a separate configuration file and include it in the Liberty configuration, as shown in the following example: Create a file named key.xml: <server> <variable name="wlp.password.encryption.key" value="myKey" /> </server> Include it in the main Liberty configuration file. <server> ... <include location="/protected/key.xml" /> </server> Adjust the permissions on the key.xml to only users that need to access the file. In a test or development environment where a basic registry is used, the basic registry user passwords can be hashed to be stored in the Liberty configuration securityUtility encode --encoding=hash basicRegUserPassword |
| 10.3 | Ensure 'enableWelcomePage' is set to 'false' (Automated) | The WebSphere Liberty welcome page is enabled by default and is displayed when the root context “/” is accessed. | Perform the following to prevent Websphere Liberty from serving a welcome page from context root folder. Add the enableWelcomePage attribute to the httpDispatcher element to ${server.config.dir}/configDropins/overrides/\*.xml and set it to false. <httpDispatcher enableWelcomePage="false" /> |
| 10.4 | Ensure 'keysPassword' is set to a custom password for ltpa keys (Automated) | The LTPA keys are generated using a password, if a password is not provided, then a default password is used. | Add a custom encrypted password to the keysPassword attribute on the ltpa element in the Liberty configuration. <ltpa keysPassword="{aes}AE/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3e v396oYRw==" > |

## 10.5 Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications

Overview  
All CIS Benchmarks focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in conjunction with other essential cyber hygiene tasks like:  
• Monitoring the base operating system for vulnerabilities and quickly updating with  
the latest security patches  
• Monitoring applications and libraries for vulnerabilities and quickly updating with  
the latest security patches  
In the end, the CIS Benchmarks are designed as a key component of a comprehensive cybersecurity program.  
This document, Security Configuration Benchmark for IBM WebSphere Liberty, provides prescriptive guidance for establishing a secure configuration posture for IBM's Open Liberty and WebSphere Liberty. This guide was tested against Open Liberty as installed by the zip packages. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.  
Intended Audience  
This document is intended for system and application administrators, security specialists, auditors, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate IBM WebSphere Liberty.  
Consensus Guidance  
This CIS Benchmark was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.  
Each CIS Benchmark undergoes two phases of consensus review. The first phase occurs during initial Benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the Benchmark. This discussion occurs until consensus has been reached on Benchmark recommendations. The second phase begins after the Benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the Benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.  
Typographical Conventions  
The following typographical conventions are used throughout this guide:  
Convention  
Meaning  
Stylized Monospace font  
Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.  
Monospace font  
Used for inline code, commands, or examples. Text should be interpreted exactly as presented.  
<italic font in brackets>  
Italic texts set in angle brackets denote a variable requiring substitution for a real value.  
Italic font  
Note  
Used to denote the title of a book, article, or other publication.  
Additional information or caveats  
Recommendation Definitions  
The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted or the component will not be included in the recommendation.  
Title  
Concise description for the recommendation's intended configuration.  
Assessment Status  
An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:  
Automated  
Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.  
Manual  
Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.  
Profile  
A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.  
Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.  
Rationale Statement  
Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.  
Impact Statement  
Any security, functionality, or operational consequences that can result from following the recommendation.  
Audit Procedure  
Systematic instructions for determining if the target system complies with the recommendation  
Remediation Procedure  
Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

## 11.1 Maintain Standard Security Configurations for

Network Devices Maintain standard, documented security configuration standards for all authorized network devices.  
● ●  
● ●

## 11.2 Liberty Features Overview

Features are the discrete units of functionality by which you control the pieces of the runtime environment that are loaded into a particular server. By adding or removing features from your server configuration, you can control what functions the server can perform.  
Note that most recommendations listed will need a feature to be configured. The recommendations do not refer to the feature normally as features with new versions can be added in future, the features can be be included on other existing features or new features that can be in future. The Liberty documentation will have more information about this. For example, here is more information about Liberty features  
Appendix: Summary Table  
CIS Benchmark Recommendation  
1  
1.1  
1.2  
1.3  
1.4  
1.5  
1.6  
1.7  
1.8  
1.9  
2  
2.1  
2.2  
2.3  
Set Correctly  
Yes No  
  
  
  
  
Install and Setup  
Ensure root does not have ownership of Websphere Liberty binaries (Manual)  
Ensure extraneous files and directories are removed (Manual)  
Ensure only defined users have access to the file system (Manual)  
  
