

# CIS IBM WebSphere Liberty Benchmark

v1.0.0 - 05-13-2022

# Terms of Use

Please see the below link for our current terms of use:

<https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/>

# Table of Contents

<b>Terms of Use .....</b>	<b>1</b>
<b>Table of Contents .....</b>	<b>2</b>
<b>Overview .....</b>	<b>7</b>
Intended Audience.....	7
Consensus Guidance .....	8
Typographical Conventions.....	9
<b>Recommendation Definitions.....</b>	<b>10</b>
Title.....	10
Assessment Status.....	10
Automated .....	10
Manual.....	10
Profile .....	10
Description.....	10
Rationale Statement .....	10
Impact Statement.....	11
Audit Procedure.....	11
Remediation Procedure.....	11
Default Value.....	11
References .....	11
CIS Critical Security Controls® (CIS Controls®).....	11
Additional Information.....	11
Profile Definitions .....	12
Acknowledgements .....	13
<b>Recommendations .....</b>	<b>14</b>
<b>1 Install and Setup.....</b>	<b>14</b>
1.1 Ensure root does not have ownership of Websphere Liberty binaries (Manual) .....	15
1.2 Ensure extraneous files and directories are removed (Manual) .....	16
1.3 Ensure only defined users have access to the file system (Manual) .....	18
1.4 Ensure that only one user ID has write access to the WebSphere Liberty configuration files (Manual) .....	19
1.5 Ensure Websphere Liberty Server Output is not set to the default value (Manual) .....	21
1.6 Ensure automated configuration updates are disabled (Automated) .....	23
1.7 Ensure the WebSphere Liberty Installation is Validated (Manual) .....	25
1.8 Ensure Websphere Liberty file system access is Restricted (Manual) .....	27
1.9 Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites (Automated) .....	29
<b>2 User Registries .....</b>	<b>31</b>
2.1 Ensure 'displayAuthenticationRealm' is set to 'false' (Automated) .....	32

2.2 Ensure Basic Registry and Quick Start security Registry are Removed (Automated).....	33
2.3 Ensure that the LDAP connection uses TLS (Automated) .....	35
<b>3 Application Deployment.....</b>	<b>37</b>
3.1 Ensure that automatic applications updates are disabled (Automated) .....	38
3.2 Ensure JDK Security Manager is Enabled (Automated) .....	40
<b>4 Web Applications.....</b>	<b>42</b>
<b>4.1 Securing Cookies.....</b>	<b>43</b>
<b>4.1.1 Securing Session Cookies .....</b>	<b>44</b>
4.1.1.1 Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies (Automated) .....	45
4.1.1.2 Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies (Manual) .....	46
4.1.1.3 Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies. (Automated).....	47
4.1.1.4 Ensure 'cookieSecure' secure attribute is set to 'true' (Automated) .....	48
<b>4.1.2 Securing Authentication Cookies.....</b>	<b>49</b>
4.1.2.1 Ensure 'sameSiteCookie' attribute is set to 'Strict' (Manual) .....	50
4.1.2.2 Ensure 'ssoDomainNames' attribute is configured for the authentication cookies. (Automated) .....	51
4.1.2.3 Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie. (Automated) .....	52
4.1.2.4 Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies (Automated).....	53
4.1.2.5 Ensure 'ssoCookieName' LTPA cookie name is set (Automated).....	54
4.1.2.6 Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies (Automated) .....	55
4.1.2.7 Ensure 'trackLoggedOutSSOCookies' is set to 'true' (Automated) .....	56
4.1.2.8 Ensure 'cookieName' JWT (JSON Web Token) cookie name is set (Automated).....	57
<b>4.1.3 Securing Other Cookies .....</b>	<b>59</b>
4.1.3.1 Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies (Automated).....	60
<b>4.2 Secure Transport .....</b>	<b>61</b>
4.2.1 Ensure 'trustDefaultCerts' is set to 'false' (Automated).....	62
4.2.2 Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security) (Automated) .....	64
4.2.3 Ensure HSTS (HTTP Strict Transport Security) is enabled (Automated) .....	66
4.2.4 Ensure that outbound TLS configurations are specified (Automated) .....	68
4.2.5 Ensure that secure ciphers suites are configured (Automated).....	70
4.2.6 Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications (Automated) .....	72
4.2.7 Ensure Hostname verification for TLS communication is enabled (Automated).....	74
4.2.8 Ensure that CA (Certificate Authority) certificates are used (Automated).....	76
4.2.9 Ensure 'ocsp.enable' certificate revocation is set to 'true' (Automated).....	77
4.2.10 Ensure mutual TLS authentication is enabled (Automated) .....	78
4.2.11 Ensure that strong algorithms are used for TLS certificates. (Manual).....	80
4.2.12 Ensure `httpPort` attribute set to `-1` (Automated).....	81
4.2.13 Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates (Manual) .....	82
4.2.14 Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms. (Manual) .....	84
4.2.15 Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules (Manual) .....	85
<b>4.3 Single Sign On (SSO) .....</b>	<b>87</b>
4.3.1 Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens (Automated) .....	88
4.3.2 Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) (Manual) .....	89
4.3.3 Ensure 'tokenReuse' is set to 'false' (Automated) .....	90
4.3.4 Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party) (Automated) .....	91
4.3.5 Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP) (Automated) .....	92
4.3.6 Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP) (Automated) ...	93

4.3.7 Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP) (Automated).....	94
4.3.8 Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP) (Automated) .....	95
4.3.9 Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP) (Automated) .....	96
4.3.10 Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0 (Automated) .....	97
4.3.11 Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0 (Automated).....	98
4.3.12 Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0 (Automated).....	99
4.3.13 Ensure 'httpsRequired' is set to 'true' in OAuth 2.0 (Automated).....	100
4.3.14 Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0 (Automated).....	101
4.3.15 Ensure 'httpsRequired' is set to 'true' in SAML (Automated) .....	102
4.3.16 Enforce 'wantAssertionsSigned' to 'true' in SAML (Automated) .....	103
4.3.17 Ensure 'authnRequestsSigned' is set to 'true' in SAML (Automated) .....	104
<b>4.4 General.....</b>	<b>105</b>
4.4.1 Ensure 'disableXPoweredBy' is set to 'true' (Automated).....	106
4.4.2 Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false' (Automated) .....	108
4.4.3 Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects (Automated) .....	109
4.4.4 Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic (Manual) .....	110
4.4.5 Ensure 'logoutOnHttpSessionExpire' is set to 'true' (Automated) .....	111
4.4.6 Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic (Manual) .....	112
4.4.7 Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic (Automated) .....	113
4.4.8 Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic (Manual) .....	114
4.4.9 Ensure "trustedSensitiveHeaderOrigin" is set to trusted host names and IP addresses for sensitive data (Automated) .....	115
4.4.10 Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses (Automated) .....	116
4.4.11 Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout (Automated) .....	117
4.4.12 Ensure security constraints are specified to protect web applications (Automated).....	118
4.4.13 Ensure application security feature is enabled (Automated) .....	120
4.4.14 Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false' (Automated).....	121
4.4.15 Ensure Web Server Document Root does not contain information that should be private (Automated) .....	122
4.4.16 Ensure HTTP session overflow is 'disabled' (Manual).....	123
4.4.17 Ensure uncovered http methods are denied (Automated).....	124
4.4.18 Ensure 'disallowServeServletsByClassName' is 'disabled' (Automated).....	126
4.4.19 Ensure server headers on requests are removed (Automated) .....	127
4.4.20 Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications (Automated) .....	129
4.4.21 Ensure 'default-error-page' is set for web applications (Manual).....	131
4.4.22 Ensure virtual hosts are defined to isolate applications (Automated) .....	133
4.4.23 Ensure virtual hosts are Defined to isolate JMX communication and application traffic (Automated) .....	135
4.4.24 Ensure whitelisting of virtual hosts to validate access based on originating endpoint (Automated).....	137
<b>5 Enterprise Java Beans (EJB) Applications .....</b>	<b>139</b>
<b>5.1 The CSiv2 (Common Secure Interoperability version 2) serverPolicy .....</b>	<b>140</b>
5.1.1 Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer (Automated) .....	141
5.1.2 Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer (Automated) .....	143
5.1.3 Ensure 'identityAssertionEnabled' is set to 'true' within the CSiv2 Attribute Layer (Automated) .....	145
<b>5.2 The CSiv2 (Common Secure Interoperability version 2) Client Policy .....</b>	<b>147</b>
5.2.1 Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech (Manual).....	148

5.2.2 Ensure 'establishTrustInClient' is 'Required' for the CSv2 Authentication Layer - needsReview/Zech (Manual) .....	150
5.2.3 Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSv2 Attribute Layer - review/Zech (Manual) .....	152
<b>5.3 Java Serialization .....</b>	<b>154</b>
5.3.1 Ensure filters are configured for Java serialization (JEP 290) (Manual) .....	155
<b>5.4 EJB Authentication .....</b>	<b>157</b>
5.4.1 Ensure that all appropriate EJB methods are protected (Automated) .....	158
<b>6 Web Services .....</b>	<b>160</b>
6.1 Ensure 'HttpsToken' is set in WS-Security policy (Automated) .....	161
6.2 Ensure 'HashPassword' is set in UsernameToken WS-Security policy (Automated) .....	163
6.3 Ensure CallbackHandler is used to access private keys in keystore files (Manual) .....	165
6.4 Ensure SOAP messages are Signed and encrypted with WS-Security policy (Manual) .....	167
6.5 Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy (Manual) .....	169
6.6 Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy (Automated) .....	170
6.7 Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications (Automated) .....	172
<b>7 Messaging .....</b>	<b>174</b>
7.1 Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names (Manual) .....	175
7.2 Ensure the 'hostNameIncludeList' attribute is set to a whitelist of host names (Manual) .....	176
7.3 Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames (Manual) .....	177
7.4 Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses (Manual) .....	178
7.5 Ensure the 'useSSL' attribute is set to 'true' for TLS Transport (Automated) .....	179
<b>8 MicroProfile Metrics .....</b>	<b>180</b>
8.1 Ensure 'authentication' is set to 'true' to protect the metrics end point (Automated) .....	181
<b>9 z/OS .....</b>	<b>183</b>
9.1 Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization (Automated) .....	184
9.2 Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates (Automated) .....	186
9.3 Ensure 'safkeyringhw:' is set to use a hardware crypto card (Manual) .....	188
9.4 Ensure 'safRegistry' is configured (Automated) .....	191
<b>10 Miscellaneous .....</b>	<b>192</b>
10.1 Ensure Unused Features are Removed (Automated) .....	193
10.2 Ensure Passwords are Encrypted (Automated) .....	195
10.3 Ensure 'enableWelcomePage' is set to 'false' (Automated) .....	198
10.4 Ensure 'keysPassword' is set to a custom password for ltpa keys (Automated) .....	200
10.5 Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications (Automated) .....	202
<b>11 Appendices .....</b>	<b>204</b>
11.1 Liberty configuration overview .....	205
11.2 Liberty Features Overview .....	206
<b>Appendix: Summary Table .....</b>	<b>207</b>
<b>Appendix: CIS Controls v7 IG 1 Mapped Recommendations .....</b>	<b>217</b>
<b>Appendix: CIS Controls v7 IG 2 Mapped Recommendations .....</b>	<b>218</b>
<b>Appendix: CIS Controls v7 IG 3 Mapped Recommendations .....</b>	<b>221</b>

<b><i>Appendix: CIS Controls v7 Unmapped Recommendations.....</i></b>	<b><i>224</i></b>
<b><i>Appendix: CIS Controls v8 IG 1 Mapped Recommendations.....</i></b>	<b><i>228</i></b>
<b><i>Appendix: CIS Controls v8 IG 2 Mapped Recommendations.....</i></b>	<b><i>229</i></b>
<b><i>Appendix: CIS Controls v8 IG 3 Mapped Recommendations.....</i></b>	<b><i>232</i></b>
<b><i>Appendix: CIS Controls v8 Unmapped Recommendations.....</i></b>	<b><i>235</i></b>
<b><i>Appendix: Change History .....</i></b>	<b><i>238</i></b>

# 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](mailto: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
<code>Stylized Monospace font</code>	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
<code>Monospace font</code>	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<i>&lt;italic font in brackets&gt;</i>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
<i>Italic font</i>	Used to denote the title of a book, article, or other publication.
<b>Note</b>	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.

## Description

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.

## **Default Value**

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

## **References**

Additional documentation relative to the recommendation.

## **CIS Critical Security Controls® (CIS Controls®)**

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) "5.1 - Establish Secure Configurations" and (v8) "4.1 - Establish and Maintain a Secure Configuration Process" so individual recommendations will not be mapped to these safeguards.

## **Additional Information**

Supplementary information that does not correspond to any other field but may be useful to the user.

## Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

Items in this profile intend to:

- be practical and prudent;
- provide security focused best practice hardening of a technology; and
- limit impact to the utility of the technology beyond acceptable means.

- **Level 2**

This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

- are intended for environments or use cases where security is more critical than manageability and usability
- acts as defense in depth measure
- may impact the utility or performance of the technology
- may include additional licensing, cost, or addition of third party software

## Acknowledgements

This Benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

### **Contributor**

Tim Harrison  
Joern Krueger  
James Scott  
Mike Wicks  
Logan McMillan

### **Editor**

Ajay Reddy  
Berkas Ates  
Zech Hein  
Kristi Peterson  
Teddy Torres  
Gary Picher

# Recommendations

## 1 Install and Setup

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

## 1.1 Ensure root does not have ownership of Websphere Liberty binaries (Manual)

### Profile Applicability:

- Level 1

### Description:

Preventing the root user from owning files in the `${wlp.install.dir}` directory prevents unauthorized commands or files from being run.

### Rationale:

When directories or files are owned by the `root` user, administration of those files requires a user to log in as `root` or elevate their existing login to have `root` privileges. Performing operations as the `root` user could result in unauthorized commands that could alter files beyond the scope of WebSphere Liberty itself.

### Audit:

Check to ensure that no files or directories underneath `${wlp.install.dir}` are owned by the `root` user or `root` group:

```
ls -l -R ${wlp.install.dir} | awk '{print $3, $4}'
```







Expected result: You should not see the `root` user or `root` group referenced in the output.

### Remediation:

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}
```

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 Configure Data Access Control Lists</b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	<b>14.6 Protect Information through Access Control Lists</b> 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.			



## 1.2 Ensure extraneous files and directories are removed (Manual)

### Profile Applicability:

- Level 1

### Description:

The installation might provide example applications, documentation, and other directories which may not serve a production use.

### Rationale:

Removing non-production resources is a defense in depth measure that reduces potential exposures introduced by these resources.

### Audit:

Check for the existence of extraneous resources:

```
$ ls -l ${wlp.user.dir}/extension \  
  ${wlp.user.dir}/shared \  
  ${wlp.user.dir}/server \  
  ${wlp.user.dir}/client
```

There should be no non-production client extension/shared resource(s)/server/client in these folders

### Remediation:

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>
```

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.8 <u>Separate Production and Non-Production Systems</u> Maintain separate environments for production and non-production systems.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	<b>18.9 <u>Separate Production and Non-Production Systems</u></b> Maintain separate environments for production and nonproduction systems. Developers should not have unmonitored access to production environments.		●	●

## 1.3 Ensure only defined users have access to the file system (Manual)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

### Audit:

Ensure that no one falling into other category have access to the Websphere Liberty file system. You should expect to see ---.

```
ls -l -R ${wlp.install.dir} | awk '{print substr($1,length($1)-3,3)}'
```

### Remediation:

Ensure that `other` has no access to the system.

```
chmod -R o-rwx ${wlp.install.dir}
```

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>6.8 Define and Maintain Role-Based Access Control</b> 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.			●
v7	<b>4.3 Ensure the Use of Dedicated Administrative Accounts</b> 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.	●	●	●

## *1.4 Ensure that only one user ID has write access to the WebSphere Liberty configuration files (Manual)*

### **Profile Applicability:**

- Level 1

### **Description:**

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.

### **Rationale:**

WebSphere Liberty server administrators sometimes need the ability to write to server configuration files, but following the principle of least privilege they should not operate with write access unless absolutely necessary. Administrators can use `sudo` and the `/etc/sudoers` file to elevate their privilege when write access is needed, while operating with read access the rest of the time. Administrators should never share user IDs and passwords.







### **Audit:**

Verify that all WebSphere administrators have been added to the group that owns the server's configuration directory, which will grant them read but not write access. Ensure that the user that owns the server's configuration directory is a non-login user. Use `sudo` and the `/etc/sudoers` file to grant the administrators the ability to elevate their privilege to the user that owns the server's configuration directory when write access is required.

### **Remediation:**

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.

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> 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.			
v7	<b><u>4.3 Ensure the Use of Dedicated Administrative Accounts</u></b> 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.			

## 1.5 Ensure Websphere Liberty Server Output is not set to the default value (Manual)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Removing write access to a server's configuration files limits the damage that can be done by a theoretical attacker that is able to inject code to be run inside the WebSphere Liberty server process.

### Audit:

Ensure the existence of the `WLP_OUTPUT_DIR` variable in the `server.env` file. This environment variable is used to specify an alternative location for server generated output such as logs, the workarea directory, and other generated files. The variable should point to an output directory that is different from the server's configuration directory. The user ID of the WebSphere Liberty server process should have write access to the output directory but only read access to the configuration directory. For more information about `server.env` please see [here](#).

```
WLP_OUTPUT_DIR=/<server-writeable directory>/
```

Confirm that the output directory exists and has write access granted to the WebSphere Liberty server process user ID:

```
ls -l $WLP_OUTPUT_DIR | awk '{print $1, $3}'
```

Confirm that the WebSphere Liberty server process user ID has read access to its configuration files:

```
ls -l ${server.config.dir} | awk '{print $1, $3}'
```

### Remediation:

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.

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u></b> 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.			
v7	<b><u>4.3 Ensure the Use of Dedicated Administrative Accounts</u></b> 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.			

## 1.6 Ensure automated configuration updates are disabled (Automated)

### Profile Applicability:

- Level 1

### Description:

WebSphere Liberty provides the ability to automatically update the server runtime when the configuration changes, without requiring a server restart.

### Rationale:

Automatic updates during runtime are very useful during development and test phases. Configuration updates must be carefully controlled in production environments to reduce the possibility that unknown changes or vulnerabilities are deployed to users.

### Audit:

Ensure the `updateTrigger` attribute on the `config` element controls in the [Liberty configuration](#) is set to a value other than `polled`.

```
grep -w -R -i 'updateTrigger' ${server.config.dir}
```

### Remediation:

Add the `updateTrigger` attribute to the `config` element in `${server.config.dir}/configDropins/overrides/*.xml` and set to `mbean` or `disabled`.

```
<config updateTrigger="mbean" />
```

### Default Value:

Liberty enables dynamic configuration updates by default.

