Security Guidelines for Production Deployment

Given below are the common security guidelines for deploying a WSO2 product in a production environment.

Also, see the production deployment guidelines and any other product-specific guidelines that might come in the respective product's documentation.

- WSO2 product-level security
- OS-level security
- Network-level security

WSO2 product-level security

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply security updates</td>
<td>Apply all the security patches relevant to your product version. If your WSO2 product version is supported by WSO2 Update Manager (WUM), you need to use WUM to get the latest fixes.</td>
</tr>
<tr>
<td></td>
<td>• If your WSO2 product is listed as a WUM-supported product here, follow the instructions in Getting Started with WUM.</td>
</tr>
<tr>
<td></td>
<td>• If you are using an older WSO2 product version, which is not WUM-supported, you need to download the security patches relevant to your product from the WSO2 Security Patch Release page and apply them to your system manually. The instructions are given in WSO2 Patch Application Process.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Change default keystores</td>
<td>Change the default key stores and create new keys for all the cryptographic operations. WSO2 products by default come with a self-signed SSL key. Since these keys are public, it is recommended to configure your own keys for security purposes. Consider the following guidelines when creating the keystores:</td>
</tr>
<tr>
<td></td>
<td>• Select a key size of at least 2048 bits.</td>
</tr>
<tr>
<td></td>
<td>• Use an SHA256 certificate.</td>
</tr>
<tr>
<td></td>
<td>• Make sure that WSO2 default certificates do not exist in any of the keystores in your production environment. For example, be sure to delete the default public certificate in the default trust store that is shipped with the product.</td>
</tr>
<tr>
<td></td>
<td>See the recommendations for using keystores in WSO2 products for information.</td>
</tr>
<tr>
<td></td>
<td>See Creating New Keystores for information on how to create and configure your own keys.</td>
</tr>
<tr>
<td>Encrypt passwords in configuration files</td>
<td>WSO2 products use a tool called Secure Vault to encrypt the plain-text passwords in configuration files.</td>
</tr>
<tr>
<td>Change default ports</td>
<td>All the default ports used by WSO2 products are listed in Default Ports of WSO2 Products. For example, the default HTTPS port is 9443 and the HTTP port is 9763. Also, Axis2 services are exposed over the following ports: 8243 and 8280.</td>
</tr>
<tr>
<td></td>
<td>To change a default port, update the &lt;Offset&gt; element in the carbon.xml file as explained in Changing the Default Ports.</td>
</tr>
<tr>
<td>Enable read-only access to external user stores (LDAPs etc.)</td>
<td>If your WSO2 product is connecting to an external user store, such as Microsoft Active Directory, for the purpose of reading and retrieving user information, be sure to enable read-only access to that user store.</td>
</tr>
<tr>
<td></td>
<td>For example, see Configuring a Read-Only LDAP User Store under Configuring User Stores for instructions.</td>
</tr>
<tr>
<td>Always communicate (with external user stores) over TLS</td>
<td>All connections from your WSO2 product to external databases, userstores (LDAP), or other services, should be over TLS, to ensure adequate network-level protection. Therefore, be sure to use external systems (user stores, databases) that are TLS-enabled.</td>
</tr>
<tr>
<td>Connect to data stores using a less privileged user</td>
<td>When connecting the WSO2 product to external databases or user stores (LDAP), be sure to go through a user who does not have permission to change the data store's schema. Be sure not to use the root user of the data store because all permissions are generally granted to the root user.</td>
</tr>
</tbody>
</table>
| Configure strong HTTP(S) security | To have strong transport-level security, use TLS 1.2 and disable SSL, TLS 1.0 and 1.1. The TLS protocol and strong ciphers are configured for an HTTP connector in the catalina-server.xml file (using the sslEnabledProtocols and ciphers attributes). See the following links for instructions:

- Configuring Transport-Level Security
- Supported Cipher Suites

Note the following:

- When deciding on the TLS protocol and the ciphers, consider the compatibility with existing client applications. Imposing maximum security might cause functional problems with client applications.
- Apply ciphers with 256 bits key length if you have applied the Unlimited strength policy. Note that Unlimited strength policy is recommended.
- Also, consider the following factors when deciding on the ciphers:
  - DES/3DES are deprecated and should not be used.
  - MD5 should not be used, due to known collision attacks.
  - RC4 should not be used, due to crypto-analytical attacks.
  - DSS is limited to a small 1024 bit key size.
  - Cipher-suites that do not provide Perfect Forward Secrecy/Forward Secrecy (PFS/FS).
  - GCM based ciphers are recommended over CBC ciphers.

| Remove weak ciphers for PassThrough transport | Applicable only to products that use the PassThrough transport, such as WSO2 API Manager and WSO2 Enterprise Integrator (ESB profile).

Remove any weak ciphers from the PassThrough transport and ensure that the server does not accept connections using those weak ciphers. The PassThrough transport is configured using the axis2.xml file (stored in the <PRODUCT_HOME>/repository/conf/axis2/ directory. You need to add the PreferredCiphers parameter under the "Transport Ins (Listeners)" section along with the list of relevant cipher suites. See Configuring the PassThrough Transport for instructions.

| Update the HTTP Response header "Server" value | By default, all WSO2 products pass "WSO2 Carbon Server" as the server value in HTTP headers when sending HTTP responses. This means that information about the WSO2 product stack will be exposed through HTTP responses. It is recommended to change this by configuring the server name in the catalina-server.xml file. See Configuring Transport Level Security for instructions.

| Enabling HTTP Strict Transport Security Headers (HSTS) | Be sure that HTTP Strict Transport Security (HSTS) is enabled for all the applications deployed in your WSO2 server. This includes the management console, and any other web applications and/or Jaggery applications.

Note that (for products based on Carbon 4.4.11 or later versions) HSTS is disabled for the applications with which the product is shipped by default. This is because HSTS validation can interrupt the development processes by validating signatures of self-signed certificates.

See the topic on Enabling HTTP Strict Transport Security Headers for instructions.

| Preventing browser caching | If there are dynamic pages in your application with sensitive information, you need to prevent browser caching. This can be done by making sure that the applications deployed in your server will return the relevant HTTP response headers.

Note that cache prevention headers are enabled for the applications with which the product is shipped by default. Therefore, you need to manually enable cache prevention headers only for all the new applications that you deploy in your server.

See the topic on Preventing browser caching for instructions.

| Increase Ephemeral Diffie-Hellman Key size | Before starting the server, open the product startup script (wso2server.sh in Linux and wso2server.bat in Windows) and enter the following with the other Java properties:

-Djdk.tls.ephemeralDHKeySize=2048
<table>
<thead>
<tr>
<th>Feature</th>
<th>Instructions</th>
</tr>
</thead>
</table>
| Disable client-initiated renegotiation | Before starting the server, open the product startup script (wso2server.sh in Linux and wso2server.bat in Windows) and enter the following with the other Java properties:  
  ```
  -Djdk.tls.rejectClientInitiatedRenegotiation=true \ 
  ```                                                                                                                                                                                                                                                                 |
| Enable HostName Verification     | If your product is using Carbon Kernel 4.4.17 or a later version, make sure that hostname verification is enabled in the product startup script (wso2server.sh in Linux and wso2server.bat in Windows) with the **Strict** mode. That is, you need to enable the following parameter:  
  ```
  -Dhttpclient.hostnameVerifier=Strict \ 
  ```  
  In Carbon versions prior to 4.4.17, be sure that hostname verification is enabled by setting the following property to `false`:  
  ```
  -Dorg.wso2.ignoreHostnameVerification=false \ 
  ```  
  See [Enabling HostName Verification](#) for instructions.                                                                                                                                                                                                                              |
| Enable additional XSS Protection | XSS attacks are prevented on the latest WSO2 products by default. This is due to output encoding of the displaying values. However, if additional protection is required, an input validation valve can be configured. See [Mitigating Cross Site Scripting Attacks](#) for instructions.                                                                                     |
| Increase JSESSIONID length       | If required, increase the session ID length by changing the `sessionIDLength` attribute of the session manager in the `context.xml` file (stored in the `<PRODUCT_HOME>/repository/conf/tomcat/context.xml` directory) as shown below. The default value is 16 bytes.  
  ```
  <Manager className="org.wso2.carbon.webapp.mgt.
  CarbonTomcatSessionManager" sessionIdLength="16"></Manager>
  ```                                                                                                                                                                                                                     |
| Change default admin credentials | All WSO2 products have the Administrator account configured by default. The default user name and password of the administrator account is "admin". To change the administrator credentials, you need to first sign in to the management console of the product as "admin", and then use the **Change Password** option under **Home->Configure->User Management->Users** in the navigator.  
  See [Changing a Password](#) for more information on how to change the password of the administrator.                                                                                                                       |
| Restrict access to the management console | Make sure that the permission for signing into the management console is granted only to the users that need to use the management console. For example, in products such as WSO2 Identity Server and WSO2 API Manager, the majority of users only need to login to the connected service providers via the WSO2 product. Therefore, such users should not have permission to log into the management console.  
  You can make sure that only administrative users have access to the product’s management console. Further, instead of granting all permission to one administrator, you can distribute the responsibilities among administrators by assigning different permissions for conducting various tasks.  
  See [Managing Users, Roles and Permissions](#) for instructions.                                                                                                                                                 |
Enable log rotation and monitoring

Ensure that you have a relevant log rotation scheme to manage logs. Log4J properties for WSO2 products can be configured in the `<PRODUCT_HOME>/repository/conf/log4j.properties` file. To roll the `wso2carbon.log` based on size, the following configurations can be used.

```
log4j.appender.CARBON_LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.CARBON_LOGFILE=${carbon.home}/repository/logs/${instance.log}/wso2carbon${instance.log}.log
log4j.appender.CARBON_LOGFILE.MaxFileSize=1000KB
log4j.appender.CARBON_LOGFILE.MaxBackupIndex=10
```

See Monitoring Logs for details on how to configure logging details in WSO2 products.

Prevent Log Forging

Log forging can be prevented by appending a UUID to the log message. See Monitoring Logs for more information on configuring the log4j.properties file.

Set appropriate JVM parameters

The recommended JDK version is JDK 1.7 or 1.8. See the installation pre-requisites for more information.

For JDK 1.7, set the appropriate Heap and Permgen values for the JVM based on your deployment scenario. These can be set in `bin/wso2server.sh` file. You do not need to set this in JDK 1.8 as the MaxPermSize value has been removed from Hotspot JVM.

```
-Xms512m -Xmx2048m -XX:MaxPermSize=1024m
```

Tip: To run the JVM with 2 GB (-Xmx2048m), you should ideally have about 4GB of memory on the physical machine.

OS-level security

This section provides the list of OS-level security guidelines for your production environment.

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run WSO2 processes with a specific OS-level user</td>
<td>Use a separate OS-level user to run WSO2 products. Make sure that the user is only granted the permissions required for running the product for that particular user. Do not use the root/administrator user of your OS because the root/administrator is granted all privileges by default.</td>
</tr>
<tr>
<td>Minimize software to avoid vulnerability</td>
<td>Make sure that you only install the software/packages that are relevant to your WSO2 product's deployment. Also, continuously monitor the software that you install. See the Installation Prerequisites to identify the minimum software your WSO2 product will need.</td>
</tr>
<tr>
<td>Enable the Firewall</td>
<td>Enable a firewall at the OS level (for example, <code>iptables</code>). This will provide protection for inbound and outbound connections of your WSO2 product. Make sure that you only open the required outbound and inbound ports from the OS-level firewall.</td>
</tr>
<tr>
<td>Restrict access to TCP ports used for clustering</td>
<td>Apply a firewall at host-level to disallow access to TCP ports used for clustering (port 4000, 4001, … by default) from unrecognized hosts. These ports should be accessible only from other members of the WSO2 product cluster.</td>
</tr>
</tbody>
</table>
| Use Secure Shell(SSH)            | Use Secure Shell (SSH) when interacting with servers and executing commands. Adhere to the following best practices when you configure SSH:  
  - Change the default ssh port to a higher value. 
  - Disable the root or administrator. 
  - Enable login with user keys. 
  - Display a legal banner or a security banner with security warnings before SSH authentication. |
| Keep the system up-to-date        | If there are security updates available for the packages installed in your OS (including the Java runtime), be sure to perform the necessary testing in a staging environment, and then proceed to install them for your OS. |
Monitor user activities