  
Ensure that only one user ID has write access to the WebSphere Liberty configuration files (Manual)  
Ensure Websphere Liberty Server Output is not set to the default value (Manual)  
Ensure automated configuration updates are disabled (Automated)  
Ensure the WebSphere Liberty Installation is Validated (Manual)  
Ensure Websphere Liberty file system access is Restricted (Manual)  
  
  
  
  
  
  
  
  
  
  
Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites (Automated)  
  
  
User Registries  
Ensure 'displayAuthenticationRealm' is set to 'false' (Automated)  
  
  
Ensure Basic Registry and Quick Start security Registry are Removed (Automated)  
  
  
Ensure that the LDAP connection uses TLS (Automated)   
  
CIS Benchmark Recommendation  
Set Correctly  
Yes No  
Application Deployment  
Ensure that automatic applications updates are disabled (Automated)  
  
  
Ensure JDK Security Manager is Enabled (Automated)   
  
Web Applications  
Securing Cookies  
3  
3.1  
3.2  
4  
4.1  
4.1.1  
Securing Session Cookies  
4.1.1.1  
4.1.1.2  
4.1.1.3  
4.1.1.4  
Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies (Automated)  
  
  
Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies (Manual)  
  
  
Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies. (Automated)  
  
  
Ensure 'cookieSecure' secure attribute is set to 'true' (Automated)  
  
  
4.1.2  
Securing Authentication Cookies  
4.1.2.1  
4.1.2.2  
4.1.2.3  
4.1.2.4  
4.1.2.5  
Ensure 'sameSiteCookie' attribute is set to 'Strict' (Manual)  
  
  
Ensure 'ssoDomainNames' attribute is configured for the authentication cookies. (Automated)  
  
  
Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie. (Automated)  
  
  
Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies (Automated)  
  
  
Ensure 'ssoCookieName' LTPA cookie name is set (Automated)  
  
  
CIS Benchmark Recommendation  
4.1.2.6  
4.1.2.7  
4.1.2.8  
Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies (Automated)  
Ensure 'trackLoggedOutSSOCookies' is set to 'true' (Automated)  
Ensure 'cookieName' JWT (JSON Web Token) cookie name is set (Automated)  
4.1.3  
Securing Other Cookies  
Set Correctly  
Yes No  
  
  
  
  
  
  
4.1.3.1  
Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies (Automated)  
  
  
4.2  
4.2.1  
4.2.2  
4.2.3  
4.2.4  
4.2.5  
4.2.6  
4.2.7  
4.2.8  
4.2.9  
Secure Transport  
Ensure 'trustDefaultCerts' is set to 'false' (Automated)  
Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security) (Automated)  
Ensure HSTS (HTTP Strict Transport Security) is enabled (Automated)  
Ensure that outbound TLS configurations are specified (Automated)  
Ensure that secure ciphers suites are configured (Automated)  
  
  
  
  
  
  
  
  
  
  
Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications (Automated)  
  
  
Ensure Hostname verification for TLS communication is enabled (Automated)  
  
  
Ensure that CA (Certificate Authority) certificates are used (Automated)  
  
  
Ensure 'ocsp.enable' certificate revocation is set to 'true' (Automated)  
  
  
4.2.10  
4.2.11  
4.2.12  
4.2.13  
4.2.14  
4.2.15  
4.3  
4.3.1  
4.3.2  
4.3.3  
4.3.4  
4.3.5  
4.3.6  
4.3.7  
CIS Benchmark Recommendation  
Ensure mutual TLS authentication is enabled (Automated)  
Ensure that strong algorithms are used for TLS certificates. (Manual)  
Ensure `httpPort` attribute set to `-1` (Automated)  
Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates (Manual)  
Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms. (Manual)  
Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules (Manual)  
Single Sign On (SSO)  
Set Correctly  
Yes No  
  
  
  
  
  
  
  
  
  
  
  
  
Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens (Automated)  
  
  
Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) (Manual)  
Ensure 'tokenReuse' is set to 'false' (Automated)  
Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party) (Automated)  
Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP) (Automated)  
  
  
  
  
  
  
  
  
Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP) (Automated)  
  
  
Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP) (Automated)  
  
  
4.3.8  
4.3.9  
4.3.10  
4.3.11  
4.3.12  
4.3.13  
4.3.14  
4.3.15  
4.3.16  
4.3.17  
4.4  
4.4.1  
4.4.2  
4.4.3  
CIS Benchmark Recommendation  
Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP) (Automated)  
Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP) (Automated)  
Set Correctly  
Yes No  
  
  
  
  
Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0 (Automated)  
  