The default settings for configuration monitoring are as follows:



```
<config updateTrigger="polled" monitorInterval="500ms"/>
```

### References:

1. [https://openliberty.io/docs/latest/reference/config/server-configuration-overview.html#\\_dynamic\\_updates](https://openliberty.io/docs/latest/reference/config/server-configuration-overview.html#_dynamic_updates)
2. [https://www.ibm.com/docs/en/was-liberty/base?topic=SSEQTP\\_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp\\_setup\\_dyn\\_upd.html](https://www.ibm.com/docs/en/was-liberty/base?topic=SSEQTP_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_setup_dyn_upd.html)
3. <https://openliberty.io/docs/latest/reference/config/config.html>



**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u></b> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			

## 1.7 Ensure the WebSphere Liberty Installation is Validated (Manual)

### Profile Applicability:

- Level 1

### Description:

Ensure that all WebSphere Liberty binaries were installed successfully and that you are running the latest fix pack version.

### Rationale:

Ensuring that all WebSphere Liberty binaries are installed successfully and that the latest fix pack version is applied provides the most up-to-date protection against vulnerabilities.

### Audit:

Validate your WebSphere Liberty installation by running the command below:

```
${wlp_install_dir}/bin/productInfo validate
```

Expected result:

```
Start product validation...  
Product validation completed successfully.
```

Validate that you are running with the latest fixpack.

```
${wlp_install_dir}/bin/productInfo version
```

Expected result:






```
Product version: YY.0.0.FP
```

In the output, YY is the last two digits of the year that the fixpack was released, and FP represents the fixpack number for that year. Ensure that this matches that latest fixpack available from IBM.

### Remediation:

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.

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>7.7 <u>Remediate Detected Vulnerabilities</u></b> Remediate detected vulnerabilities in software through processes and tooling on a monthly, or more frequent, basis, based on the remediation process.			
v7	<b>11.4 <u>Install the Latest Stable Version of Any Security-related Updates on All Network Devices</u></b> Install the latest stable version of any security-related updates on all network devices.			

## 1.8 Ensure Websphere Liberty file system access is Restricted (Manual)

### Profile Applicability:

- Level 1

### Description:

The permissions of files and directories underneath `${wlp.install.dir}` should follow the principle of least privilege.

### Rationale:

If the permissions of files and directories underneath `${wlp.install.dir}` are too broad, unintended users may be able to modify or read those files. Write permission should be limited to a single administrator, read permission should be limited to a specifically-defined group containing users that require that access, and all other users should have their access entirely removed.

### Impact:

If the file permissions are too broad, a security vulnerability exists where unintended users will have the ability to modify and read WebSphere Liberty files.

### Audit:

Check the permissions of `${wlp.install.dir}`:

```
ls -l -R ${wlp.install.dir} | awk '{print $1}'
```







Expected result: All files should show permissions of the form `(d)rw(x)r-(x)---`. The `d`(directory) and `x`(executable) flags may or may not exist depending on the file type, however the `r`(read) and `w`(write) flags should always appear in the same position as the above output.

### Remediation:

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}
```

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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.			

## 1.9 Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Use additional configuration files to hold sensitive configuration information. Restrict access to files with sensitive information. This avoids unauthorized access to information such as passwords.

Use additional configuration files to organization configuration information by type. This helps users only update their portion of the server configuration and not accidentally or intentionally change configuration information in other areas.

Included files should also be ignored if there is a merge conflict with the main file. This prevents a user from accidentally or intentionally overriding information in the main configuration file.

To protect important configuration settings from merges or changes, information can be set in the `${server.config.dir}/configDropins/overrides/`.

### Audit:

Ensure that the `onConflict` attribute on all the `location` elements and is set to `IGNORE` in the [Liberty configuration](#).

```
grep -w -R -i 'include location' ${server.config.dir}
```

Review the file permissions on included files.

```
ls -al <includedFileName.xml>
```

Review the main [Liberty configuration](#) file for any sensitive or grouped elements that can be moved to an included file.

### Remediation:

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.







**Default Value:**

In WebSphere Liberty, the onConflict attribute in the include element is `MERGE` by default.

**References:**

1. <https://openliberty.io/docs/latest/reference/config/include.html>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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.			

## **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.



## 2.1 Ensure 'displayAuthenticationRealm' is set to 'false' (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Do not display the user registry information when prompting the user for credentials to avoid showing sensitive information like the LDAP host and port.

### Audit:

Ensure the `displayAuthenticationRealm` attribute for first `webAppSecurity` element is set to `false` in the [Liberty configuration](#).

```
grep -w -R -i 'displayAuthenticationRealm' ${server.config.dir}
```

### Remediation:

Set the `displayAuthenticationRealm` attribute in the `webAppSecurity` element to `false` in `${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<webAppSecurity ... displayAuthenticationRealm="false" />
```

### Default Value:

The default value of the `displayAuthenticationRealm` attribute in the `webAppSecurity` element is set to `false`.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=applications-configuring-web-security-related-properties-in-liberty>

## 2.2 Ensure Basic Registry and Quick Start security Registry are Removed (Automated)

### Profile Applicability:

- Level 1

### Description:

The Basic Registry and Quick Start Security user registries are indented for developing and testing environments.

### Rationale:

The Basic and Quick Start Security user registries are not robust enough for production to be used. Registries like LDAP, Database or a custom user registry should be used for production.

### Audit:

Ensure there are no uses of `basicRegistry` or `quickStartSecurity` elements in the [Liberty configuration](#).

```
grep -w -i -R 'basicRegistry' ${server.config.dir}
grep -w -i -R 'quickStartSecurity' ${server.config.dir}
```






### Remediation:




Use robust user registries, such as an LDAP registry or a Custom registry for production.

### References:

1. <https://openliberty.io/docs/latest/reference/feature/appSecurity-2.0.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 Configure Data Access Control Lists</b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v8	<b>6.7 Centralize Access Control</b> Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<p><b>14.6 <u>Protect Information through Access Control Lists</u></b></p> <p>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.</p>			

## 2.3 Ensure that the LDAP connection uses TLS (Automated)

### Profile Applicability:

- Level 1

### Description:

TLS (Transport Layer Security) provides secure communication over a network.

### Rationale:

Secure the data sent to the LDAP server which can include user authentication and search requests.

### Audit:

Ensure that the `sslEnabled` attribute of all `ldapRegistry` elements is to `true`. The `sslRef` of all `ldapRegistry` elements must match the `id` attribute of an `ssl` element. Both configurations are set on the [Liberty configuration](#).

```
grep -w -R -i 'sslEnabled' ${server.config.dir}
```

The `sslEnabled` attribute may match to elements in addition to the `ldapRegistry` element.

### Remediation:

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" />
```





### Default Value:

The default value of the `sslEnabled` attribute on the `ldapRegistry` element is `false`.

### References:

1. <https://openliberty.io/docs/latest/reference/config/ldapRegistry.html>
2. <https://openliberty.io/docs/latest/reference/config/ssl.html>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>16.5 <u>Encrypt Transmittal of Username and Authentication Credentials</u></b> Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.			

### **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.

### 3.1 Ensure that automatic applications updates are disabled (Automated)

#### Profile Applicability:

- Level 1

#### Description:

Automatic application updates will dynamically update the runtime behavior of the applications of any changes to the application.

#### Rationale:

Automatic updates during runtime are very useful during development and test phases. Application updates must be carefully controlled in production environments to reduce the possibility that unknown changes or vulnerabilities are deployed to users.

#### Audit:

Review the `updateTrigger` attribute and `dropinsEnabled` attributes in the `applicationMonitor` element in the [Liberty configuration](#).

Ensure the `updateTrigger` attribute is set to a value other than `polled`.

Ensure the `dropinsEnabled` attribute is set to `false`.

```
grep -w -R -i 'updateTrigger' ${server.config.dir}
grep -w -R -i 'dropinsEnabled' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

Liberty enables dynamic application by default.

The default settings for application monitoring are as follows:

```
<applicationMonitor updateTrigger="polled" pollingRate="500ms"
    dropins="dropins" dropinsEnabled="true"/>
```

## References:

1. <https://openliberty.io/docs/latest/reference/config/applicationMonitor.html>
2. [https://www.ibm.com/docs/en/was-liberty/base?topic=SSEQTP\\_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp\\_setup\\_dyn\\_upd.html](https://www.ibm.com/docs/en/was-liberty/base?topic=SSEQTP_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_setup_dyn_upd.html)



## 3.2 Ensure JDK Security Manager is Enabled (Automated)

### Profile Applicability:

- Level 2

### Description:

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.

### Rationale:

In some situations there is a need to restrict applications from performing certain operations like read and writing to files, opening network connections etc. JDK's security manager provides a way to configure a security policy that can be applied to restrict the applications operations.

Note: The security manager is proposed to be deprecated and removed in future releases of JDK.

### Impact:

Generally enabling the security manager can impact the performance because of the additional checks needed to restrict operations by applications. Applications can also fail to run with these additional checks in which case appropriate permissions have to be configured for the application either in the server configuration files or the application specific permissions.xml file.

### Audit:

Ensure that the `websphere.java.security` property is configured 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 in [Liberty configuration](#)

```
<javaPermission ... />
```

### Remediation:

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" />
```

**Default Value:**

Security Manager is not enabled by default.

**References:**

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=security-java-2>
2. <https://openjdk.java.net/jeps/411>
3. <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/SecurityManager.html>

## 4 Web Applications

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

## 4.1 Securing Cookies

### 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

#### 4.1.1.1 Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies (Automated)

##### Profile Applicability:

- Level 1

##### Description:

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.

##### Rationale:

Some browsers treat cookies without a `SameSite` attribute as if they have the `SameSite` attribute value of `Lax`.

##### Audit:

Ensure the `cookieSameSite` attribute is set to `Strict` in the `httpSession` element in the [Liberty configuration](#)

```
grep -w -R -i 'cookieSameSite' ${server.config.dir}
```

##### Remediation:

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"/>
```

##### Default Value:

The default value is `Disabled`.

##### References:

1. <https://openliberty.io/blog/2020/03/25/set-samesite-attribute-cookies-liberty.html>

### 4.1.1.2 Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies (Manual)

#### Profile Applicability:

- Level 1

#### Description:

The HttpOnly attribute on a cookie prevents the cookie from being accessed by the client side scripts.

#### Rationale:

Enabling HttpOnly attribute mitigates Cross-Site Scripting (XSS) attacks.

#### Audit:

Ensure the `cookieHttpOnly` attribute in the `httpSession` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'cookieHttpOnly' ${server.config.dir}
```

#### Remediation:

Set the `cookieHttpOnly` attribute to `true` in the `httpSession` element in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<httpSession cookieHttpOnly="true" />
```

#### Default Value:

The default value is set to `true`

#### References:

1. <https://openliberty.io/docs/latest/reference/config/httpSession.html>

#### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.		●	●

### 4.1.1.3 Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies. (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The domain name attribute in a cookie specifies which hosts can receive the cookie.

#### Rationale:

Ensure the JSESSIONID cookie is sent to the correct domain by configuring the domain name. This will prevent the cookie to be sent to servers in other domains.

#### Audit:

Ensure the `cookieDomain` attribute in the `httpSession` element is set to the appropriate domain in the [Liberty configuration](#)

```
grep -w -R -i 'cookieDomain' ${server.config.dir}
```

#### Remediation:

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" />
```

#### References:

1. <https://openliberty.io/docs/latest/reference/config/httpSession.html>



#### 4.1.1.4 Ensure 'cookieSecure' secure attribute is set to 'true' (Automated)

##### Profile Applicability:

- Level 1

##### Description:

The secure flag on a cookie will restrict the browser to send the cookies only on encrypted channels like HTTPS.

##### Rationale:

Cookies with the secure flag will only be sent over encrypted HTTPS requests.

##### Audit:

Ensure `cookieSecure` is set to `true` in the `httpSession` element in the [Liberty configuration](#)

```
grep -w -R -i 'cookieSecure' ${server.config.dir}
```

##### Remediation:

Set the `cookieSecure` attribute to `true` in the `httpSession` element in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<httpSession cookieSecure="true"/>
```





##### Default Value:

The default value is `false`

##### References:

1. <https://openliberty.io/docs/latest/reference/config/httpSession.html>

##### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

### 4.1.2 Securing Authentication Cookies

Security cookies used for authentication including `LTPA` and `JWT` cookies

### 4.1.2.1 Ensure 'sameSiteCookie' attribute is set to 'Strict' (Manual)

#### Profile Applicability:

- Level 1

#### Description:

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.

#### Rationale:

Cookies without a `SameSite` attribute are treated as if they have the `SameSite` attribute value of `Lax` for some browsers.

#### Audit:

Ensure the `sameSiteCookie` attribute is set to `Strict` in the `webAppSecurity` element in the [Liberty configuration](#)

```
grep -w -R -i 'sameSiteCookie' ${server.config.dir}
```

#### Remediation:

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"/>
```

#### Default Value:

The default value is `Disabled`.

#### References:

1. <https://openliberty.io/blog/2020/03/25/set-samesite-attribute-cookies-liberty.html>

### 4.1.2.2 Ensure 'ssoDomainNames' attribute is configured for the authentication cookies. (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The domain name attribute in a cookie specifies which hosts can receive the cookie.

#### Rationale:

Cookies that have their domain attribute set to a specific domain will only be sent to the hosts in that domain or sub-domain which helps in controlling its propagation.

#### Audit:

Ensure the `ssoDomainNames` attribute in the `webAppSecurity` element is set to the appropriate domain name in the [Liberty configuration](#)

```
grep -w -R -i 'ssoDomainNames' ${server.config.dir}
```

#### Remediation:

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"/>
```

#### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

### 4.1.2.3 Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie. (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The secure flag on a cookie will restrict the browser to send the cookies only on encrypted channels like HTTPS.

#### Rationale:

Cookies with the secure attribute set will only be sent over encrypted HTTPS requests.

#### Audit:

Ensure the `setCookieSecureFlag` attribute in the `jwtSso` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'setCookieSecureFlag' ${server.config.dir}
```

#### Remediation:

Set the `setCookieSecureFlag` attribute to `true` in the `jwtSso` element in `${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<jwtSso setCookieSecureFlag="true"/>
```

#### Default Value:

The default value is `true`.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/jwtSso.html>

#### 4.1.2.4 Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies (Automated)

##### Profile Applicability:

- Level 1

##### Description:

Cookies with the secure flag will only be sent over encrypted HTTPS requests.

##### Rationale:

Transport cookies over a secure TLS connection to avoid clear text transmission of the cookie information. A stolen cookie by a third-party intruder can allow them to act as that user until it expires

##### Audit:

Ensure the `ssoRequiresSSL` attribute is set to `true` in the `webAppSecurity` element in the [Liberty configuration](#)

```
grep -w -R -i 'ssoRequiresSSL' ${server.config.dir}
```

##### Remediation:





Set the `ssoRequiresSSL` attribute is set to `true` in the `webAppSecurity` element on `${server.config.dir}/configDropins/overrides/*.xml`

```
<webAppSecurity ssoRequiresSSL="true" />
```

##### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

##### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 Encrypt Sensitive Data in Transit</b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 Encrypt All Sensitive Information in Transit</b> Encrypt all sensitive information in transit.			

### 4.1.2.5 Ensure 'ssoCookieName' LTPA cookie name is set (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The server will authenticate only the LTPA cookie name that is configured. When the request contains other LTPA cookie names, they will be ignored.

#### Rationale:

Changing the LTPA cookie name helps make the cookie uniquely usable across multiple Liberty servers. It also helps hide the intended use of the cookie which helps prevent a bad actor from knowing how to try and misuse the cookie.

#### Audit:

Ensure the `ssoCookieName` attribute in the `webAppSecurity` is set to something other than `LtpaToken2` in the [Liberty configuration](#)

```
grep -w -R -i 'ssoCookieName' ${server.config.dir}
```

Also, Ensure the `useOnlyCustomCookieName` attribute in the `webAppSecurity` is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'useOnlyCustomCookieName' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

- The `ssoCookieName` attribute has a default value of `LtpaToken2`.
- The `useOnlyCustomCookieName` attribute is set to `false` by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

#### 4.1.2.6 Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies (Automated)

##### Profile Applicability:

- Level 1

##### Description:

The HttpOnly attribute on a cookie prevents the cookie from being accessed by the client side scripts.

##### Rationale:

Preventing client-side scripts from accessing authentication cookies helps prevent Cross-Site Scripting (XSS) attacks.

##### Audit:

Ensure the `httpOnlyCookies` attribute in the `webAppSecurity` is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'httpOnlyCookies' ${server.config.dir}
```

##### Remediation:

Set the `httpOnlyCookies` attribute to `true` in the `webAppSecurity` in the `${server.config.dir}/configDropins/overrides/*.xml`

```
<webAppSecurity httpOnlyCookies="true" />
```

##### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>



### 4.1.2.7 Ensure 'trackLoggedOutSSOCookies' is set to 'true' (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The `trackLoggedOutSSOCookies` attribute keeps track of the LTPA cookies that are logged out in a running server.

#### Rationale:

Prevent the misuse of LTPA tokens after users have logged out.

#### Audit:

Ensure the `trackLoggedOutSSOCookies` attribute in the `webAppSecurity` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'trackLoggedOutSSOCookies' ${server.config.dir}
```

#### Remediation:

Set `trackLoggedOutSSOCookies` to `true` in the `webAppSecurity` element in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<webAppSecurity trackLoggedOutSSOCookies="true" />
```

#### Default Value:

The default value of `trackLoggedOutSSOCookies` is `false`

#### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

#### 4.1.2.8 Ensure 'cookieName' JWT (JSON Web Token) cookie name is set (Automated)

##### Profile Applicability:

- Level 1

##### Description:

The server will authenticate only the JWT cookie name that is configured. When the request contains other JWT cookie names, they will be ignored.

##### Rationale:

Changing the JWT cookie name helps make the cookie uniquely usable across multiple Liberty servers. It also helps hide the intended use of the cookie which helps prevent a bad actor from knowing how to try and misuse the cookie.

##### Audit:

Ensure the `cookieName` attribute in the `jwtSso` element is set to anything except `JWT` in the [Liberty configuration](#)

```
grep -w -R -i 'cookieName' ${server.config.dir}
```

Also, Ensure the `useOnlyCustomCookieName` attribute in the `webAppSecurity` is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'useOnlyCustomCookieName' ${server.config.dir}
```

##### Remediation:

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" />
```

##### Default Value:

- The `cookieName` attribute has a default value of `JWT`.
- The `useOnlyCustomCookieName` attribute is set to `false` by default.

**References:**

1. <https://openliberty.io/docs/latest/reference/config/jwtSso.html>
2. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

### **4.1.3 Securing Other Cookies**

### 4.1.3.1 Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies (Automated)

#### Profile Applicability:

- Level 1

#### Description:

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.

#### Rationale:

Cookies without a `SameSite` attribute are treated as if they have the `SameSite` attribute value of `Lax` for some browsers.

#### Audit:

Ensure all other defined cookie names are added to the `Strict` attribute comma-separated list (or `*` is used to include all cookies) in the `samesite` element used by the `httpEndpoint` element in the [Liberty configuration](#).