Monitor the activities of your OS users. You can do this by enabling OS-level logs and by reviewing them regularly. You can also set up a centralized log monitoring system for this purpose.

Session Data Cleanup

In a production environment, there is a possibility for a deadlock/database lock to occur when running a session data cleanup task in high load scenarios. To mitigate this, configure the following property to clean data in chunks. Configure this property in the `<IS_HOME>/repository/conf/identity/identity.xml` file under `<SessionDataCleanUp>` with the required chunk size. This value is in the number of records and depends on the database type and server capacity. It also depends on the amount of load generated by single sign-on (SSO). A higher value increases the chances of deadlocks and a lower value increases the time it takes for a cleanup.

```
<DeleteChunkSize>50000</DeleteChunkSize>
```

For more information on configuring sessions in production, see Authentication Session Persistence in the WSO2 Identity Server documentation.

Make regular backups

Make sure to backup important files and archive them continuously.

See Backup and Recovery Recommendations for more information.

### Network-level security

This section provides the list of security guidelines for configuring the network.

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a failover setup</td>
<td>When your WSO2 products are clustered, you need to regularly monitor the health of your server instances. For example, you need to monitor resource-level factors such as the server’s resource utilization, response time anomalies, and the number of incoming network connections. Server monitoring will help you identify when additional server instances (failover instances) are required. You can also make decisions about network routing changes that you need to do in order to avoid server downtime.</td>
</tr>
<tr>
<td></td>
<td>• See Deployment and Clustering/Key Concepts for information on configuring failover.</td>
</tr>
<tr>
<td></td>
<td>• See Monitoring WSO2 products for information on the monitoring options for WSO2 products.</td>
</tr>
<tr>
<td>Maintain network loggin</td>
<td>Be sure to maintain and monitor logs for your proxy servers, load balancers, and other network devices.</td>
</tr>
<tr>
<td>Check open ports and services</td>
<td>Periodically check for open ports using port scanning tools and make sure that only the necessary ports are open to both internal and external networks. Be sure that only the ports relevant to your WSO2 products are open for communication. If there are other ports started, be sure to monitor them. See Default Ports of WSO2 Products for the full list of ports in all WSO2 products.</td>
</tr>
<tr>
<td>Configure device-level security</td>
<td>All your network devices should be periodically checked for anomalies. For example, you need to verify routing tables and firewall rules. Also, make sure that the default credentials are changed before the first use of those devices.</td>
</tr>
<tr>
<td>Apply firmware updates</td>
<td>Firmware updates for your network devices should be applied regularly.</td>
</tr>
</tbody>
</table>
| Block the `/services` and `/carbon` contexts from the DMZ | Access to the `/services` and `/carbon` contexts should be blocked from the DMZ level (i.e., from the proxy server, load balancer and/or firewall).

- The `/services` context is used in WSO2 products to expose admin services. These admin services are used for performing administrative operations using SOAP requests.
- The `/carbon` context is used in WSO2 products to expose the management console (administration console) of the product. The management console is a user interface for performing some of the administrative operations of a product.
- In addition to the `/services` and `/carbon` contexts, be sure to expose only the required applications in your product to users beyond the DMZ level in your network.

It is recommended to use a whitelisting approach when allowing access to resources in your product from the DMZ level.