  
Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0 (Automated)  
Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0 (Automated)  
Ensure 'httpsRequired' is set to 'true' in OAuth 2.0 (Automated)  
  
  
  
  
  
  
Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0 (Automated)  
  
  
Ensure 'httpsRequired' is set to 'true' in SAML (Automated)  
Enforce 'wantAssertionsSigned' to 'true' in SAML (Automated)  
Ensure 'authnRequestsSigned' is set to 'true' in SAML (Automated)  
  
  
  
  
  
  
General  
Ensure 'disableXPoweredBy' is set to 'true' (Automated)   
Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false' (Automated)  
  
  
  
Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects (Automated)  
  
  
4.4.4  
4.4.5  
4.4.6  
4.4.7  
4.4.8  
4.4.9  
4.4.10  
4.4.11  
4.4.12  
4.4.13  
4.4.14  
4.4.15  
4.4.16  
4.4.17  
CIS Benchmark Recommendation  
Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic (Manual)  
Ensure 'logoutOnHttpSessionExpire' is set to 'true' (Automated)  
Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic (Manual)  
Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic (Automated)  
Set Correctly  
Yes No  
  
  
  
  
  
  
  
  
Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic (Manual)  
  
  
Ensure ''trustedSensitiveHeaderOrigin'' is set to trusted host names and IP addresses for sensitive data (Automated)  
Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses (Automated)  
Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout (Automated)  
  
  
  
  
  
  
Ensure security constraints are specified to protect web applications (Automated)  
  
  
Ensure application security feature is enabled (Automated)  
Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false' (Automated)  
  
  
  
  
Ensure Web Server Document Root does not contain information that should be private (Automated)  
  
  
Ensure HTTP session overflow is 'disabled' (Manual)  
  
Ensure uncovered http methods are denied (Automated)   
  
  
CIS Benchmark Recommendation  
Ensure 'disallowServeServletsByClassName' is 'disabled' (Automated)  
Ensure server headers on requests are removed (Automated)  
Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications (Automated)  
Ensure 'default-error-page' is set for web applications (Manual)  
Ensure virtual hosts are defined to isolate applications (Automated)  
Ensure virtual hosts are Defined to isolate JMX communication and application traffic (Automated)  
Ensure whitelisting of virtual hosts to validate access based on originating endpoint (Automated)  
Enterprise Java Beans (EJB) Applications  
Set Correctly  
Yes No  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
The CSIv2 (Common Secure Interoperability version 2) serverPolicy  
Ensure 'sslEnabled' is set to 'true' within the CSIv2 Transport Layer (Automated)  
  
  
Ensure 'establishTrustInClient' is set to 'required' within the CSIv2 Authentication Layer (Automated)  
  
  
Ensure 'identityAssertionEnabled' is set to 'true' within the CSIv2 Attribute Layer (Automated)  
  
  
The CSIv2 (Common Secure Interoperability version 2) Client Policy  
Ensure 'sslEnabled' is set to 'true' within the CSIv2 TransportLayer - needsReview/Zech (Manual)  
  
  
4.4.18  
4.4.19  
4.4.20  
4.4.21  
4.4.22  
4.4.23  
4.4.24  
5  
5.1  
5.1.1  
5.1.2  
5.1.3  
5.2  
5.2.1  
CIS Benchmark Recommendation  
Ensure 'establishTrustInClient' is 'Required' for the CSIv2 Authentication Layer - needsReview/Zech (Manual)  
Set Correctly  
Yes No  
  
  
Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSIv2 Attribute Layer - review/Zech (Manual)  
  
  
Java Serialization  
Ensure filters are configured for Java serialization (JEP 290) (Manual)  
  
  
EJB Authentication  
Ensure that all appropriate EJB methods are protected (Automated)  
  
  
Web Services  
Ensure 'HttpsToken' is set in WS-Security policy (Automated)  
  
  
Ensured 'HashPassword' is set in UsernameToken WSSecurity policy (Automated)  
  
  
Ensure CallbackHandler is used to access private keys in keystore files (Manual)  
  
  
Ensure SOAP messages are Signed and encrypted with WS-Security policy (Manual)  
  
  
Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy (Manual)  
Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy (Automated)  
  
  
  
  
5.2.2  
5.2.3  
5.3  
5.3.1  
5.4  
5.4.1  
6  
6.1  
6.2  
6.3  
6.4  
6.5  
6.6  
6.7  
7  
7.1  
7.2  
7.3  
7.4  
7.5  
8  
8.1  
9  
9.1  
9.2  
9.3  
9.4  
10  
CIS Benchmark Recommendation  
Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications (Automated)  
Messaging  
Set Correctly  
Yes No  
  