```
grep -w -R -i 'samesite' ${server.config.dir}
```

#### Remediation:

Set the `strict` attribute to `*` in the `samesite` element in the `httpEndpoint` element in `${server.config.dir}/configDropins/overrides/*.xml`.

```
<httpEndpoint
...
  <samesite strict="*" />
</httpEndpoint>
```

#### References:

1. <https://openliberty.io/blog/2020/03/25/set-samesite-attribute-cookies-liberty.html>

## 4.2 Secure Transport

## 4.2.1 Ensure 'trustDefaultCerts' is set to 'false' (Automated)

### Profile Applicability:

- Level 2

### Description:

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.

### Rationale:

Restricting the certificates trusted by the server to only the SSL/TLS configurations in the server in a production environment eliminates the risk of trusting other certificates when making outbound secure connections.

### Audit:

Ensure the `trustDefaultCerts` attribute is set to `false` in all `ssl` elements in the [Liberty configuration](#).

```
grep -w -R -i 'trustDefaultCerts' ${server.config.dir}
```

### Remediation:

Add the `trustDefaultCerts` attribute to all `ssl` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set the value to `false`.

```
<ssl trustDefaultCerts="false" />
```



### Default Value:



The default configuration templates provided by WebSphere Liberty trust public certificates from the Java runtime by default, the templates set the `trustDefaultCerts` attribute to `true`.

### References:

1. <https://openliberty.io/docs/latest/reference/feature/transportSecurity-1.0.html>
2. <https://openliberty.io/docs/latest/reference/config/ssl.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 Encrypt Sensitive Data in Transit</b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			



## 4.2.2 Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security) (Automated)

### Profile Applicability:

- Level 1

### Description:

The latest versions of TLS provide drop support for less secure cryptographic features and add support for more advanced cryptographic algorithms.

### Rationale:

TLS 1.2 and higher versions are recommended for secure communication.

### Audit:

Review the TLS version level set by the of the `sslProtocol` attribute in all `ssl` elements in the [Liberty configuration](#).

```
grep -w -R -i 'sslProtocol' ${server.config.dir}
```

### Remediation:

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" />
```



### Default Value:



In WebSphere Liberty, when using the IBM JRE the default version is `SSL_TLSv2` and when using the Oracle JRE the default value is `SSL` when transport security is enabled.

### References:

1. <https://openliberty.io/docs/latest/reference/config/ssl.html>
2. <https://openliberty.io/docs/latest/reference/feature/transportSecurity-1.0.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 Encrypt Sensitive Data in Transit</b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

### 4.2.3 Ensure HSTS (HTTP Strict Transport Security) is enabled (Automated)

#### Profile Applicability:

- Level 1

#### Description:

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.

#### Rationale:

Enabling HSTS helps mitigate passive eavesdropper and active man-in-the-middle (MITM) attacks.

#### Audit:

Ensure the `addstricttransportsecurityheader` attribute is set in the `webContainer` element in the [Liberty configuration](#)

```
grep -w -R -i 'addstricttransportsecurityheader' ${server.config.dir}
```

If it is not set in the Liberty configuration above, then check if it is set for individual apps in the `web.xml` files for each application.

```
grep -w -R -i 'ADD_STS_HEADER_WEBAPP' ${server.config.dir}
```

#### Remediation:





Set the `addstricttransportsecurityheader` attribute in the `webContainer` element in the `${server.config.dir}/configDropins/overrides/*.xml` as follows:

```
<webContainer addstricttransportsecurityheader="max-age=31536000;includeSubDomains" />
```

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=slia-securing-liberty-by-using-http-strict-transport-security-hsts>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 4.2.4 Ensure that outbound TLS configurations are specified (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Configure additional TLS configuration elements for outbound connections with listed hostnames and ports allows to separate the TLS connection configurations for outbound calls for more fine grain control.

### Audit:

Ensure the `outboundSSLRef` attribute is set to a valid `ssl` configuration on the `sslDefault` element in the [Liberty configuration](#).

```
grep -w -R -i 'outboundSSLRef' ${server.config.dir}
```

Also ensure that `host` and `port` attributes are set on `outboundConnection` elements for all `ssl` elements used for outbound requests.

```
grep -w -R -i 'outboundConnection' ${server.config.dir}
```

### Remediation:

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>
```

## References:

1. <https://openliberty.io/docs/latest/reference/config/ssl.html>
2. <https://openliberty.io/docs/latest/reference/feature/transportSecurity-1.0.html>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 4.2.5 Ensure that secure ciphers suites are configured (Automated)

### Profile Applicability:

- Level 1

### Description:

A cipher suite includes a set of algorithms used when making secure TLS connections. Strong cipher suites contain more secure algorithms.

### Rationale:

WebSphere Liberty provides groups of different strength cipher suites to use for the TLS handshake. The highest cipher group or a custom cipher list should be used for secure communication.

### Audit:

Ensure the `securityLevel` attribute is set to `HIGH` in all `ssl` elements in the [Liberty configuration](#).

```
grep -w -R -i 'securityLevel' ${server.config.dir}
```

Or ensure the `enabledCiphers` attribute is set to a customized list of ciphers in all `ssl` elements. The `enabledCiphers` attribute overrides the `securityLevel` attribute.

```
grep -w -R -i 'enabledCiphers' ${server.config.dir}
```

For either configuration above, ensure the `enforceCipherOrder` attribute is set to `true`.

```
grep -w -R -i 'enforceCipherOrder' ${server.config.dir}
```

### Remediation:

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" />
```

**Default Value:**

WebSphere Liberty sets the `securityLevel` attribute to `HIGH` and the `enforceCipherOrder` attribute is set to `false` by default.

**References:**

1. <https://openliberty.io/docs/latest/reference/config/ssl.html>
2. <https://www.ibm.com/docs/en/sdk-java-technology/8?topic=suites-cipher>
3. <https://docs.oracle.com/en/java/javase/11/docs/specs/security/standard-names.html#jsse-cipher-suite-names>



## 4.2.6 Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Applications should set the transport guarantee to CONFIDENTIAL to enforce TLS secure communication when accessing the application.

### Audit:

Ensure the transport-guarantee attribute in the user-data-constraint element on the security-constraint element in all applications' /WEB-INF/web.xml files is set to CONFIDENTIAL. Applications may also be in the \${server.config.dir}/dropins directory or a custom location.

```
grep -w -R -i 'transport-guarantee' ${wlp.user.dir}/shared/apps/
```

For legacy web applications, the transport guarantee can also be set in the ibm-web-ext.xml or ibm-web-ext.xml files.

**Note:** The security-constraints can also be set in the code using annotations like @ServletSecurity.TransportGuarantee

### Remediation:

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





### Default Value:

WebSphere Liberty does not set a transport guarantee for applications by default.

## References:

1. <https://openliberty.io/docs/latest/application-configuration-hardening.html>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 4.2.7 Ensure Hostname verification for TLS communication is enabled (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Hostname verification mitigates man-in-the-middle security vulnerability attacks.

### Audit:

Ensure the `verifyHostname` attribute is set to `true` in the SSL configuration in the [Liberty configuration](#).

```
grep -w -R 'verifyHostname' ${server.config.dir}
```

For JAX-RS client applications ensure the `disableCNCheck` attribute is set to `false` in the `webTarget` element.

```
grep -w -R 'disableCNCheck' ${server.config.dir}
```

### Remediation:

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" />
```

### Default Value:

- The default value of the `verifyHostname` attribute is `false`
- The default value of the `disableCNCheck` attribute is `false`

## References:

1. <https://openliberty.io/blog/2019/06/21/microprofile-rest-client-19006.html>
2. <https://openliberty.io/docs/latest/reference/config/ssl.html>

## 4.2.8 Ensure that CA (Certificate Authority) certificates are used (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Using trusted Certificate Authority (CA) signed certificates for TLS communications mitigates against using untrusted or revoked certificates and eliminates warning messages in the browser.

### Audit:





Ensure that `Owner` and `Issuer` values are not the same, which typically implies that the certificate is not be self signed. Other information in the certificate can also confirm this.

```
keytool -list -v -keystore <keystore file name> -storepass <store password> -storetype <store type> | grep 'Owner\\|Issuer'
```

### Remediation:

Add non self signed CA certificates as described [here](#).

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

## 4.2.9 Ensure 'ocsp.enable' certificate revocation is set to 'true' (Automated)

### Profile Applicability:

- Level 1

### Description:

Certificate revocation is the process of canceling the digital certificate of the revoked user and keeping track of them.

### Rationale:

Enabling certificate revocation prevents use of revoked certificates.

### Audit:

Ensure the `ocsp.enable` variable is set to `true` in `java.security` file.

```
grep 'ocsp.enable' $JAVA_HOME/jre/lib/security/java.security
```

### Remediation:

Add `ocsp.enable=true` in `$JAVA_HOME/jre/lib/security/java.security` file.

```
ocsp.enable=true
```

### Default Value:

Default value depends on JDK distribution.

## 4.2.10 Ensure mutual TLS authentication is enabled (Automated)

### Profile Applicability:

- Level 2

### Description:

Mutual TLS authentication requires that both the server and the client authenticate to the other during SSL/TLS handshake.

### Rationale:

Mutual authentication, also known as two-way authentication, is certificate-based authentication for clients. Use of mutual TLS authentication whenever possible is recommended.

### Audit:

Ensure the `clientAuthentication` attribute in the `ssl` element is set to `true` and the `trustedHeaderOrigin` attribute in `httpDispatcher` element is set to appropriate value in the [Liberty configuration](#).

```
grep -w -R 'clientAuthentication' ${server.config.dir}
grep -w -R 'trustedHeaderOrigin' ${server.config.dir}
```

### Remediation:

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" />
```

### References:

1. <https://www.ibm.com/support/pages/how-setup-liberty-use-certificate-based-authentication>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			



## 4.2.11 Ensure that strong algorithms are used for TLS certificates. (Manual)

### Profile Applicability:

- Level 1

### Description:

Certificates created with stronger algorithms use stronger hashes which is more secure. Also, some browsers warn when using weak certificates.

### Rationale:

With increased CPU speeds and computing power of computers, some algorithms are not secure to use in production systems anymore such as SHA1 or MD5. On top of that, even for accepted algorithms such as SHA256 or higher, key sizes need to be higher than 2048 bits. Weak algorithms and small key sizes are not secure.

### Audit:

Ensure that Signature algorithm name is SHA256 or higher and Subject Public Key Algorithm is equal or higher than 2048 bit.

```
keytool -list -v -keystore <keystore file name> -storepass <store password> -  
storetype <store type> | grep 'Signature algorithm name\|Subject Public Key  
Algorithm'  
...  
Signature algorithm name: SHA256withRSA  
Subject Public Key Algorithm: 2048-bit RSA key  
...
```

### Remediation:

Create certificates with SHA256 or higher algorithm and 2048 or higher key bit size.

## 4.2.12 Ensure `httpPort` attribute set to `-1` (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Disabling the HTTP port forces communication to use the HTTPS port which is encrypted to give better defense against man-in-the-middle attacks.

### Audit:

Ensure the `httpPort` attribute in the `httpEndpoint` element is set to "-1" in [Liberty configuration](#).

```
grep -w -R 'httpPort' ${server.config.dir}
```

### Remediation:





Set the `httpPort` attribute to -1 in the `httpEndpoint` element in `${server.config.dir}/configDropins/overrides/*.xml`.

```
<httpEndpoint ... httpPort="-1"/>
```

### References:

1. <https://openliberty.io/docs/latest/reference/feature/transportSecurity-1.0.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 4.2.13 Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates (Manual)

### Profile Applicability:

- Level 2

### Description:

Hardware security modules (HSMs) are hardened, tamper-resistant hardware devices that secure cryptographic processes by generating, protecting, and managing keys used for encrypting and decrypting data and creating digital signatures and certificates.

### Rationale:

Using hardware security modules (HSM) to store and use certificates for the SSL/TLS communication provides greater security.

### Impact:

Hardware crypto cards/modules are external devices that can store certificates that can be used for the SSL/TLS communications. They store and protect the cryptographic keys throughout their lifecycles. The use of these devices have to be weighed against the security requirements and the additional expenses this would entail.

The specific configuration needed to support depends on the crypto card and JDK. The following applies when using the IBM JDK and when the keyStore type is PKCS11.

### Audit:

Ensure that the `location` attribute is set to the specific hardware crypto configuration file. In addition ensure that `type` attribute is set to `PKCS11`. Both these are under the `keyStore` element of the SSL configuration in the [Liberty configuration](#).

```
grep -w -R -i 'location' ${server.config.dir}
grep -w -R -i 'type' ${server.config.dir}
```

### Remediation:

Configure the `location` attribute to the specific hardware crypto card configuration file. In addition configure the `type` attribute to `PKCS11`. Both these attributes are in the `keyStore` element used by the SSL configuration in `${server.config.dir}/configDropins/overrides/*.xml`. In this example, the `HWCrypto.cfg` contains the hardware crypto configuration information.

```
<keyStore id="hwKeyStore"
  location="${server.config.dir}/HWCrypto.cfg"
  type="PKCS11"
  fileBased="false"
  provider="IBMPKCS11Impl"
  ...
/>
```

## References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-keystores>

## 4.2.14 Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms. (Manual)

### Profile Applicability:

- Level 2

### Description:

The National Institute of Standards and Technology (NIST) SP800-131a recommendation enforces stronger cryptographic keys and more robust algorithms.

### Rationale:

The NIST recommendation (SP 800-131A) provides more specific guidance to the use of stronger cryptographic keys and more robust algorithms for SSL/TLS connections and is recommended to be used if it is appropriate for your environment.

### Audit:

The SP800-131a recommendation can be enforced by setting the system property `com.ibm.jsse2.sp800-131` to `strict` when using IBM JDK. For other JDKs, follow their specific instructions.

Set `com.ibm.jsse2.sp800-131=strict` in the `jvm.options` file. More information on how to set the system properties provided in the reference section.

```
grep -w -R -i 'com.ibm.jsse2.sp800-131=strict' ${server.config.dir}
```

### Remediation:

Configure the system property `-Dcom.ibm.jsse2.sp800-131` to `strict` in the `jvm.options` file. The link in reference section provides more information on how to customize the Liberty environment with system properties.

In the `jvm.options` file add the following

```
-Dcom.ibm.jsse2.sp800-131=strict  
...
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-setting-up-run-in-sp800-131a>
2. <https://www.ibm.com/docs/en/was-liberty/base?topic=manually-customizing-liberty-environment>
3. <https://www.ibm.com/support/pages/setting-generic-jvm-arguments-websphere-application-server-v85-liberty-profile>

## 4.2.15 Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules (Manual)

### Profile Applicability:

- Level 2

### Description:

The National Institute of Standards and Technology (NIST) issues the FIPS 140 Publication Series to coordinate the requirements and standards for cryptography modules that include both hardware and software components. Protection of a cryptographic module within a security system is necessary to maintain the confidentiality and integrity of the information protected by the module. This standard specifies the security requirements that will be satisfied by a cryptographic module.

### Rationale:

Federal Information Processing Standards (FIPS) are standards and guidelines that are issued by the National Institute of Standards and Technology (NIST) for federal government computer systems. You can set up Liberty to comply with FIPS requirements by setting system properties and specifying a FIPS-validated security provider.

### Audit:

The FIPS standard can be enforced by setting the system properties –

`Dcom.ibm.jsse2.usefipsprovider=true` and –

`Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS` when using IBM JDK. For other JDKs, follow their specific instructions.

```
grep -w -R -i '-Dcom.ibm.jsse2.usefipsprovider=true' ${server.config.dir}
grep -w -R -i '-Dcom.ibm.jsse2.usefipsProviderName=IBMJCEPlusFIPS'
${server.config.dir}
```

In addition, configure to use the `IBMJCEPlusFIPS` security provider in the `$JAVA_HOME/jre/lib/security/java.security` file.

```
grep -w -R -i 'com.ibm.crypto.plus.provider.IBMJCEPlusFIPS' ${JAVA_HOME}
```

### Remediation:

Configure the system property `-Dcom.ibm.jsse2.usefipsprovider` to `true` and – `Dcom.ibm.jsse2.usefipsProviderName` to `IBMJCEPlusFIPS` in the `jvm.options` file. The link in reference section provides more information on how to customize the Liberty environment with system properties.

```
-Dcom.ibm.jsse2.usefipsprovider=true  
-Dcom.ibm.jsse2.usefipsProviderName=`IBMJCEPlusFIPS`  
...
```

In addition, configure the `IBMJCEPlusFIPS` provider in the `$JAVA_HOME/jre/lib/security/java.security` file.

```
security.provider.1=com.ibm.crypto.plus.provider.IBMJCEPlusFIPS  
...
```

## References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-setting-up-fips-compliance>
2. <https://www.ibm.com/support/pages/setting-generic-jvm-arguments-websphere-application-server-v85-liberty-profile>

## 4.3 Single Sign On (SSO)

Security recommendations for SSO technologies like OpenId connect (OIDC), SAML, OAuth



### 4.3.1 Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The signatureAlgorithm is used to sign the JWT and can be configured to use different algorithms like HS256 (symmetric), RS256 (asymmetric).

#### Rationale:

The OpenID Connect server provider should use an asymmetric algorithm to sign JSON Web Tokens (JWT) for better security since different keys are used for signing and verifying the token.

#### Audit:

Ensure the signatureAlgorithm attribute is set to a stronger algorithm like RS256 in the openidConnectProvider element for the OIDC provider and openidConnectClient for the OIDC client in the [Liberty configuration](#)

```
grep -w -R -i 'signatureAlgorithm' ${server.config.dir}
```

#### Remediation:

Add the signatureAlgorithm attribute to the openidConnectProvider element and set it to a strong algorithm, such as RS256 in \${server.config.dir}/configDropins/overrides/\*.xml.

```
<openidConnectProvider signatureAlgorithm="RS256" />
<openidConnectClient signatureAlgorithm="RS256" />
```

#### Default Value:

The WebSphere Liberty OpenID Connect Server Provider signs the token ID by default with the signature algorithm of HS256.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectProvider.html>

### 4.3.2 *Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) (Manual)*

#### **Profile Applicability:**

- Level 2

#### **Description:**

SPNEGO authentication supports both constrained and unconstrained delegation. Constrained delegation provides the ability to specify and enforce application trust boundaries on a user's behalf by limiting the flow of the SPNEGO tokens only to trusted servers.

#### **Rationale:**

Constrained delegation gives service administrators the ability to specify and enforce application trust boundaries by limiting the scope where application services can act on behalf of a user.

#### **Impact:**

Enabling constrained delegation requires additional configuration at the external KDC (Kerberos Distribution Center) server that is creating the SPNEGO token. The additional configuration depends on the KDC.

#### **Audit:**

If SPNEGO authentication is configured in the Liberty server make sure that the constrained delegation feature is also enabled in the [Liberty configuration](#).

```
grep -w -R -i 'spnego' ${server.config.dir}
grep -w -R -i 'constrainedDelegation' ${server.config.dir}
```

#### **Remediation:**

In `${server.config.dir}/configDropins/overrides/<any file name>.xml`, add the `constrainedDelegation-1.0` feature to the `featureManager` element:

```
<feature>constrainedDelegation-1.0</feature>
```

#### **References:**

1. <https://openliberty.io/docs/latest/reference/feature/constrainedDelegation-1.0.html>
2. [https://www.ibm.com/docs/en/was-liberty/nd?topic=SSAW57\\_liberty/com.ibm.websphere.wlp.doc/ae/twlp\\_config\\_kerb\\_constrained\\_del.html](https://www.ibm.com/docs/en/was-liberty/nd?topic=SSAW57_liberty/com.ibm.websphere.wlp.doc/ae/twlp_config_kerb_constrained_del.html)

### 4.3.3 Ensure 'tokenReuse' is set to 'false' (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The "jti" claim in the JWT token can be made unique to prevent the JWT tokens from being replayed.

#### Rationale:

The OpenID Connect relying party should not reuse JSON Web Tokens to prevent a token replay attack.

#### Audit:

Ensure the existence of the `tokenReuse` attribute and that it is set to `false` in the `openidConnectClient` element in the [Liberty configuration](#):

```
grep -w -R -i 'tokenReuse' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `tokenReuse` attribute is set to `false` by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectClient.html>

### 4.3.4 Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The issuer claim in the JWT token is used by the RP (relying party) to verify the OP (OIDC provider) token issuer.

#### Rationale:

The issuer claim in a JSON Web Token (JWT) should be required and validated by the OpenID Connect relying party. This helps to ensure the authenticity of the JWT by matching the issuer claim to the name attribute or the redirect attribute of the client configuration in the OpenID Connect server provider.

#### Audit:

Ensure the existence of the `disableIssChecking` attribute and that it is set to `false` in the `disableIssChecking` element in the [Liberty configuration](#):

```
grep -w -R -i 'disableIssChecking' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `disableIssChecking` attribute is set to `false` by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectClient.html>

### 4.3.5 Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

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.

#### Rationale:

Hostname verification verifies the request is talking to the correct server and has not been redirected to an unknown server thus mitigating man-in-the-middle security vulnerability attacks.

#### Audit:

Ensure the existence of the `hostNameVerificationEnabled` attribute set to true in the `openidConnectClient` element in the [Liberty configuration](#):

```
grep -w -R -i 'hostNameVerificationEnabled' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The WebSphere Liberty hostname verification for an OpenID Connect relying party is disabled by default, `hostNameVerificationEnabled` is set to `false`.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectClient.html>

### 4.3.6 Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The signatureAlgorithm is used by the RP to verify the signed ID tokens sent by the OP.

#### Rationale:

When the OpenID Connect Server Provider uses a signature algorithm to sign the ID tokens, sensitive information is protected and the relying party can verify the authenticity of the JSON Web Token (JWT).

#### Audit:

Ensure the existence of the signatureAlgorithm attribute and that it is set to a value other than none in the openidConnectClient element in the [Liberty configuration](#).

```
grep -w -R -i 'signatureAlgorithm' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The signatureAlgorithm attribute is set to HS256 by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectClient.html>

### 4.3.7 Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The signatureAlgorithm is used by the OP to sign the ID tokens.

#### Rationale:

The OpenID Connect Server Provider should use a signature algorithm to sign the ID tokens. This protects sensitive information and helps the relying party verify the authenticity of the JWT.

#### Audit:

Ensure the existence of the signatureAlgorithm attribute is set to a value other than none in the openidConnectProvider element in the [Liberty configuration](#)

```
grep -w -R -i 'signatureAlgorithm' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The signatureAlgorithm attribute is set to HS256 by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectProvider.html>

### 4.3.8 Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

HTTPS protocol protects the integrity and confidentiality of data between the client and the server.

#### Rationale:

Encrypting the communication between the OpenID Connect relying part and the OpenID Connect server provider using HTTPS protects sensitive information.

#### Audit:

Ensure the existence of the `httpsRequired` attribute and that it is set to `true` in the `openidConnectClient` element in the [Liberty configuration](#):

```
grep -w -R -i 'httpsRequired' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `httpsRequired` attribute is set to `true` by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectClient.html>



### 4.3.9 Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP) (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The tokenEndpoint is used to verify the authenticity of the RP (relying party).

#### Rationale:

When the OpenID Connect server provider requires a token endpoint authentication method the provider can verify the authenticity of the relying party.

#### Audit:

Ensure the existence of the `tokenEndpointAuthMethodsSupported` attribute and that it does not include a value of `none` in the `openidConnectProvider` element in the [Liberty configuration](#)

```
grep -w -R -i 'tokenEndpointAuthMethodsSupported' ${server.config.dir}
```

#### Remediation:

Add the `tokenEndpointAuthMethodsSupported` attribute to the `openidConnectProvider` element to `${server.config.dir}/configDropins/overrides/*.xml` and set it to a valid list of authentication methods, such as `client_secret_post`, to ensure that an authentication method is required for the token endpoint.

```
<openidConnectProvider tokenEndpointAuthMethodsSupported="client_secret_post,
client_secret_basic" />
```

#### Default Value:

The `tokenEndpointAuthMethodsSupported` attribute is set to `client_secret_post, client_secret_basic` by default.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/openidConnectProvider.html>

### 4.3.10 Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0 (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The OAuth access token contains sensitive data and should be hashed to protect it.

#### Rationale:

Encoding the OAuth access token using a hashing algorithm protects sensitive information.

#### Audit:

Ensure the existence of the `accessTokenEncoding` attribute and that it is set to a value other than `plain` in the `oauthProvider` element in the [Liberty configuration](#).

```
grep -w -R -i 'accessTokenEncoding' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `accessTokenEncoding` is set to `plain` for backward compatibility by default.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-oauthprovider>

### 4.3.11 Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0 (Automated)

#### Profile Applicability:

- Level 1

#### Description:

Public clients can be blocked to access OAuth applications for better control.

#### Rationale:

Blocking OAuth public clients prevents client IDs or client secrets from being exposed

#### Audit:

Ensure the existence of the `allowPublicClients` attribute and that it is set to `false` in the `oauthProvider` element in the [Liberty configuration](#):

```
grep -w -R -i 'allowPublicClients' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `allowPublicClients` attribute is set to `false` by default.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-oauthprovider>

### 4.3.12 Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0 (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The OAuth client secret is encoded using the 'clientSecretEncoding' attribute.

#### Rationale:

Encoding the OAuth client secret at highest level available reduces the possibility of it being decoded by an unauthorized party.

#### Audit:

Ensure the existence of the `clientSecretEncoding` attribute and that it is set to a value other than `xor` in the `oauthProvider` element in the [Liberty configuration](#).

```
grep -w -R -i 'clientSecretEncoding' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `clientSecretEncoding` attribute is set to `xor` by default.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-oauthprovider>

### 4.3.13 Ensure 'httpsRequired' is set to 'true' in OAuth 2.0 (Automated)

#### Profile Applicability:

- Level 1

#### Description:

HTTPS protocol protects the integrity and confidentiality of data between the client and the server.

#### Rationale:

Encrypting communication between the OAuth client and the provider and using HTTPS protects sensitive information.

#### Audit:

Ensure the existence of the `httpsRequired` attribute and that it is set to `true` in the `oauthProvider` element in the [Liberty configuration](#):

```
grep -w -R -i 'httpsRequired' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The `httpsRequired` attribute is set to `true` by default.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-oauthprovider>

### 4.3.14 Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0 (Automated)

#### Profile Applicability:

- Level 1

#### Description:

Resource owner validation check validates the resource owner credentials.

#### Rationale:

Verifying the credentials of the resource owner prevents unauthorized access.

#### Audit:

Ensure the existence of the `skipResourceOwnerValidation` attribute and that it is set to `false` in the `oauthProvider` element in the [Liberty configuration](#):

```
grep -w -R -i 'skipResourceOwnerValidation' ${server.config.dir}
```

#### Remediation:

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" />
```

#### Default Value:

The WebSphere Liberty OAuth resource owner validation is enabled by default, `skipResourceOwnerValidation` is set to `false`.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-oauthprovider>

### 4.3.15 Ensure 'httpsRequired' is set to 'true' in SAML (Automated)

#### Profile Applicability:

- Level 1

#### Description:

HTTPS protocol protects the integrity and confidentiality of data between the client and the server.

#### Rationale:

Transport communication accessing a SAML WebSSO service provider end point should be secured with HTTPS (TLS) to protect sensitive information.

#### Audit:

Ensure the `httpsRequired` attribute is set to `true` in on all `samlWebSso20` elements in the [Liberty configuration](#).

```
grep -w -R -i 'httpsRequired' ${server.config.dir}
```

#### Remediation:

Add the `httpsRequired` attribute to all `samlWebSso20` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set it to `true`.

```
<samlWebSso20 ...  
  httpsRequired="true"  
>
```

#### Default Value:

WebSphere Liberty sets the `httpsRequired` to `true` on `samlWebSso20` elements.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/samlWebSso20.html>

### 4.3.16 Enforce 'wantAssertionsSigned' to 'true' in SAML (Automated)

#### Profile Applicability:

- Level 1

#### Description:

A SAML (Security Assertions Markup Language) authentication assertion is issued as proof of an authentication event and can be signed.

#### Rationale:

WebSphere Liberty should require that SAML assertions (<saml:Assertion> elements) contain a signature element that signs the assertion so that it can be verified securely by the server.

#### Audit:

Ensure the `wantAssertionsSigned` attribute is set to `true` on all `samlWebSso20` elements in the [Liberty configuration](#).

```
grep -w -R -i 'wantAssertionsSigned' ${server.config.dir}
```

#### Remediation:

Add the `wantAssertionsSigned` attribute to all `samlWebSso20` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set it to `true`.

```
<samlWebSso20 ...  
  wantAssertionsSigned="true"  
>
```

#### Default Value:

WebSphere Liberty sets the `wantAssertionsSigned` to `true` on `samlWebSso20` elements.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/samlWebSso20.html>



### 4.3.17 Ensure 'authnRequestsSigned' is set to 'true' in SAML (Automated)

#### Profile Applicability:

- Level 1

#### Description:

SAML providers can sign the request messages for authenticity.

#### Rationale:

SAML authentication request messages (<samlp:AuthnRequest> messages) can be signed so it can be verified securely by the receiver.

#### Audit:

Ensure the `authnRequestsSigned` attribute is set to `true` on all `samlWebSso20` elements in the [Liberty configuration](#).

```
grep -w -R -i 'authnRequestsSigned' ${server.config.dir}
```

#### Remediation:

Add the `authnRequestsSigned` attribute to all `samlWebSso20` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set it to `true`.

```
<samlWebSso20 ...  
  authnRequestsSigned="true"  
>
```

#### Default Value:

WebSphere Liberty sets the `authnRequestsSigned` to `true` on `samlWebSso20` elements.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/samlWebSso20.html>

## 4.4 General

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

### 4.4.1 Ensure 'disableXPoweredBy' is set to 'true' (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The `disableXPoweredBy` setting can reveal the server's identity.

#### Rationale:

Preventing the Liberty server from advertising its presence in this manner will prevent malicious attackers from determining the server's identity and exploiting any security vulnerabilities.

#### Audit:

Ensure the `disableXPoweredBy` attribute for the `webContainer` element is set to `true` in the [Liberty configuration](#).

```
grep -w -R 'disableXPoweredBy' ${server.config.dir}
```

#### Remediation:

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" />
```




#### Default Value:




For servlet-5.0 and newer, the default value is `true`. For previous versions the value is `false`.

#### References:

1. <https://openliberty.io/docs/latest/reference/config/webContainer.html>

#### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 Securely Dispose of Data</b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<p><b><u>13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b></p> <p>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.</p>			

## 4.4.2 Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false' (Automated)

### Profile Applicability:

- Level 1

### Description:

Setting the preserveFullyQualifiedReferrerUrl attribute to false ensures that the host for the referrer URL is removed, and that the redirect is to localhost.

### Rationale:

Using a fully qualified referrer URL containing the hostname may open your systems to potential URL redirect attacks.

### Audit:

Ensure the preserveFullyQualifiedReferrerUrl attribute is set to false in the webAppSecurity element in the [Liberty configuration](#)

```
grep -w -R -i 'preserveFullyQualifiedReferrerUrl' ${server.config.dir}
```

### Remediation:

Set the preserveFullyQualifiedReferrerUrl attribute to false in the webAppSecurity element on \${server.config.dir}/configDropins/overrides/\*.xml

```
<webAppSecurity preserveFullyQualifiedReferrerUrl="false" />
```

### Default Value:

The preserveFullyQualifiedReferrerUrl attribute is false by default.

### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

### 4.4.3 Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects (Automated)

#### Profile Applicability:

- Level 1

#### Description:

For logout page redirects, explicit domain names can be listed.

#### Rationale:

Defining approved domain names for redirects prevents the server from redirecting to a disallowed domain.

#### Audit:

Ensure the `logoutPageRedirectDomainNames` attribute in the `webAppSecurity` element lists the domain names that are allowed for the logout page redirect in the [Liberty configuration](#)

```
grep -w -R -i 'logoutPageRedirectDomainNames' ${server.config.dir}
```

#### Remediation:

Set the `logoutPageRedirectDomainNames` attribute in the `webAppSecurity` element to a pipe(`|`) separated list of domain names that are allowed for the logout page redirect in `${server.config.dir}/configDropins/overrides/*.xml`

For Example, for the two domains `ibm.com` and `openliberty.io`

```
<webAppSecurity logoutPageRedirectDomainNames="ibm.com|openliberty.io" />
```

#### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

#### 4.4.4 Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic (Manual)

##### Profile Applicability:

- Level 1

##### Description:

Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

##### Rationale:

Defining an IP address exclude list protects against unwanted inbound connections.

##### Audit:

Ensure the `hostNameExcludeList` attribute is set to a list of host names in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'hostNameExcludeList' ${server.config.dir}
```

##### Remediation:

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" />
```

##### References:

1. <https://openliberty.io/docs/latest/reference/config/httpEndpoint.html>

#### 4.4.5 Ensure 'logoutOnHttpSessionExpire' is set to 'true' (Automated)

##### Profile Applicability:

- Level 1

##### Description:

Logout users after the HTTP session timer expires.

##### Rationale:

Logging out users after the HTTP session expires syncs the session and the LTPA authentication token and prompts the user to login again when accessing the resource.

##### Audit:

Ensure the `logoutOnHttpSessionExpire` attribute is set to `true` in the `webAppSecurity` element in the [Liberty configuration](#)

```
grep -w -R -i 'logoutOnHttpSessionExpire' ${server.config.dir}
```

##### Remediation:

Set the `logoutOnHttpSessionExpire` attribute to `true` in the `webAppSecurity` element on `${server.config.dir}/configDropins/overrides/*.xml`

```
<webAppSecurity logoutOnHttpSessionExpire="true" />
```

##### Default Value:

The `logoutOnHttpSessionExpire` attribute is `false` by default.

##### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>



## 4.4.6 Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic (Manual)

### Profile Applicability:

- Level 1

### Description:

Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address include list allows only wanted inbound connections.

### Audit:

Ensure the `hostNameIncludeList` attribute is set to a list of host names in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'hostNameIncludeList' ${server.config.dir}
```

### Remediation:

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" />
```

### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

#### 4.4.7 Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic (Automated)

##### Profile Applicability:

- Level 1

##### Description:

IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

##### Rationale:

Defining an IP address include list allows only wanted inbound connections.

##### Audit:

Ensure the `addressIncludeList` attribute is set to a list of IP addresses in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'addressIncludeList' ${server.config.dir}
```

##### Remediation:

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" />
```

##### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

## 4.4.8 Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic (Manual)

### Profile Applicability:

- Level 1

### Description:

IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address exclude list protects against unwanted inbound connections.

### Audit:

Ensure the `addressExcludeList` attribute is set to a list of hostnames in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'addressExcludeList' ${server.config.dir}
```

### Remediation:

Add the `addressExcludeList` attribute to all `tcpOptions` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set it to a comma-separated list of IP addresses.

```
<tcpOptions addressExcludeList="254.1.0.9,*.1.255.0" />
```

### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

## 4.4.9 Ensure "trustedSensitiveHeaderOrigin" is set to trusted host names and IP addresses for sensitive data (Automated)

### Profile Applicability:

- Level 1

### Description:

The web server plug-in uses private headers to provide information about the original request. A subset of these headers is considered sensitive. Incoming sensitive private headers are not trusted from any source. To allow sensitive private header processing for specific trusted sources, specify a comma-separated list of IP addresses and hostnames.

### Rationale:

### Audit:

Ensure the `trustedSensitiveHeaderOrigin` attribute for first `httpDispatcher` element is set to appropriate value in the [Liberty configuration](#).

```
grep -w -R 'trustedSensitiveHeaderOrigin' ${server.config.dir}
```

### Remediation:

Add the settings below to `${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<httpDispatcher ...  
    trustedSensitiveHeaderOrigin="localhost, 127.0.0.1, 192.168.*.*,  
0:0:0:0:0:ffff:*:*, *.ibm.com"/>
```

### Default Value:

The default value is `none`.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-httpdispatcher>
2. <https://openliberty.io/blog/2021/02/19/configure-trustedHeaderOrigin-21002.html>

## 4.4.10 Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

### Audit:

Ensure the `trustedHeaderOrigin` attribute for first `httpDispatcher` element is set to appropriate value in the [Liberty configuration](#).

```
grep -w -R 'trustedHeaderOrigin' ${server.config.dir}
```

### Remediation:

Add the settings below to `${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<httpDispatcher ...  
    trustedHeaderOrigin="localhost, 127.0.0.1, 192.168.*.*,  
0:0:0:0:0:0:0:0:ffff:*:*, *.ibm.com"/>
```

### Default Value:

The default value is `*`.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=configuration-httpdispatcher>
2. <https://openliberty.io/blog/2021/02/19/configure-trustedHeaderOrigin-21002.html>

#### 4.4.11 Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout (Automated)

##### Profile Applicability:

- Level 1

##### Description:

Once a user is logged out, the logout page redirects can be controlled to be redirected to a specific set of trusted domains instead of just localhost.

##### Rationale:

One can control the domain names to be directed to once a logout happens. This will ensure that the redirection is not happening to an untrusted server.