  
Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names (Manual)  
Ensure the 'hostNameIncludeList attribute' is set to a whitelist of host names (Manual)  
Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames (Manual)  
Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses (Manual)  
Ensure the `useSSL` attribute is set to `true` for TLS Transport (Automated)  
  
  
  
  
  
  
  
  
  
  
MicroProfile Metrics  
Ensure 'authentication' is set to 'true' to protect the metrics end point (Automated)  
  
  
z/OS  
Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization (Automated)  
Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates (Automated)  
  
  
  
  
Ensure 'safkeyringhw:' is set to use a hardware crypto card (Manual)  
  
  
Ensure 'safRegistry' is configured (Automated)  
  
  
Miscellaneous  
CIS Benchmark Recommendation  
Ensure Unused Features are Removed (Automated)  
Ensure Passwords are Encrypted (Automated)  
Ensure 'enableWelcomePage' is set to 'false' (Automated)  
Set Correctly  
Yes No  
  
  
  
  
  
  
Ensure 'keysPassword' is set to a custom password for ltpa keys (Automated)  
  
  
Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications (Automated)  
  
  
Appendices  
Liberty configuration overview  
Liberty Features Overview  
10.1  
10.2  
10.3  
10.4  
10.5  
11  
11.1  
11.2  
Appendix: CIS Controls v7 IG 1 Mapped Recommendations  
1.1  
1.3  
1.4  
1.5  
1.7  
1.8  
1.9  
2.2  
4.4.1  
4.4.19  
4.4.20  
4.4.21  
5.3.1  
5.4.1  
8.1  
10.2  
10.3  
10.4  
10.5  
Recommendation  
Ensure root does not have ownership of Websphere Liberty binaries  
Set Correctly  
Yes No  
   
Ensure only defined users have access to the file system    
Ensure that only one user ID has write access to the WebSphere Liberty configuration files  
Ensure Websphere Liberty Server Output is not set to the

## 11.4 Install the Latest Stable Version of Any Securityrelated Updates on All Network Devices

Install the latest stable version of any security-related updates on all network  
devices.  
● ● ●

## 11.5 Manage Network Devices Using Multi-Factor

Authentication and Encrypted Sessions Manage all network devices using multi-factor authentication and encrypted sessions.  
● ●  
● ●

## 12.2 Establish and Maintain a Secure Network

Architecture Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.

## 12.3 Securely Manage Network Infrastructure

Securely manage network infrastructure. Example implementations include version-controlled-infrastructure-as-code, and the use of secure network protocols, such as SSH and HTTPS.

## 12.5 Centralize Network Authentication, Authorization,

and Auditing (AAA) Centralize network AAA.  
● ●  
Controls Version  
Control  
IG 1 IG 2 IG 3  
v7

## 12.6 Use of Secure Network Management and

Communication Protocols Use secure network management and communication protocols (e.g., 802.1X, Wi-Fi Protected Access 2 (WPA2) Enterprise or greater).

## 13.2 Remove Sensitive Data or Systems Not Regularly

Accessed by Organization Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.  
● ● ●

## 13.10 Perform Application Layer Filtering

Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.

## 14.4 Encrypt All Sensitive Information in Transit

Encrypt all sensitive information in transit.  
● ●  
● ●

## 14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.  
● ● ●

## 14.8 Encrypt Sensitive Information at Rest

Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.  
v8  
v7  
● ●  
●

## 16.2 Configure Centralized Point of Authentication

Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.  
● ●

## 16.5 Encrypt Transmittal of Username and

Authentication Credentials Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.  
● ●  
● ●

## 16.8 Separate Production and Non-Production Systems

Maintain separate environments for production and non-production systems.  
● ●  
Controls Version  
Control  
IG 1 IG 2 IG 3  
v7

## 16.11 Leverage Vetted Modules or Services for Application

Security Components Leverage vetted modules or services for application security components, such as identity management, encryption, and auditing and logging. Using platform features in critical security functions will reduce developers’ workload and minimize the likelihood of design or implementation errors. Modern operating systems provide effective mechanisms for identification, authentication, and authorization and make those mechanisms available to applications. Use only standardized, currently accepted, and extensively reviewed encryption algorithms. Operating systems also provide mechanisms to create and maintain secure audit logs.  
● ●

## 18.9 Separate Production and Non-Production Systems

Maintain separate environments for production and nonproduction systems. Developers should not have unmonitored access to production environments.  
● ●