##### Audit:

Ensure that pipe (|) separated list of domain names are provided in `logoutPageRedirectDomainNames` attribute of `webAppSecurity` element in [Liberty configuration](#).

```
grep -w -R -i 'logoutPageRedirectDomainNames' ${server.config.dir}
```

##### Remediation:

Add/set the setting below to `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<webAppSecurity logoutPageRedirectDomainNames="<domain name list>" />
```

##### References:

1. <https://openliberty.io/docs/latest/reference/config/webAppSecurity.html>

## 4.4.12 Ensure security constraints are specified to protect web applications (Automated)

### Profile Applicability:

- Level 1

### Description:

Protect web applications by configuring security constraints for all web resources using either deployment descriptor and/or annotations.

### Rationale:

Specifying security constraints allows fine grained access control to protected resources. This can be done either using deployment descriptor and/or annotations.

### Audit:

Ensure the `<security-constraint>` elements are specified in the `web.xml` file for each application.

```
grep -w -R -i 'security-constraint' ${server.config.dir}
```

Note: If using annotations, make sure that the appropriate methods are protected using the `@ServletSecurity` annotation.

### Remediation:

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...
}
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=applications-getting-started-security-in-liberty>



## 4.4.13 Ensure application security feature is enabled (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Enabling Security for applications prevents unauthorized users from accessing an application. This is the basic security feature to be added in addition to other features that the server might need.

### Audit:

Ensure that any one of the appSecurity-\* feature is enabled in the [Liberty configuration](#).

Note that a feature can be individually specified or can be included in other features. For more information see [Liberty Features Overview](#).

```
grep -w -R 'appSecurity' ${server.config.dir}
```

### Remediation:

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>
```

### References:

1. <https://openliberty.io/docs/latest/reference/feature/appSecurity-3.0.html>

#### 4.4.14 Ensure

*'invalidateOnUnauthorizedSessionRequestException' is set to 'false' (Automated)*

##### Profile Applicability:

- Level 1

##### Description:

When a user tries to access a session owned by another user, the `UnauthorizedSessionRequestException` is raised so the user cannot continue.

##### Rationale:

If this is set to true, there is no exception raised. The only reason to enable this is to suppress the exception when the session is accessed after a user's token is expired.

##### Audit:

Ensure that `invalidateOnUnauthorizedSessionRequestException` attribute of `httpSession` is set to true in the [Liberty configuration](#).

```
grep -w -R 'invalidateOnUnauthorizedSessionRequestException'
${server.config.dir}
```

##### Remediation:

Set the `invalidateOnUnauthorizedSessionRequestException` attribute to true in the `httpSession` element in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<httpSession invalidateOnUnauthorizedSessionRequestException="true" />
```

##### Default Value:

`invalidateOnUnauthorizedSessionRequestException` is false by default.

##### References:

1. <https://openliberty.io/docs/latest/reference/config/httpSession.html>

#### *4.4.15 Ensure Web Server Document Root does not contain information that should be private (Automated)*

##### **Profile Applicability:**

- Level 1

##### **Description:**

WAR files contain servable content. The Web container will serve any files found in the root(Web Server Document Root) of the WAR file. This is fine as long as you place only servable content in the root. Thus, you should never place content that shouldn't be shown to users in the root of the WAR. For example, don't put property files, class files, or other important information there. If you must place such information in the WAR file, place it within the WEB-INF directory, as permitted by the servlet specification. Information there is never served by the Web.

##### **Rationale:**

##### **Audit:**

Ensure that there is no content that shouldn't be shown to users in the root of the WAR. For example, check for property files, class files, or other important information.

##### **Remediation:**

Remove all content that shouldn't be shown to users in the root of the WAR. For example, relocate property files, class files, or other important information within the WEB-INF directory, as permitted by the servlet specification.

## 4.4.16 Ensure HTTP session overflow is 'disabled' (Manual)

### Profile Applicability:

- Level 1

### Description:

Applications that use in-memory HTTP sessions can configure if overflow is allowed and the number of sessions that can be created.

### Rationale:

Disabling session overflow and the number of maximum number of sessions allowed helps to avoid denial-of-service attacks in which attackers generate new sessions until all JVM memory is exhausted.

### Audit:

Ensure that the `allowOverflow` attribute of `httpSession` element is set to `false` and the `maxInMemorySessionCount` value of the `httpSession`` element is set to an appropriate value in [Liberty configuration](#).

```
grep -w -R -i 'allowOverflow' ${server.config.dir}
grep -w -R -i 'maxInMemorySessionCount' ${server.config.dir}
```

### Remediation:

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" />
```

### Default Value:

WebSphere Liberty allows session overflow by default, `allowOverflow` is set to `true`. The maximum session count, `maxInMemorySessionCount`, is 1000 by default.

### References:

1. <https://openliberty.io/docs/latest/reference/config/httpSession.html>

## 4.4.17 Ensure uncovered http methods are denied (Automated)

### Profile Applicability:

- Level 1

### Description:

Servlets are secured by URL and each URL that is to be secured must be specified in the web.xml file describing the application.

### Rationale:

A servlet can have multiple aliases and an application can have many servlets, making it easy to accidentally forget to secure an alias or URL for a servlet. If just one servlet URL is insecure, an intruder might be able to bypass security. Use wildcards to secure servlets wherever possible instead of specific URLs and configure the application to deny access to uncovered http methods.

### Impact:

Open Liberty secures URLs and not the underlying classes, if just one servlet URL is insecure, an intruder might be able to bypass security.

### Audit:

Review all of your web applications to ensure that each servlet has an alias associated with security constraints.

Review each application's web.xml file located in the application's

`${wlp.user.dir}/shared/apps/<app_name>/WEB-INF/` directory and determine if a `deny-uncovered-http-methods` element is set. Applications may also be in the `${server.config.dir}/dropins` directory or a custom location.

```
grep -w -R -i 'deny-uncovered-http-methods'
${wlp.user.dir}/shared/apps/<app_name>/WEB-INF/
```

**Note:** The security-constraints can also be set in the code using annotations like `@ServletSecurity`.

### Remediation:

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 `WEB-INF/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`

## References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=applications-getting-started-security-in-liberty>
2. <https://openliberty.io/docs/latest/application-configuration-hardening.html>

## 4.4.18 Ensure 'disallowServeServletsByClassName' is 'disabled' (Automated)

### Profile Applicability:

- Level 1

### Description:

Servlets can be served by class name or via a normal URL alias.

### Rationale:

Serving servlets by class name allows anyone that knows the class name of any servlet to invoke it directly which leads to a security exposure from possible misuse of the servlet.

### Audit:

Ensure the `disallowServeServletsByClassName` attribute on the `webContainer` element in the [Liberty configuration](#) is set to `true`.

```
grep -w -R -i 'disallowServeServletsByClassName' ${server.config.dir}
```

Serving servlets by class name can also be set on the `enable-serving-servlets-by-class-name` attributes in the `web-ext` element for individual `webApplication` elements. For legacy web applications, servlet serving by class name can also be set in the `ibm-web-ext.xml` or `ibm-web-ext.xmi` files in the application.

### Remediation:

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.

### Default Value:

WebSphere Liberty disables serving servlets by class name by default, `disallowServeServletsByClassName` is set to `true`.

### References:

1. <https://openliberty.io/docs/latest/reference/config/webApplication.html#web-ext>
2. <https://openliberty.io/docs/latest/reference/config/webContainer.html>
3. <https://openliberty.io/docs/latest/application-configuration-hardening.html>

## 4.4.19 Ensure server headers on requests are removed (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Revealing the specific software version of the server may allow Websphere Liberty to become more vulnerable to attacks against software that is known to contain security holes. Enabling the `removeServerHeader` on the [Liberty configuration](#) removes this information and overrides the default value that is sent down in the HTTP header further masking any information about Websphere Liberty implementation.

### Audit:

Ensure the `removeServerHeader` attribute set to true in the `httpOptions` element in [Liberty configuration](#).

```
grep -w -R -i 'removeServerHeader' ${server.config.dir}
```

### Remediation:

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>
```

### Default Value:







Liberty server headers are enabled by default.

### References:

1. <https://openliberty.io/docs/latest/reference/config/httpOptions.html>



**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 <u>Securely Dispose of Data</u></b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	<b>13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b> 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.			

## 4.4.20 Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Directory browsing enables the web applications to expose the file names in the application.

### Audit:

Ensure the `directoryBrowsingEnabled` attribute on the `webContainer` element configured in the [Liberty configuration](#) is set to `false`.

```
grep -w -R -i 'directoryBrowsingEnabled' ${server.config.dir}
```

Directory browsing can also be set on the `enable-directory-browsing` attributes in the `web-ext` element for individual `webApplication` elements.

For legacy web applications, directory browsing can also be set in the `ibm-web-ext.xml` or `ibm-web-ext.xmi` files in the application.

### Remediation:

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.







### Default Value:

In WebSphere Liberty directory browsing for web applications is disabled by default, `directoryBrowsingEnabled` is set to `false`.

### References:

1. <https://openliberty.io/docs/latest/reference/config/webContainer.html>
2. <https://openliberty.io/docs/latest/reference/config/webApplication.html#web-ext>
3. <https://openliberty.io/docs/latest/application-configuration-hardening.html>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 <u>Securely Dispose of Data</u></b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	<b>13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b> 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.			

## 4.4.21 Ensure 'default-error-page' is set for web applications (Manual)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Without a default error page, displaying the exception stack may reveal information about the application. For example, the names of classes and methods that are in the stack information. The exception message text is also displayed which could contain sensitive information.

### Audit:

Ensure that the `default-error-page` attribute is set to an error page in the `web-ext` element of all of the `webApplication` elements in the [Liberty configuration](#).

```
grep -w -R -i 'default-error-page' ${server.config.dir}
```

The default error page can also be set in the `web.xml` file in the application. For legacy web applications, the default error page can also be set in the `ibm-web-ext.xml` or `ibm-web-ext.xmi` files in the application.

### Remediation:

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"/>
```







### Default Value:

WebSphere Liberty does not provide a default error page for web applications.

### References:

1. <https://openliberty.io/docs/latest/reference/config/webApplication.html#web-ext>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 <u>Securely Dispose of Data</u></b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	<b>13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b> 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.			

## 4.4.22 Ensure virtual hosts are defined to isolate applications (Automated)

### Profile Applicability:

- Level 1

### Description:

Isolate applications by configuring separate virtual hosts.

### Rationale:

Use separate ports using virtual hosts to decouple access to different applications.

### Audit:

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](#).

```
grep -w -R -i 'virtual-host name' ${server.config.dir}
```

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 [Liberty configuration](#).

```
grep -w -R -i 'virtualHost' ${server.config.dir}
```

### Remediation:

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>
```

### Default Value:

WebSphere Liberty provides a default virtual host (default\_host) that matches requests from any incoming host and port combination.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-using-virtual-hosts>

### 4.4.23 Ensure virtual hosts are Defined to isolate JMX communication and application traffic (Automated)

#### Profile Applicability:

- Level 1

#### Description:

Isolate JMX communication and application traffic using separate virtual hosts for applications.

#### Rationale:

Use separate ports using virtual hosts to decouple JMX and applications communication.

#### Audit:

Ensure the following to isolate the JMX communication with application traffic. This needs to be configured for every application so that only the JMX communication uses the default virtual host and all applications traffic use the respective virtual hosts.

1. Configure the application with `virtual-host name` in the `enterpriseApplication` or `webApplication` elements in the [Liberty configuration](#).

```
grep -w -R -i 'virtual-host name' ${server.config.dir}
```

Note: The `virtual-host name` can also be set in the `ibm-web.bnd.xml` or `ibm-web-bnd.xmi` file in the application.

2. Configure the application to use the specific virtual host in [Liberty configuration](#).

```
grep -w -R -i 'virtualHost' ${server.config.dir}
```

#### Remediation:

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>
```

### Default Value:

WebSphere Liberty provides a default virtual host (default\_host) that matches requests from any incoming host and port combination including both the JMX and the application traffic.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-using-virtual-hosts>

## 4.4.24 Ensure whitelisting of virtual hosts to validate access based on originating endpoint (Automated)

### Profile Applicability:

- Level 1

### Description:

Restrict access to default and system applications based on originating endpoint.

### Rationale:

Restrict access to default and system application based on an originating endpoint.

### Audit:

1. Ensure that a specific `httpEndpoint` is defined in the [Liberty configuration](#).

```
grep -w -R -i 'httpEndpoint' ${server.config.dir}
```

2. Ensure that the `allowFromEndPoint` attribute is set in the `virtualHost` element to the `httpEndpoint` specified above in [Liberty configuration](#).

```
grep -w -R -i 'virtualHost' ${server.config.dir}
```

### Remediation:

1. Configure a `httpEndpoint` element in the [Liberty configuration]`${server.config.dir}/configDropins/overrides/*.xml`.

For example,

```
<httpEndpoint id="localhostOnly" host="localhost" httpPort="9081"
httpsPort="9444"/>
```

2. Configure the `default_host` `virtualHost` element with `allowFromEndPointRef` pointing to the `httpEndpoint` value configured above.

```
<virtualHost id="default_host" allowFromEndpointRef="localhostOnly">
  <hostAlias>*:9081</hostAlias>
  <hostAlias>*:9444</hostAlias>
</virtualHost>
```

### Default Value:

WebSphere Liberty provides a default virtual host (`default_host`) that matches requests from any incoming host and port combination.

**References:**

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-using-virtual-hosts>

## 5 Enterprise Java Beans (EJB) Applications

## **5.1 The CSlv2 (Common Secure Interoperability version 2) serverPolicy**

CSlv2 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 CSlv2 policy on the server side. One can configure one or more of these to protect the EJB applications as per the CSlv2 specification.

### 5.1.1 Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The CSiv2 Transport policy configures security at the transport layer when accessing EJB applications using RMI/IIOP.

#### Rationale:

Setting the configuration at the transport layer for RMI/IIOP requests will ensure that the data is passed through the IIOPS secure channel.

#### Audit:

Ensure the `sslEnabled` attribute in the `ORB > serverPolicy.csiv2 > layers > transportLayer` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'sslEnabled' ${server.config.dir}
```

#### Remediation:

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>
```





#### Default Value:

The default value of the `sslEnabled` attribute on `csiv2` in the `orb` element is `true`.

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-inbound-csiv2-transport-layer>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 5.1.2 Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer (Automated)

### Profile Applicability:

- Level 1

### Description:

The CSiv2 Authentication policy configures security at the Authentication layer when accessing EJB applications using RMI/IIOP.

### Rationale:

The `establishTrustInClient` option of the authentication layer is set to `Required` to force the client to use one of the authentication mechanisms specified.

### Audit:

If the clients are required to use authentication layer to provide credentials, ensure the `establishTrustInClient` attribute in the `ORB > serverPolicy.csiv2 > layers > authenticationLayer` element is set to `Required` in the [Liberty configuration](#)

```
grep -w -R -i 'establishTrustInClient' ${server.config.dir}
```

### Remediation:

Set the `establishTrustInClient` attribute in `ORB > serverPolicy.csiv2 > layers > authenticationLayer` to `Required` in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<orb id="defaultOrb">
  <serverPolicy.csiv2>
    <layers>
      <authenticationLayer mechanisms="LTPA,GSSUP"
establishTrustInClient="Required"/>
    </layers>
  </serverPolicy.csiv2>
</orb>
```

### Default Value:



The default value of the `establishTrustInClient` attribute on `csiv2` in the `orb` element is `Required`.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-inbound-csiv2-authentication-layer>



## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><b>16.11 <u>Leverage Vetted Modules or Services for Application Security Components</u></b></p> <p>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.</p>			

### 5.1.3 Ensure 'identityAssertionEnabled' is set to 'true' within the CSiv2 Attribute Layer (Automated)

#### Profile Applicability:

- Level 1

#### Description:

The CSiv2 Attribute policy configures security at the Attribute layer when accessing EJB applications using RMI/IIOP.

#### Rationale:

If identity assertion is required by the server, the `identityAssertionTypes` attribute of the attribute layer is set to specify the identity token types that the server supports. If other authentication mechanisms are used to validate the client, this is not required.

#### Audit:

Ensure the `identityAssertionEnabled` attribute in the `ORB > serverPolicy.csiv2 > layers > attributeLayer` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'identityAssertionEnabled' ${server.config.dir}
```

And also ensure `identityAssertionTypes` is specified and does not include `ITTAnonymous`

```
grep -w -R -i 'identityAssertionTypes' ${server.config.dir}
```

#### Remediation:

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>
```

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-inbound-csiv2-attribute-layer>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 <u>Perform Application Layer Filtering</u></b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 <u>Implement Application Firewalls</u></b> 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.			●

## **5.2 The CSlv2 (Common Secure Interoperability version 2) Client Policy**

CSlv2 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 CSlv2 policy on the client side. One can configure one or more of these to protect the EJB applications as per the CSlv2 specification.

## 5.2.1 Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech (Manual)

### Profile Applicability:

- Level 1

### Description:

The CSiv2 Transport policy configures security at the transport layer when accessing EJB applications using RMI/IIOP.

### Rationale:

Setting the configuration at the transport layer for RMI/IIOP requests will ensure that the data is passed through the IIOPS secure channel.

### Audit:

Ensure the `sslEnabled` attribute in the `ORB > clientPolicy.csiv2 > layers > transportLayer` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'sslEnabled' ${server.config.dir}
```

### Remediation:

Set the `sslEnabled` attribute in `ORB > clientPolicy.csiv2 > layers > transportLayer` to `true` in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<orb id="defaultOrb">
  <clientPolicy.csiv2>
    <layers>
      <transportLayer sslEnabled="true"/>
    </layers>
  </clientPolicy.csiv2>
</orb>
```





### Default Value:

The default value of the `sslEnabled` attribute on `csiv2` in the `orb` element is `true`.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-outbound-csiv2-transport-layer>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 5.2.2 Ensure 'establishTrustInClient' is 'Required' for the CSiv2 Authentication Layer - needsReview/Zech (Manual)

### Profile Applicability:

- Level 1

### Description:

The CSiv2 Authentication policy configures security at the Authentication layer when accessing EJB applications using RMI/IIOP.

### Rationale:

The `establishTrustInClient` option of the authentication layer is set to `Required` to force the client to use one of the authentication mechanisms specified.

### Audit:

Ensure the `establishTrustInClient` attribute in the `ORB > serverPolicy.csiv2 > layers > authenticationLayer` element is set to `Required` in the [Liberty configuration](#)

```
grep -w -R -i 'establishTrustInClient' ${server.config.dir}
```

### Remediation:



Set the `establishTrustInClient` attribute in `ORB > clientPolicy.csiv2 > layers > authenticationLayer` to `Required` in the `${server.config.dir}/configDropins/overrides/<any file name>.xml`

```
<orb id="defaultOrb">
  <clientPolicy.csiv2>
    <layers>
      <authenticationLayer mechanisms="LTPA,GSSUP"
establishTrustInClient="Required"/>
    </layers>
  </clientPolicy.csiv2>
</orb>
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-outbound-csiv2-authentication-layer>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><b><u>16.11 Leverage Vetted Modules or Services for Application Security Components</u></b></p> <p>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.</p>			



### 5.2.3 Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSiv2 Attribute Layer - review/Zech (Manual)

#### Profile Applicability:

- Level 1

#### Description:

The CSiv2 Attribute policy configures security at the Attribute layer when accessing EJB applications using RMI/IIOP.

#### Rationale:

The `identityAssertionTypes` attribute of the attribute layer is set to specify the identity token types that the server supports.

#### Audit:

Ensure the `identityAssertionEnabled` attribute in the `ORB > serverPolicy.csiv2 > layers > attributeLayer` element is set to `true` in the [Liberty configuration](#)

```
grep -w -R -i 'identityAssertionEnabled' ${server.config.dir}
```

And also ensure `identityAssertionTypes` is specified and does not include `ITTAnonymous`

```
grep -w -R -i 'identityAssertionTypes' ${server.config.dir}
```

#### Remediation:

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">
  <clientPolicy.csiv2>
    <layers>
      <attributeLayer identityAssertionEnabled="true"
identityAssertionTypes="ITTX509CertChain, ITTDistinguishedName"/>
    </layers>
  </clientPolicy.csiv2>
</orb>
```

#### References:

1. <https://www.ibm.com/docs/en/was-liberty/core?topic=liberty-configuring-outbound-csiv2-attribute-layer>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 <u>Perform Application Layer Filtering</u></b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 <u>Implement Application Firewalls</u></b> 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.			●

## 5.3 Java Serialization

### 5.3.1 Ensure filters are configured for Java serialization (JEP 290) (Manual)

#### Profile Applicability:

- Level 1

#### Description:

Java provides configuration to allow incoming streams of object-serialization data to be filtered in order to improve both security and robustness. This allows

#### Rationale:

There have been issues with deserialization of untrusted data in Java. To mitigate this, openJDK provides a flexible mechanism using filters to restrict the classes that can be deserialized. In addition, it also provides metrics to control the filter size, depth etc.

#### Impact:

The filters to allow classes to be deserialized or deny classes to be serialized should be based on individual scenarios. They need to be done with care to mitigate any side impacts. More information in the links provided in the reference section.

#### Audit:

Check for the System property `jdk.serialFilter` in the `${server.config.dir}/bootstrap.properties` file or the security property in `$JAVA_HOME/conf/security/java.properties` and confirm it has the correct filters set to restrict Java deserialization to the trusted classes.

```
jdk.serialFilter
```

#### Remediation:







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
```

#### References:

1. <https://openjdk.java.net/jeps/290>
2. <https://openjdk.java.net/jeps/415>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>4.4 <u>Implement and Manage a Firewall on Servers</u></b> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v7	<b>9.4 <u>Apply Host-based Firewalls or Port Filtering</u></b> 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.			

## 5.4 EJB Authentication

## 5.4.1 Ensure that all appropriate EJB methods are protected (Automated)

### Profile Applicability:

- Level 1

### Description:

Access to the EJB methods should be protected appropriately using roles in deployment descriptor or annotations.

### Rationale:

Protecting the appropriate EJB methods will ensure that users have to provide the correct credentials to access them.

### Audit:

Ensure the `<method-permission>` elements are specified in the `ejb-jar.xml` deployment descriptor file for each application.

```
grep -w -R -i 'method-permission' ${server.config.dir}
```

Note: If using annotations, make sure that the appropriate methods are protected using one of the `@RolesAllowed`/`@PermitAll`/`@DenyAll` annotations.

### Remediation:

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 () {...}
    ...
}
```

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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.			



## 6 Web Services

## 6.1 Ensure 'HttpsToken' is set in WS-Security policy (Automated)

### Profile Applicability:

- Level 1

### Description:

Protect JAX-WS Web services applications by enabling HTTPS secure transport in WS-Security policy. Enabling HTTPS secure transport in WS-Security policy protects JAX-WS web services. Enable HTTPS for secure communications.

### Rationale:

Using TLS to protect the JAX-WS Web services applications will provide additional protection which may not be provided by SOAP message level security.

### Audit:

Ensure the `HttpsToken` assertion is specified for transport binding in policy specified in `wsdl` or `policy attachment` files for each web service.

```
grep -w -R -i 'HttpsToken' ${server.config.dir}
```

### Remediation:





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>
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=security-web-services-https-transport-policy-assertions>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>12.2 Establish and Maintain a Secure Network Architecture</u></b> Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.			
v7	<b><u>11.1 Maintain Standard Security Configurations for Network Devices</u></b> Maintain standard, documented security configuration standards for all authorized network devices.			

## 6.2 Ensured 'HashPassword' is set in UsernameToken WS-Security policy (Automated)

### Profile Applicability:

- Level 1

### Description:

Hashing passwords in the Username token of WS-Security policy obfuscates the password which is more secure.

### Rationale:

Using one-way hash of password in username token instead of plaintext password prevents password leaks.

### Audit:

Ensure the `HashPassword` policy assertion is specified in WS-Security policy for Username token in `wsdl` or `policy attachment` files for each web service.

```
grep -w -R -i 'HashPassword' ${server.config.dir}
```

### Remediation:




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>
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=level-authentication-web-services-clients-username-token>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.11 <u>Encrypt Sensitive Data at Rest</u></b> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	<b>14.8 <u>Encrypt Sensitive Information at Rest</u></b> 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.			

## 6.3 Ensure CallbackHandler is used to access private keys in keystore files (Manual)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Using CallbackHandler to retrieve user and keystore passwords is more secure than specifying plaintext passwords in [Liberty configuration](#).

### Audit:

Ensure that the `org.apache.ws.security.crypto.merlin.keystore.private.password` attribute is not specified in the `encryptionProperties` and `signatureProperties` element in Web Services Security client and provider configurations.

```
grep -w -R -i  
'org.apache.ws.security.crypto.merlin.keystore.private.password'  
${server.config.dir}
```

Ensure that the `ws-security.password` attribute is not specified in the Web Services Security client and provider configurations.

```
grep -w -R -i 'ws-security.password' ${server.config.dir}
```

### Remediation:

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.callback-
handler="com.myCompany.myExample.myCBH" ...>
...
</wsSecurityClient>
<wsSecurityProvider id="default" ws-security.callback-
handler="com.myCompany.myExample.myCBH" ...>
...
</wsSecurityProvider>

```

## References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=level-developing-password-callback-handler-ws-security>
2. <https://www.ibm.com/docs/en/was-liberty/base?topic=configuration-wssecurityclient>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.11 <u>Encrypt Sensitive Data at Rest</u></b> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	<b>14.8 <u>Encrypt Sensitive Information at Rest</u></b> 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.			

## 6.4 Ensure SOAP messages are Signed and encrypted with WS-Security policy (Manual)

### Profile Applicability:

- Level 1

### Description:

Signing and encrypting SOAP messages protects JAX-WS web services.

### Rationale:

Signing and encrypting SOAP messages protects against message tampering and information disclosure.

### Audit:

Ensure that `SignedParts` and `EncryptedParts` are specified in the WS-Security policy in `wsdl` or policy attachment files for each web service.

```
grep -w -R 'SignedParts\|EncryptedParts' ${server.config.dir}
```

### Remediation:

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>
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=level-protection-web-services-x509-token>



**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 6.5 Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy (Manual)

### Profile Applicability:

- Level 1

### Description:

Using keys with at least 2048 bits in length when signing and encrypting SOAP messages using WS-Security policy is more secure.

### Rationale:

Using 2048 bit keys for signing and encrypting is more secure than using 1024 bit keys.

### Audit:





Use a `keytool` to view the keys used in signing and encrypting and ensure that the key size is 2048 bits or higher.

```
keytool -list -v -keystore <keystore file name> -storepass <store password> -storetype <store type> | grep 'Signature algorithm name\|Subject Public Key Algorithm'
...
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
...
```

### Remediation:

Create signing and encryption keys with key size of 2048 bits or higher.

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>12.2 Establish and Maintain a Secure Network Architecture</b> Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.			
v7	<b>11.1 Maintain Standard Security Configurations for Network Devices</b> Maintain standard, documented security configuration standards for all authorized network devices.			

## 6.6 Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Using strong signature and encryption algorithms is more secure.

### Audit:

Ensure the `AlgorithmSuite` in WS-Security policy specifies `SHA2` signature algorithm and strong encryption algorithm like `AES 256 bits` in `wsdl` or `policy attachment` files for each web service.

```
grep -w -R -i 'AlgorithmSuite' ${server.config.dir}
```

### Remediation:





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>
```

### References:

1. <http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.html>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 <u>Encrypt Sensitive Data in Transit</u></b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 <u>Encrypt All Sensitive Information in Transit</u></b> Encrypt all sensitive information in transit.			

## 6.7 Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications (Automated)

### Profile Applicability:

- Level 1

### Description:

Enable hostname verification in JAX-WS applications by setting the `http.conduit.tlsClientParameters.disableCNCheck` HTTP conduit property to false. or Do not disable hostname verification in JAX-WS applications by setting the `http.conduit.tlsClientParameters.disableCNCheck` HTTP conduit property to true. or Enable hostname verification in JAX-WS applications. or Do not disable hostname verification in JAX-WS applications.

Disabling hostname verification in JAX-WS applications by setting the `http.conduit.tlsClientParameters.disableCNCheck` HTTP conduit property to false. Disable hostname verification in JAX-WS applications by setting the http conduit property `http.conduit.tlsClientParameters.disableCNCheck` to false.

### Rationale:

Hostname verification enables the client to trust the server it is communicating with during the SSL/TLS handshake. Hostname verification mitigates the man-in-the-middle attackers which can spoof SSL/TLS servers via an arbitrary valid certificate by ensuring that SSL server hostname matches a domain name in the subject's Common Name (CN) of the X.509 certificate.

### Audit:

Ensure the `http.conduit.tlsClientParameters.disableCNCheck` property is not set to true in the [Liberty configuration](#). Note that this can be also set in the application's `ibm-ws-bnd.xml` binding file.

```
grep -w -R 'http.conduit.tlsClientParameters.disableCNCheck'
${server.config.dir}
```

### Remediation:

Add the `http.conduit.tlsClientParameters.disableCNCheck` property to `webservice-endpoint` element in `${server.config.dir}/configDropins/overrides/<any file name>.xml`. Set the `http.conduit.tlsClientParameters.disableCNCheck` attributes value to `false` for both provider and client sides. The same setting can be disabled through `WEB-INF/ibm-ws-bnd.xml` of the web application or `META-INF/ibm-ws-bnd.xml` of the EJB module.

```

...
<!-- ***** Provider Side ***** -->
<webApplication ... >
  <webservices-bnd ... >
    <webservice-endpoint ... >
      <properties
http.conduit.tlsClientParameters.disableCNCheck="false" />
      </webservice-endpoint>
    </webservices-bnd>
  </webApplication>
...
<!-- ***** Client Side (service-ref) ***** -->
<webApplication ... >
  <webservices-bnd ... >
    <service-ref ... >
      <properties
http.conduit.tlsClientParameters.disableCNCheck="false" />
      </service-ref>
    </webservices-bnd>
  </webApplication>

```





### Default Value:

The http.conduit.tlsClientParameters.disableCNCheck property is set to false by default.

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=level-enabling-ssl-communication-web-services-access>
2. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-ws-bndxml-file>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>12.3 Securely Manage Network Infrastructure</b> Securely manage network infrastructure. Example implementations include version-controlled-infrastructure-as-code, and the use of secure network protocols, such as SSH and HTTPS.			
v7	<b>11.1 Maintain Standard Security Configurations for Network Devices</b> Maintain standard, documented security configuration standards for all authorized network devices.			

## 7 Messaging

This section covers the hardening guidelines for the messaging features.

## 7.1 Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names (Manual)

### Profile Applicability:

- Level 1

### Description:

Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address exclude list protects against unwanted inbound connections.

### Audit:

Ensure the `hostNameExcludeList` attribute is set to a list of host names in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'hostNameExcludeList' ${server.config.dir}
```

### Remediation:

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" />
```

### References:

1. <https://openliberty.io/docs/22.0.0.1/reference/config/wasJmsOutbound.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 Perform Application Layer Filtering</b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 Implement Application Firewalls</b> 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.			●



## 7.2 Ensure the 'hostNameIncludeList attribute' is set to a whitelist of host names (Manual)

### Profile Applicability:

- Level 1

### Description:

Host names can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address include list allows only wanted inbound connections.

### Audit:

Ensure the `hostNameIncludeList` attribute is set to a list of host names in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'hostNameIncludeList' ${server.config.dir}
```

### Remediation:

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" />
```

### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 Perform Application Layer Filtering</b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 Implement Application Firewalls</b> 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.			●

## 7.3 Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames (Manual)

### Profile Applicability:

- Level 1

### Description:

IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address exclude list protects against unwanted inbound connections.

### Audit:

Ensure the `addressExcludeList` attribute is set to a list of hostnames in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'addressExcludeList' ${server.config.dir}
```

### Remediation:

Add the `addressExcludeList` attribute to all `tcpOptions` elements to `${server.config.dir}/configDropins/overrides/*.xml` and set it to a comma-separated list of IP addresses.

```
<tcpOptions addressExcludeList="254.1.0.9,*.1.255.0" />
```

### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 Perform Application Layer Filtering</b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 Implement Application Firewalls</b> 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.			●

## 7.4 Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses (Manual)

### Profile Applicability:

- Level 1

### Description:

IP addresses can be allowed or blocked from creating inbound TCP connections to different HTTP endpoints.

### Rationale:

Defining an IP address include list allows only wanted inbound connections.

### Audit:

Ensure the `addressIncludeList` attribute is set to a list of IP addresses in all `tcpOptions` elements in the [Liberty configuration](#).

```
grep -w -R -i 'addressIncludeList' ${server.config.dir}
```

### Remediation:

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" />
```

### References:

1. <https://openliberty.io/docs/latest/reference/config/tcpOptions.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>13.10 Perform Application Layer Filtering</b> Perform application layer filtering. Example implementations include a filtering proxy, application layer firewall, or gateway.			●
v7	<b>9.5 Implement Application Firewalls</b> 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.			●

## 7.5 Ensure the `useSSL` attribute is set to `true` for TLS Transport (Automated)

### Profile Applicability:

- Level 1

### Description:

Configuring TLS provides secure communication for JmsOutbound connections.

### Rationale:

TLS provides secure communication channel.

### Audit:

Ensure the `useSSL` attribute for the `wasJmsOutbound` element is set to `true` in the [Liberty configuration](#).

```
grep -w -R 'useSSL' ${server.config.dir}
```

### Remediation:

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" />
```





### Default Value:

The `useSSL` attribute is `false` by default.

### References:

1. <https://openliberty.io/docs/22.0.0.1/reference/config/wasJmsOutbound.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.10 Encrypt Sensitive Data in Transit</b> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	<b>14.4 Encrypt All Sensitive Information in Transit</b> Encrypt all sensitive information in transit.			

## 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.

## 8.1 Ensure 'authentication' is set to 'true' to protect the metrics end point (Automated)

### Profile Applicability:

- Level 1

### Description:

Protect access to the metrics end point in the MicroProfile feature so that only valid users are allowed to access it.

### Rationale:

Enabling authentication for the metrics end point will ensure that only authenticated users can access it.

### Audit:

Ensure the `authentication` attribute is set to `true` in the `mpMetrics` element in the [Liberty configuration](#)

```
grep -w -R -i 'authentication' ${server.config.dir}
```

### Remediation:

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"/>
```




### Default Value:




The default value is `true`.

### References:

1. <https://openliberty.io/docs/latest/reference/config/mpMetrics.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.3 <u>Configure Data Access Control Lists</u></b> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<p><b>14.6 <u>Protect Information through Access Control Lists</u></b></p> <p>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.</p>			

## 9 z/OS

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



## 9.1 Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization (Automated)

### Profile Applicability:

- Level 1

### Description:

The SAF role mapper should be used to perform SAF authorization checks when accessing applications.

### Rationale:

On the z/OS platform using the native z/OS facilities like the SAF authorization is recommended for application access checks.

### Audit:

Ensure the `zosSecurity-1.0` feature is enabled and the `safAuthorization` element is configured in the **Liberty configuration**.

```
grep -w -R 'zosSecurity-1.0' ${server.config.dir}
grep -w -R 'safAuthorization' ${server.config.dir}
```

### Remediation:



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" />
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/zos?topic=liberty-configuring-authorization-applications-in>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	12.5 <u>Centralize Network Authentication, Authorization, and Auditing (AAA)</u> Centralize network AAA.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<b>16.2 <u>Configure Centralized Point of Authentication</u></b> Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.			

## 9.2 Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates (Automated)

### Profile Applicability:

- Level 1

### Description:

The SAF Keyrings is a z/OS facility to hold the certificates that are used during the SSL/TLS communication.

### Rationale:

On z/OS using the native z/OS facilities like the SAF Keyrings to manage the SSL/TLS certificates is recommended.

### Audit:

Ensure the `location` attribute in the SSL configurations points to a valid SAF Keyring and set the `fileBased` attribute to false in the [Liberty configuration](#).

```
grep -i -R 'safkeyring:' ${server.config.dir}
grep -i -R 'fileBased' ${server.config.dir}
```

### Remediation:

Configure the `location` attribute in the `keyStore` elements referenced by the SSL configurations to point to a valid SAF Keyring location that contains the SSL/TLS certificates and set the `fileBased` attribute to false in





`${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<keyStore id="DefaultKeyStore" location="safkeyring:///Keyring.LIBERTY"
type="JCERACFKS" fileBased="false" readOnly="true" />
<keyStore id="DefaultTrustStore" location="safkeyring:///Keyring.LIBERTY"
type="JCERACFKS" fileBased="false" readOnly="true" />
```

### References:

1. <https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-enabling-ssl-communication-in>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>12.5 Centralize Network Authentication, Authorization, and Auditing (AAA)</u> Centralize network AAA.			
v7	<u>16.2 Configure Centralized Point of Authentication</u> Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.			

## 9.3 Ensure 'safkeyringhw:' is set to use a hardware crypto card (Manual)

### Profile Applicability:

- Level 2

### Description:

Use hardware crypto card to store to store cryptographic keys and certificates.

### Rationale:

Using hardware crypto card to store and load cryptographic keys and certificates. Using hardware crypto cards can provide additional security guards against theft. The keys and certificates are used during cryptographic functions like encryption/decryption and also during the SSL/TLS communication.

### Audit:

Hardware crypto cards are external devices that can store certificates that can be used for the SSL/TLS communications. They store and protect the cryptographic keys throughout their lifecycles. The use of these devices have to be weighed against the security requirements and the additional expenses this would entail.

There are *two* ways a hardware crypto card can be used. One can configure to use the keys contained in only the hardware crypto card or configure a hybrid provider that can handle keys in both the crypto card as well as in the software.

When using just the hardware crypto card, ensure that `safkeyringhw:` is set correctly in the `location` attribute of the keyStore elements of the SSL configurations. In addition, make sure that `type` attribute is set to `JCECCARACFKS` in the [Liberty configuration](#).

```
grep -i -R 'safkeyringhw:' ${server.config.dir}
grep -i -w -R 'type' ${server.config.dir}
```

Also make sure that the `com.ibm.crypto.hdwrCCA.provider.IBMJCECCA` provider is listed in the `$JAVA_HOME/jre/lib/security/java.security` file.

```
grep -w -R 'com.ibm.crypto.hdwrCCA.provider.IBMJCECCA' ${JAVA_HOME}
```

When using the hybrid provider, ensure that `safkeyringhybrid:` is set correctly in the `location` attribute of the keyStore elements of the SSL configurations. In addition, make sure that `type` attribute is set to `JCEHYBRIDRACFKS` in the [Liberty configuration](#).

```
grep -i -R 'safkeyringhybrid:' ${server.config.dir}
grep -i -w -R 'type' ${server.config.dir}
```

Also make sure that both the `com.ibm.crypto.ibmjcehybrid.provider.IBMJCEHYBRID` and the `com.ibm.crypto.hdwrCCA.provider.IBMJCECCA` providers are listed in the `$JAVA_HOME/jre/lib/security/java.security` file.

```
grep -w -R 'com.ibm.crypto.ibmjcehybrid.provider.IBMJCEHYBRID' ${JAVA_HOME}
grep -w -R 'com.ibm.crypto.hdwrCCA.provider.IBMJCECCA' ${JAVA_HOME}
```

## Remediation:

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
...
```

## References:

1. <https://www.ibm.com/support/pages/enabling-hardware-cryptography-liberty-zos>
2. <https://www.ibm.com/docs/en/was-liberty/zos?topic=liberty-enabling-jce-hybrid-provider>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>12.6 Use of Secure Network Management and Communication Protocols</u></b> Use secure network management and communication protocols (e.g., 802.1X, Wi-Fi Protected Access 2 (WPA2) Enterprise or greater).			
v7	<b><u>11.5 Manage Network Devices Using Multi-Factor Authentication and Encrypted Sessions</u></b> Manage all network devices using multi-factor authentication and encrypted sessions.			

## 9.4 Ensure 'safRegistry' is configured (Automated)

### Profile Applicability:

- Level 1

### Description:

The SAF user registry in the z/OS Operating System is robust and secure.

### Rationale:

On z/OS using the native z/OS facilities like the SAF registry for authentication is recommended.

### Audit:

Ensure the `zosSecurity-1.0` feature is enabled and the `safRegistry` element is configured in the **Liberty configuration**.

```
grep -w -R 'zosSecurity-1.0' ${server.config.dir}
grep -w -R 'safRegistry' ${server.config.dir}
```

### Remediation:

Configure the `zosSecurity-1.0` feature and set the `safRegistry` element in `${server.config.dir}/configDropins/overrides/<any file name>.xml`.

```
<feature>zosSecurity-1.0</feature>
<safRegistry realm="myrealm" />
```

### References:

1. <https://www.ibm.com/docs/el/was-liberty/zos?topic=liberty-activating-configuring-saf-registry-zos>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>12.5 Centralize Network Authentication, Authorization, and Auditing (AAA)</u> Centralize network AAA.		●	●
v7	<u>16.2 Configure Centralized Point of Authentication</u> Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.		●	●



## **10 Miscellaneous**

Includes miscellaneous recommendations

## 10.1 Ensure Unused Features are Removed (Automated)

### Profile Applicability:

- Level 1

### Description:

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.

### Rationale:

Enabling only the necessary [Liberty features](#) minimizes the disk and memory footprint of the server as well as a faster start time. Having unused features could increase the chance of having a vulnerability due to the default settings of some features.

### Audit:

Review the configuration for features not being used in [Liberty configuration](#). This includes aggregated features containing multiple features. The features are listed under in the `featureManager` element.

To review all of the individual features used in the server, review the `messages.log` for the `CWWKF0012I` message.

```
grep -w -R -i CWWKF0012I ${server.config.dir}
```

### Remediation:

Removed any unneeded features listed under the `featureManager` element in the [Liberty configuration](#)

### Default Value:

The default server template enables the `jsp-2.3` feature. Other templates in the `${wlp.install.dir}/templates` directory add different features depending on the template theme.









Example of the `defaultServer` template features in the `server.xml` file after running `${wlp.install.dir}/bin server create myServer`

```
<featureManager>
  <feature>jsp-2.3</feature>
</featureManager>
```

### References:

1. <https://openliberty.io/docs/latest/reference/feature/feature-overview.html>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b><u>4.4 Implement and Manage a Firewall on Servers</u></b> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v8	<b><u>4.5 Implement and Manage a Firewall on End-User Devices</u></b> 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.			
v7	<b><u>9.2 Ensure Only Approved Ports, Protocols and Services Are Running</u></b> Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.			

## 10.2 Ensure Passwords are Encrypted (Automated)

### Profile Applicability:

- Level 1

### Description:

Sensitive information is stored in the [Liberty configuration](#) and can be in clear text unless encrypted or hashed.

### Rationale:

Passwords should be encrypted or hashed to avoid inappropriate access to user accounts or using the passwords for unauthorized logins to remote systems such as databases or LDAP servers. Password information should also be kept in separate files with limited access to avoid unauthorized access or changes to configurations. The password encryption key also needs to be customized or a default key is used to encrypt passwords.

### Audit:

Review the passwords in the [Liberty configuration](#) and the `${wlp.user.dir}/clients/<Client Name>/client.xml` files.

Check the values on any password related attributes that are defined as `Reversably encoded password (string)` type. Some examples of password related attributes include `password`, `bindPassword`, `secret`, `keyStoreSecret`, etc. All values should be obscured using encryption or hashing.

Some example greps to check for attributes containing `password` or `secret` in the name.

```
grep -R -i 'password' ${server.config.dir}
grep -R -i 'secret' ${server.config.dir}
```

Also check for the `wlp.password.encryption.key` property that is used for encrypting and decrypting and where it is located. The `wlp.password.encryption.key` property should be set to a custom value and stored in a separate file.

```
grep -R -i 'wlp.password.encryption.key' ${server.config.dir}
```

If the `wlp.password.encryption.key` property is in a standalone file, review the permissions for accessing the file.

```
ls -al <keyFileName.xml>
```

### Remediation:

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/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3ev396oYRw=="
</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
```







### Default Value:

Passwords are not encrypted by default in the [Liberty configuration](#).

### References:

1. <https://openliberty.io/docs/latest/reference/command/securityUtility-encode.html>
2. <https://openliberty.io/docs/latest/reference/config/ldapRegistry.html>

**CIS Controls:**

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 <u>Securely Dispose of Data</u></b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	<b>13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b> 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.			

## 10.3 Ensure 'enableWelcomePage' is set to 'false' (Automated)

### Profile Applicability:

- Level 1

### Description:

The WebSphere Liberty welcome page is enabled by default and is displayed when the root context “/” is accessed.

### Rationale:

For production, the Liberty welcome page should be disabled to avoid sharing unnecessary information about the server runtime.

### Audit:

Ensure the `enableWelcomePage` attribute is set to false in the `httpDispatcher` element in the [Liberty configuration](#):

```
grep -w -R -i 'enableWelcomePage' ${server.config.dir}
```

### Remediation:

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" />
```




### Default Value:




The WebSphere Liberty welcome page is enabled by default, the `enableWelcomePage` attribute is set to `true`.

### References:

1. <https://openliberty.io/docs/latest/reference/config/httpDispatcher.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>3.5 Securely Dispose of Data</b> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<p><b><u>13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u></b></p> <p>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.</p>			



## 10.4 Ensure 'keysPassword' is set to a custom password for ltpa keys (Automated)

### Profile Applicability:

- Level 1

### Description:

The LTPA keys are generated using a password, if a password is not provided, then a default password is used.

### Rationale:

The LTPA password should be customized to avoid using the default password.

### Audit:

Ensure the existence of a `keysPassword` attribute on the `ltpa` element in the [Liberty configuration](#). The value should be obscured using encryption.

```
grep -R -i 'keysPassword' ${server.config.dir}
```

### Remediation:

Add a custom encrypted password to the `keysPassword` attribute on the `ltpa` element in the [Liberty configuration](#).

```
<ltpa  
keysPassword="{aes}AE/PrLc9wshAKURioFvxb41SrVbsWjZTZ8lv72ioH3yMlJN4RQj3A9aT3e  
v396oYRw==" >
```




### Default Value:




WebSphere Liberty provides a default password for the LTPA keys.

### References:

1. <https://openliberty.io/docs/latest/reference/config/ltpa.html>
2. <https://openliberty.io/docs/latest/reference/command/securityUtility-encode.html>

### CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>5.2 Use Unique Passwords</b> Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	<b>4.2 Change Default Passwords</b> Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.			

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

### Profile Applicability:

- Level 1

### Description:

Configuring authorization for your application is to verify whether a user or group belongs to a specified role, and whether this role has the privilege to access a resource.

### Rationale:

Defining authorization roles ensures users cannot access resources they are not authorized to use.

### Audit:

Ensure that `security-role` is defined mapping any desired roles in the [Liberty configuration](#).

```
grep -w -R 'security-role' ${server.config.dir}
```

### Remediation:

Create the necessary role mappings for applications in the  
`${server.config.dir}/configDropins/overrides/<any file name>.xml`

#### Example:

```
<application-bnd>
  <security-role name="admin">
    <group name="Manager" />
    <group name="TeamLead" />
  </security-role>
  <security-role name="user">
    <group name="Employee" />
  </security-role>
</application-bnd>
```

Follow steps in the [Liberty Authorization Doc](#)

### References:

1. <https://openliberty.io/docs/latest/authorization.html>

## CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<b>6.8 <u>Define and Maintain Role-Based Access Control</u></b> 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.			●
v7	<b>14.6 <u>Protect Information through Access Control Lists</u></b> 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.	●	●	●

## 11 Appendices

## 11.1 Liberty configuration overview

Liberty's runtime environment operates from a set of built-in configuration default settings, and you only need to specify configuration that overrides those default settings. You do this by editing either the server.xml file or another XML file that is included in server.xml and any XML files in the configDropins directory at run time.

The Liberty configuration service parses the primary server.xml file and any files it includes, as well as the configuration files in the configDropins directory, merges the contents over the default configuration values provided by the installed bundles, then feeds the resulting property sets into the OSGi Configuration Admin Service (CA). CA injects each set of properties into the service that owns the set, if it is registered with CA.

For more information about server.env, jvm.options, bootstrap.properties beside Websphere Liberty Configuration please see more information about [Liberty Server Configuration Overview](#)

To summarize from the above information, here is the precedence order (lower to higher) the configuration is read during runtime

1. Defaults defined by the server
2. \${server.config.dir}/configDropins/defaults/\*.xml (Beware of alphabetical priority order - A later configuration overrides an earlier one)
3. \${server.config.dir}/server.xml
4. \${server.config.dir}/configDropins/overrides/\*.xml (Based on alphabetical order - A later configuration overrides an earlier one)

## 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 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		Set Correctly	
		Yes	No
<b>1</b>	<b>Install and Setup</b>		
1.1	Ensure root does not have ownership of Websphere Liberty binaries (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Ensure extraneous files and directories are removed (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Ensure only defined users have access to the file system (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.6	Ensure automated configuration updates are disabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>2</b>	<b>User Registries</b>		
2.1	Ensure 'displayAuthenticationRealm' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Ensure that the LDAP connection uses TLS (Automated)	<input type="checkbox"/>	<input type="checkbox"/>



CIS Benchmark Recommendation		Set Correctly	
		Yes	No
<b>3</b>	<b>Application Deployment</b>		
3.1	Ensure that automatic applications updates are disabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.2	Ensure JDK Security Manager is Enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4</b>	<b>Web Applications</b>		
<b>4.1</b>	<b>Securing Cookies</b>		
<b>4.1.1</b>	<b>Securing Session Cookies</b>		
4.1.1.1	Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.2	Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.3	Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies. (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.4	Ensure 'cookieSecure' secure attribute is set to 'true' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4.1.2</b>	<b>Securing Authentication Cookies</b>		
4.1.2.1	Ensure 'sameSiteCookie' attribute is set to 'Strict' (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.2	Ensure 'ssoDomainNames' attribute is configured for the authentication cookies. (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.3	Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie. (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.4	Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.5	Ensure 'ssoCookieName' LTPA cookie name is set (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.1.2.6	Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.7	Ensure 'trackLoggedOutSSOCookies' is set to 'true' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.8	Ensure 'cookieName' JWT (JSON Web Token) cookie name is set (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4.1.3</b>	<b>Securing Other Cookies</b>		
4.1.3.1	Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4.2</b>	<b>Secure Transport</b>		
4.2.1	Ensure 'trustDefaultCerts' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.3	Ensure HSTS (HTTP Strict Transport Security) is enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.4	Ensure that outbound TLS configurations are specified (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.5	Ensure that secure ciphers suites are configured (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.6	Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.7	Ensure Hostname verification for TLS communication is enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.8	Ensure that CA (Certificate Authority) certificates are used (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.9	Ensure 'ocsp.enable' certificate revocation is set to 'true' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.2.10	Ensure mutual TLS authentication is enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.11	Ensure that strong algorithms are used for TLS certificates. (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.12	Ensure `httpPort` attribute set to `-1` (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.13	Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.14	Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms. (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.15	Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4.3</b>	<b>Single Sign On (SSO)</b>		
4.3.1	Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.2	Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.3	Ensure 'tokenReuse' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.4	Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.5	Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.6	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.7	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.3.8	Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.9	Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.10	Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0 (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.11	Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0 (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.12	Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0 (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.13	Ensure 'httpsRequired' is set to 'true' in OAuth 2.0 (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.14	Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0 (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.15	Ensure 'httpsRequired' is set to 'true' in SAML (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.16	Enforce 'wantAssertionsSigned' to 'true' in SAML (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.17	Ensure 'authnRequestsSigned' is set to 'true' in SAML (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>4.4</b>	<b>General</b>		
4.4.1	Ensure 'disableXPowereBy' is set to 'true' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.2	Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.3	Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.4.4	Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.5	Ensure 'logoutOnHttpSessionExpire' is set to 'true' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.6	Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.7	Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.8	Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.9	Ensure "trustedSensitiveHeaderOrigin" is set to trusted host names and IP addresses for sensitive data (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.10	Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.11	Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.12	Ensure security constraints are specified to protect web applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.13	Ensure application security feature is enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.14	Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.15	Ensure Web Server Document Root does not contain information that should be private (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.16	Ensure HTTP session overflow is 'disabled' (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.17	Ensure uncovered http methods are denied (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.4.18	Ensure 'disallowServeServletsByClassName' is 'disabled' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.22	Ensure virtual hosts are defined to isolate applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.23	Ensure virtual hosts are Defined to isolate JMX communication and application traffic (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4.24	Ensure whitelisting of virtual hosts to validate access based on originating endpoint (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>5</b>	<b>Enterprise Java Beans (EJB) Applications</b>		
<b>5.1</b>	<b>The CSiv2 (Common Secure Interoperability version 2) serverPolicy</b>		
5.1.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
5.1.2	Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
5.1.3	Ensure 'identityAssertionEnabled' is set to 'true' within the CSiv2 Attribute Layer (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>5.2</b>	<b>The CSiv2 (Common Secure Interoperability version 2) Client Policy</b>		
5.2.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech (Manual)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5.2.2	Ensure 'establishTrustInClient' is 'Required' for the CSiv2 Authentication Layer - needsReview/Zech (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.2.3	Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSiv2 Attribute Layer - review/Zech (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
<b>5.3</b>	<b>Java Serialization</b>		
5.3.1	Ensure filters are configured for Java serialization (JEP 290) (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
<b>5.4</b>	<b>EJB Authentication</b>		
5.4.1	Ensure that all appropriate EJB methods are protected (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>6</b>	<b>Web Services</b>		
6.1	Ensure 'HttpsToken' is set in WS-Security policy (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.2	Ensured 'HashPassword' is set in UsernameToken WS-Security policy (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.3	Ensure CallbackHandler is used to access private keys in keystore files (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure SOAP messages are Signed and encrypted with WS-Security policy (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
6.6	Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.7	Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>7</b>	<b>Messaging</b>		
7.1	Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
7.2	Ensure the 'hostNameIncludeList' attribute is set to a whitelist of host names (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
7.3	Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
7.4	Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure the 'useSSL' attribute is set to 'true' for TLS Transport (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>8</b>	<b>MicroProfile Metrics</b>		
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>9</b>	<b>z/OS</b>		
9.1	Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
9.2	Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'safkeyringhw:' is set to use a hardware crypto card (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
9.4	Ensure 'safRegistry' is configured (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>10</b>	<b>Miscellaneous</b>		



CIS Benchmark Recommendation		Set Correctly	
		Yes	No
10.1	Ensure Unused Features are Removed (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
10.5	Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
<b>11</b>	<b>Appendices</b>		
<b>11.1</b>	<b>Liberty configuration overview</b>		
<b>11.2</b>	<b>Liberty Features Overview</b>		

# Appendix: CIS Controls v7 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Ensure only defined users have access to the file system	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>
10.5	Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: CIS Controls v7 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Ensure extraneous files and directories are removed	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Ensure only defined users have access to the file system	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Ensure that the LDAP connection uses TLS	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.2	Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.4	Ensure 'cookieSecure' secure attribute is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.4	Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.1	Ensure 'trustDefaultCerts' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.3	Ensure HSTS (HTTP Strict Transport Security) is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.4	Ensure that outbound TLS configurations are specified	<input type="checkbox"/>	<input type="checkbox"/>
4.2.6	Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.8	Ensure that CA (Certificate Authority) certificates are used	<input type="checkbox"/>	<input type="checkbox"/>
4.2.10	Ensure mutual TLS authentication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.12	Ensure `httpPort` attribute set to `-1`	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.1.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.2.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
6.1	Ensure 'HttpsToken' is set in WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure SOAP messages are Signed and encrypted with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.6	Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.7	Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure the `useSSL` attribute is set to `true` for TLS Transport	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
9.1	Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
9.2	Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'safkeyringhw:' is set to use a hardware crypto card	<input type="checkbox"/>	<input type="checkbox"/>
9.4	Ensure 'safRegistry' is configured	<input type="checkbox"/>	<input type="checkbox"/>
10.1	Ensure Unused Features are Removed	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>
10.5	Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: CIS Controls v7 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Ensure extraneous files and directories are removed	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Ensure only defined users have access to the file system	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Ensure that the LDAP connection uses TLS	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.2	Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.4	Ensure 'cookieSecure' secure attribute is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.4	Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.1	Ensure 'trustDefaultCerts' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.3	Ensure HSTS (HTTP Strict Transport Security) is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.4	Ensure that outbound TLS configurations are specified	<input type="checkbox"/>	<input type="checkbox"/>
4.2.6	Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.8	Ensure that CA (Certificate Authority) certificates are used	<input type="checkbox"/>	<input type="checkbox"/>
4.2.10	Ensure mutual TLS authentication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.12	Ensure `httpPort` attribute set to `-1`	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.1.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.1.3	Ensure 'identityAssertionEnabled' is set to 'true' within the CSiv2 Attribute Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.2.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.2.3	Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSiv2 Attribute Layer - review/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
6.1	Ensure 'HttpsToken' is set in WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.2	Ensured 'HashPassword' is set in UsernameToken WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.3	Ensure CallbackHandler is used to access private keys in keystore files	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure SOAP messages are Signed and encrypted with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.6	Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
6.7	Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications	<input type="checkbox"/>	<input type="checkbox"/>
7.1	Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names	<input type="checkbox"/>	<input type="checkbox"/>
7.2	Ensure the 'hostNameIncludeList' attribute is set to a whitelist of host names	<input type="checkbox"/>	<input type="checkbox"/>
7.3	Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames	<input type="checkbox"/>	<input type="checkbox"/>
7.4	Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure the 'useSSL' attribute is set to 'true' for TLS Transport	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
9.1	Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization	<input type="checkbox"/>	<input type="checkbox"/>
9.2	Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'safkeyringhw:' is set to use a hardware crypto card	<input type="checkbox"/>	<input type="checkbox"/>
9.4	Ensure 'safRegistry' is configured	<input type="checkbox"/>	<input type="checkbox"/>
10.1	Ensure Unused Features are Removed	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>
10.5	Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications	<input type="checkbox"/>	<input type="checkbox"/>



# Appendix: CIS Controls v7 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.6	Ensure automated configuration updates are disabled	<input type="checkbox"/>	<input type="checkbox"/>
2.1	Ensure 'displayAuthenticationRealm' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
3.1	Ensure that automatic applications updates are disabled	<input type="checkbox"/>	<input type="checkbox"/>
3.2	Ensure JDK Security Manager is Enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.1	Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.3	Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.1	Ensure 'sameSiteCookie' attribute is set to 'Strict'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.2	Ensure 'ssoDomainNames' attribute is configured for the authentication cookies.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.3	Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.5	Ensure 'ssoCookieName' LTPA cookie name is set	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.6	Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.7	Ensure 'trackLoggedOutSSOCookies' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.8	Ensure 'cookieName' JWT (JSON Web Token) cookie name is set	<input type="checkbox"/>	<input type="checkbox"/>
4.1.3.1	Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.5	Ensure that secure ciphers suites are configured	<input type="checkbox"/>	<input type="checkbox"/>
4.2.7	Ensure Hostname verification for TLS communication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.9	Ensure 'ocsp.enable' certificate revocation is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.11	Ensure that strong algorithms are used for TLS certificates.	<input type="checkbox"/>	<input type="checkbox"/>
4.2.13	Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.14	Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms.	<input type="checkbox"/>	<input type="checkbox"/>
4.2.15	Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules	<input type="checkbox"/>	<input type="checkbox"/>
4.3.1	Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens	<input type="checkbox"/>	<input type="checkbox"/>
4.3.2	Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.3	Ensure 'tokenReuse' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.3.4	Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.5	Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.6	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.7	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.8	Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.9	Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.10	Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.11	Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.12	Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.13	Ensure 'httpsRequired' is set to 'true' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.14	Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.15	Ensure 'httpsRequired' is set to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.3.16	Enforce 'wantAssertionsSigned' to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.3.17	Ensure 'authnRequestsSigned' is set to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.4.2	Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.4.3	Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects	<input type="checkbox"/>	<input type="checkbox"/>
4.4.4	Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.5	Ensure 'logoutOnHttpSessionExpire' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.6	Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.7	Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.8	Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.9	Ensure 'trustedSensitiveHeaderOrigin' is set to trusted host names and IP addresses for sensitive data	<input type="checkbox"/>	<input type="checkbox"/>
4.4.10	Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses	<input type="checkbox"/>	<input type="checkbox"/>
4.4.11	Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout	<input type="checkbox"/>	<input type="checkbox"/>
4.4.12	Ensure security constraints are specified to protect web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.13	Ensure application security feature is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.4.14	Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.15	Ensure Web Server Document Root does not contain information that should be private	<input type="checkbox"/>	<input type="checkbox"/>
4.4.16	Ensure HTTP session overflow is 'disabled'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.17	Ensure uncovered http methods are denied	<input type="checkbox"/>	<input type="checkbox"/>
4.4.18	Ensure 'disallowServeServletsByClassName' is 'disabled'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.22	Ensure virtual hosts are defined to isolate applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.23	Ensure virtual hosts are Defined to isolate JMX communication and application traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.24	Ensure whitelisting of virtual hosts to validate access based on originating endpoint	<input type="checkbox"/>	<input type="checkbox"/>
5.1.2	Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
5.2.2	Ensure 'establishTrustInClient' is 'Required' for the CSlv2 Authentication Layer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: CIS Controls v8 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
10.1	Ensure Unused Features are Removed	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: CIS Controls v8 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Ensure extraneous files and directories are removed	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.6	Ensure automated configuration updates are disabled	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Ensure that the LDAP connection uses TLS	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.2	Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.4	Ensure 'cookieSecure' secure attribute is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.4	Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.1	Ensure 'trustDefaultCerts' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.3	Ensure HSTS (HTTP Strict Transport Security) is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.4	Ensure that outbound TLS configurations are specified	<input type="checkbox"/>	<input type="checkbox"/>
4.2.6	Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.8	Ensure that CA (Certificate Authority) certificates are used	<input type="checkbox"/>	<input type="checkbox"/>
4.2.10	Ensure mutual TLS authentication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.12	Ensure `httpPort` attribute set to `-1`	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.1.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.1.2	Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.2.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.2.2	Ensure 'establishTrustInClient' is 'Required' for the CSiv2 Authentication Layer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
6.1	Ensure 'HttpsToken' is set in WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.2	Ensured 'HashPassword' is set in UsernameToken WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.3	Ensure CallbackHandler is used to access private keys in keystore files	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure SOAP messages are Signed and encrypted with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.6	Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
6.7	Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure the 'useSSL' attribute is set to 'true' for TLS Transport	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
9.1	Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization	<input type="checkbox"/>	<input type="checkbox"/>
9.2	Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'safkeyringhw:' is set to use a hardware crypto card	<input type="checkbox"/>	<input type="checkbox"/>
9.4	Ensure 'safRegistry' is configured	<input type="checkbox"/>	<input type="checkbox"/>
10.1	Ensure Unused Features are Removed	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>



# Appendix: CIS Controls v8 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Ensure root does not have ownership of Websphere Liberty binaries	<input type="checkbox"/>	<input type="checkbox"/>
1.2	Ensure extraneous files and directories are removed	<input type="checkbox"/>	<input type="checkbox"/>
1.3	Ensure only defined users have access to the file system	<input type="checkbox"/>	<input type="checkbox"/>
1.4	Ensure that only one user ID has write access to the WebSphere Liberty configuration files	<input type="checkbox"/>	<input type="checkbox"/>
1.5	Ensure Websphere Liberty Server Output is not set to the default value	<input type="checkbox"/>	<input type="checkbox"/>
1.6	Ensure automated configuration updates are disabled	<input type="checkbox"/>	<input type="checkbox"/>
1.7	Ensure the WebSphere Liberty Installation is Validated	<input type="checkbox"/>	<input type="checkbox"/>
1.8	Ensure Websphere Liberty file system access is Restricted	<input type="checkbox"/>	<input type="checkbox"/>
1.9	Ensure that the 'onConflict attribute' is set to 'IGNORE' to restrict config file overwrites	<input type="checkbox"/>	<input type="checkbox"/>
2.2	Ensure Basic Registry and Quick Start security Registry are Removed	<input type="checkbox"/>	<input type="checkbox"/>
2.3	Ensure that the LDAP connection uses TLS	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.2	Ensure 'cookieHttpOnly' HttpOnly attribute is set to 'true' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.4	Ensure 'cookieSecure' secure attribute is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.4	Ensure 'ssoRequiresSSL' secure attribute is set to 'true' for the LTPA Cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.1	Ensure 'trustDefaultCerts' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.2	Ensure 'sslProtocol' is set to the latest versions of TLS (Transport Layer Security)	<input type="checkbox"/>	<input type="checkbox"/>
4.2.3	Ensure HSTS (HTTP Strict Transport Security) is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.4	Ensure that outbound TLS configurations are specified	<input type="checkbox"/>	<input type="checkbox"/>
4.2.6	Ensure 'transport-guarantee' is set to 'CONFIDENTIAL' for all web applications	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.8	Ensure that CA (Certificate Authority) certificates are used	<input type="checkbox"/>	<input type="checkbox"/>
4.2.10	Ensure mutual TLS authentication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.12	Ensure `httpPort` attribute set to `-1`	<input type="checkbox"/>	<input type="checkbox"/>
4.4.1	Ensure 'disableXPoweredBy' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.19	Ensure server headers on requests are removed	<input type="checkbox"/>	<input type="checkbox"/>
4.4.20	Ensure 'directoryBrowsingEnabled' is set to 'false' for web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.21	Ensure 'default-error-page' is set for web applications	<input type="checkbox"/>	<input type="checkbox"/>
5.1.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 Transport Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.1.2	Ensure 'establishTrustInClient' is set to 'required' within the CSiv2 Authentication Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.1.3	Ensure 'identityAssertionEnabled' is set to 'true' within the CSiv2 Attribute Layer	<input type="checkbox"/>	<input type="checkbox"/>
5.2.1	Ensure 'sslEnabled' is set to 'true' within the CSiv2 TransportLayer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.2.2	Ensure 'establishTrustInClient' is 'Required' for the CSiv2 Authentication Layer - needsReview/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.2.3	Ensure 'identityAssertionTypes' is specified to the correct identity tokens in CSiv2 Attribute Layer - review/Zech	<input type="checkbox"/>	<input type="checkbox"/>
5.3.1	Ensure filters are configured for Java serialization (JEP 290)	<input type="checkbox"/>	<input type="checkbox"/>
5.4.1	Ensure that all appropriate EJB methods are protected	<input type="checkbox"/>	<input type="checkbox"/>
6.1	Ensure 'HttpsToken' is set in WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.2	Ensured 'HashPassword' is set in UsernameToken WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.3	Ensure CallbackHandler is used to access private keys in keystore files	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure SOAP messages are Signed and encrypted with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure that 2048 bit keys are used for signing and encrypting SOAP messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
6.6	Ensure 'AlgorithmSuite' is set to that strong algorithms for signing and encrypting messages with WS-Security policy	<input type="checkbox"/>	<input type="checkbox"/>
6.7	Ensure 'http.conduit.tlsClientParameters.disableCNCheck' is set to 'false' to enable hostname verification for JAX-WS applications	<input type="checkbox"/>	<input type="checkbox"/>
7.1	Ensure the 'hostNameExcludeList' attribute is set to a whitelist of host names	<input type="checkbox"/>	<input type="checkbox"/>
7.2	Ensure the 'hostNameIncludeList' attribute is set to a whitelist of host names	<input type="checkbox"/>	<input type="checkbox"/>
7.3	Ensure the 'addressExcludeList' attribute is set to a whitelist of hostnames	<input type="checkbox"/>	<input type="checkbox"/>
7.4	Ensure the 'addressIncludeList' attribute is set to a whitelist of IP addresses	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure the 'useSSL' attribute is set to 'true' for TLS Transport	<input type="checkbox"/>	<input type="checkbox"/>
8.1	Ensure 'authentication' is set to 'true' to protect the metrics end point	<input type="checkbox"/>	<input type="checkbox"/>
9.1	Ensure 'zosSecurity-1.0' feature is 'enabled' for SAF authorization	<input type="checkbox"/>	<input type="checkbox"/>
9.2	Ensure the location attribute in the SSL configurations points to a valid SAF Keyring containing SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'safkeyringhw:' is set to use a hardware crypto card	<input type="checkbox"/>	<input type="checkbox"/>
9.4	Ensure 'safRegistry' is configured	<input type="checkbox"/>	<input type="checkbox"/>
10.1	Ensure Unused Features are Removed	<input type="checkbox"/>	<input type="checkbox"/>
10.2	Ensure Passwords are Encrypted	<input type="checkbox"/>	<input type="checkbox"/>
10.3	Ensure 'enableWelcomePage' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
10.4	Ensure 'keysPassword' is set to a custom password for ltpa keys	<input type="checkbox"/>	<input type="checkbox"/>
10.5	Ensure 'security-role' is defined for role based authorization checks for Web and EJB applications	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: CIS Controls v8 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
2.1	Ensure 'displayAuthenticationRealm' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
3.1	Ensure that automatic applications updates are disabled	<input type="checkbox"/>	<input type="checkbox"/>
3.2	Ensure JDK Security Manager is Enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.1	Ensure 'cookieSameSite' SameSite attribute is set to 'Strict' for session cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.1.3	Ensure 'cookieDomain' cookie domain name attribute is set for the session cookies.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.1	Ensure 'sameSiteCookie' attribute is set to 'Strict'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.2	Ensure 'ssoDomainNames' attribute is configured for the authentication cookies.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.3	Ensure 'setCookieSecureFlag' secure attribute is set to 'true' for the `JWT` cookie.	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.5	Ensure 'ssoCookieName' LTPA cookie name is set	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.6	Ensure 'httpOnlyCookies' HttpOnly attribute is set to 'True' for the authentication cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.7	Ensure 'trackLoggedOutSSOCookies' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.1.2.8	Ensure 'cookieName' JWT (JSON Web Token) cookie name is set	<input type="checkbox"/>	<input type="checkbox"/>
4.1.3.1	Ensure 'samesite' SameSite attribute is set to 'Strict' for additional cookies	<input type="checkbox"/>	<input type="checkbox"/>
4.2.5	Ensure that secure ciphers suites are configured	<input type="checkbox"/>	<input type="checkbox"/>
4.2.7	Ensure Hostname verification for TLS communication is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.2.9	Ensure 'ocsp.enable' certificate revocation is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.2.11	Ensure that strong algorithms are used for TLS certificates.	<input type="checkbox"/>	<input type="checkbox"/>
4.2.13	Ensure that hardware crypto cards/modules (HSM) are used to store SSL/TLS certificates	<input type="checkbox"/>	<input type="checkbox"/>
4.2.14	Ensure SP800-131a recommendation is used for stronger cryptographic keys and more robust algorithms.	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.2.15	Ensure that the Federal Information Processing Standards (FIPS) are used for the cryptographic modules	<input type="checkbox"/>	<input type="checkbox"/>
4.3.1	Ensure 'signatureAlgorithm' asymmetric key algorithm is set for encrypting the JSON Web Tokens	<input type="checkbox"/>	<input type="checkbox"/>
4.3.2	Ensure that constrained delegation is configured for SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.3	Ensure 'tokenReuse' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.3.4	Ensure 'disableIssChecking' issuer claim is set to 'false' in the RP (Relying Party)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.5	Ensure 'hostNameVerificationEnabled' is set to 'true' in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.6	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.7	Ensure 'signatureAlgorithm' is set to a secure algorithm in OIDC Provider (OP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.8	Ensure 'httpsRequired' is set to 'true' in OIDC Relying Party (RP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.9	Ensure 'tokenEndpointAuthMethodsSupported' is set to a valid authentication method in OIDC Provider (OP)	<input type="checkbox"/>	<input type="checkbox"/>
4.3.10	Ensure 'accessTokenEncoding' is set to a strong hash algorithm in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.11	Ensure 'allowPublicClients' is set to 'false' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.12	Ensure 'clientSecretEncoding' is set to a strong encoding type in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.13	Ensure 'httpsRequired' is set to 'true' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.14	Ensure 'skipResourceOwnerValidation' is set to 'false' in OAuth 2.0	<input type="checkbox"/>	<input type="checkbox"/>
4.3.15	Ensure 'httpsRequired' is set to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.3.16	Enforce 'wantAssertionsSigned' to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.3.17	Ensure 'authnRequestsSigned' is set to 'true' in SAML	<input type="checkbox"/>	<input type="checkbox"/>
4.4.2	Ensure 'preserveFullyQualifiedReferrerUrl' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.3	Ensure 'logoutPageRedirectDomainNames' is set to relevant domain names for logout page redirects	<input type="checkbox"/>	<input type="checkbox"/>

Recommendation		Set Correctly	
		Yes	No
4.4.4	Ensure 'hostNameExcludeList' is set to the hostnames to be excluded for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.5	Ensure 'logoutOnHttpSessionExpire' is set to 'true'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.6	Ensure 'hostNameIncludeList' is set to the host names that will be allowed for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.7	Ensure 'addressIncludeList' is set to the IP addresses that will be allowed for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.8	Ensure 'addressExcludeList' is set to the IP addresses to be excluded for web traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.9	Ensure "trustedSensitiveHeaderOrigin" is set to trusted host names and IP addresses for sensitive data	<input type="checkbox"/>	<input type="checkbox"/>
4.4.10	Ensure 'trustedHeaderOrigin' is set to trusted host names and IP addresses	<input type="checkbox"/>	<input type="checkbox"/>
4.4.11	Ensure 'logoutPageRedirectDomainNames' is set to valid host names to redirect after logout	<input type="checkbox"/>	<input type="checkbox"/>
4.4.12	Ensure security constraints are specified to protect web applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.13	Ensure application security feature is enabled	<input type="checkbox"/>	<input type="checkbox"/>
4.4.14	Ensure 'invalidateOnUnauthorizedSessionRequestException' is set to 'false'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.15	Ensure Web Server Document Root does not contain information that should be private	<input type="checkbox"/>	<input type="checkbox"/>
4.4.16	Ensure HTTP session overflow is 'disabled'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.17	Ensure uncovered http methods are denied	<input type="checkbox"/>	<input type="checkbox"/>
4.4.18	Ensure 'disallowServeServletsByClassName' is 'disabled'	<input type="checkbox"/>	<input type="checkbox"/>
4.4.22	Ensure virtual hosts are defined to isolate applications	<input type="checkbox"/>	<input type="checkbox"/>
4.4.23	Ensure virtual hosts are Defined to isolate JMX communication and application traffic	<input type="checkbox"/>	<input type="checkbox"/>
4.4.24	Ensure whitelisting of virtual hosts to validate access based on originating endpoint	<input type="checkbox"/>	<input type="checkbox"/>

# Appendix: Change History

Date	Version	Changes for this version
May 13, 2022	1.0.0	Document Created