

# AppSentinels API Security Platform

## WSO2 Integration

## Contents

|  |   |
|--|---|
| Contents.....                                  | 1 |
| Install the JAR File .....                     | 2 |
| Register the Handler in Velocity Template..... | 2 |
| Instructions .....                             | 2 |
| Handler Properties Reference .....             | 3 |
| Configure Logging (Optional).....              | 3 |
| Add Logger Definition .....                    | 3 |
| Append to Loggers List: .....                  | 3 |
| Restart WSO2 API Manager .....                 | 4 |
| Payload Capture Limitations.....               | 4 |
| WSO2 Internal Buffer Limit (16 KB) .....       | 4 |
| Impact .....                                   | 4 |
| Increasing the Buffer Size .....               | 4 |
| Troubleshoot.....                              | 5 |
| Verify Installation.....                       | 5 |
| Common Runtime Issues .....                    | 5 |
| AppSentinels Logger Debug Commands.....        | 5 |
| Upgrade Notes .....                            | 5 |
| Reference.....                                 | 5 |

## Install the JAR File

- Download the AppSentinels WSO2 Logger JAR file from below location:  
**Location:** [Appsentinels WSO2 Logger](#)
- Copy the Downloaded JAR file to the WSO2 API Manager library directory:  
**/opt/wso2am/repository/components/lib/**

## Register the Handler in Velocity Template

- Edit the velocity template configuration file to register the AppSentinels handler with WSO2 API Manager:
  - **<wso2-root>/repository/resources/api\_templates/velocity\_template.xml**

### Instructions

Locate the section in **velocity\_template.xml** where handlers are registered. It will look similar to the following:

```
****  
  
## print the handlers  
#if( $handlers.size() > 0 )  
<handlers xmlns="http://ws.apache.org/ns/synapse">  
****
```

Immediately after this section, add the following handler block:

```
<handler class="org.appsentinels.AppSentinelsLogger">  
<property name="edgeControllerUri" value="http://<controller-host>:<port>/mergedlog"/>  
<property name="instanceName" value="wso2-controller"/>  
<property name="maxPayloadSize" value="16384"/>  
<property name="skipHttpMethods" value="OPTIONS,HEAD"/>  
<property name="logSenderThreads" value="2"/>  
<property name="requestTimeoutMs" value="500"/>  
<property name="connectTimeoutMs" value="1000"/>  
<property name="hostname" value="my-host"/>  
<property name="skipExtensions" value="  
"svg,txt,png,PNG,jpg,jpeg,io,tff,woff,woff2,ico,css,pdf,html,mp4,php,ajax,js,gif,tiff,ttf,docx,sv  
g,json,wasm,aspx"/>
```

```
<property name="collectContentTypes" value="json,xml,graphql,form"/>
</handler>
```

## Handler Properties Reference

| Property            | Default Value       | Description  |
|---------------------|---------------------|--|
| edgeControllerUri   | <required>          | URL of the AppSentinels Edge Controller endpoint                   |
| instanceName        | wso2-custom-handler | Identifier for this WSO2 instance                                  |
| maxPayloadSize      | 16384               | Maximum payload bytes to capture (see Payload Capture Limitations) |
| skipHttpMethods     | OPTIONS,HEAD        | HTTP methods to exclude from logging                               |
| logSenderThreads    | 2                   | Number of async threads for sending logs                           |
| requestTimeoutMs    | 500                 | HTTP request timeout in milliseconds                               |
| connectTimeoutMs    | 1000                | HTTP connection timeout in milliseconds                            |
| hostname            | my-host             | Hostname reported in log entries                                   |
| skipExtensions      | css,js,png          | File extensions to exclude from logging                            |
| collectContentTypes | json,xml            | Content types whose payloads are captured                          |

**Note:** Adjust all property values to match your environment before deploying.

## Configure Logging (Optional)

Enable debug logging for the AppSentinels handler by modifying the Log4j2 configuration

### Add Logger Definition

- Append the following two lines at the end of the file `/opt/wso2am/repository/conf/log4j2.properties`
  - `logger.appsentinels_logger.name = org.appsentinels.AppSentinelsLogger`
  - `logger.appsentinels_logger.level = ERROR`

### Append to Loggers List:

- Find the loggers property in the same `log4j2.properties` file and append `appsentinels_logger` to the comma-separated list:

**loggers = AUDIT\_LOG, trace-messages, ..., appsentinels\_logger**

**Note: Do not remove any existing loggers. Simply append appsentinels\_logger at the end of the existing list.**

## Restart WSO2 API Manager

After completing all configuration changes, restart the WSO2 API Manager to apply them:

- `cd <wso2-root>/bin`
- `./api-manager.sh restart`

After the restart of api-manager the **AppSentinels Logger Handler** will be activated and API should be visible on dashboard which are going via this WSO2

## Payload Capture Limitations

### WSO2 Internal Buffer Limit (16 KB)

- The handler captures payloads using WSO2's internal pass-through transport buffer (ControlledByteBuffer). This buffer has a fixed capacity of 16,384 bytes (16 KB) in most WSO2 implementations.

### Impact

- Upstream traffic is always preserved and unmodified (no data corruption).
- We discard the payload capture if the payload size is more than WSO2 internal Buffer size

### Increasing the Buffer Size

To capture larger payloads, edit the WSO2 deployment configuration:

- Open `/opt/wso2am/repository/conf/deployment.toml`

Add or update the following transport buffer settings:

```
[transport.http]
```

```
io_buffer_size = 32768]
```

```
[transport.https]
```

```
io_buffer_size = 32768
```

**Note:** The logged payload size will always be the lower of `maxPayloadSize` (handler property) and `io_buffer_size` (transport config). Restart WSO2 after making changes.

## Troubleshoot

### Verify Installation

Confirm the JAR file is present in the library directory:

- `ls -la /opt/wso2am/repository/components/lib/`

Monitor WSO2 logs in real time:

- `tail -f /opt/wso2am/repository/logs/wso2carbon.log`

### Common Runtime Issues

| Issue                      | Resolution  |
|----------------------------|---|
| Handler not loading        | Check WSO2 logs for class loading errors.   |
| Performance degradation    | Monitor API latency after deployment. Adjust <code>logSenderThreads</code> and <code>timeout</code> values. |
| Old handler version loaded | Clean both <code>lib/</code> and <code>dropins/</code> directories before deploying a new JAR.              |

### AppSentinels Logger Debug Commands

Use the following commands to inspect handler-specific log output:

- Check AppSentinels debug messages:
  - `tail -f /opt/wso2am/repository/logs/wso2carbon.log | grep AppSentinelsLogger`
- Check payload extraction messages:
  - `tail -f /opt/wso2am/repository/logs/wso2carbon.log | grep "Extracted.*payload"`
- Check async log sender messages:
  - `tail -f /opt/wso2am/repository/logs/wso2carbon.log | grep "AsyncLogSender"`

### Upgrade Notes

**Note:** WSO2 caches and copies JARs into the **dropins/** directory and may load old versions. Always clean up both **lib/** and **dropins/** before deploying a new handler JAR.

## Reference

| Item              | Path / Value  |
|-------------------|---|
| JAR location      | <code>/opt/wso2am/repository/components/lib/</code>                                     |
| Velocity template | <code>&lt;wso2-root&gt;/repository/resources/api_templates/velocity_template.xml</code> |
| Log4j2 config     | <code>/opt/wso2am/repository/conf/log4j2.properties</code>                              |

|                   |   |
|-------------------|---|
| Deployment config | /opt/wso2am/repository/conf/deployment.toml |
| WSO2 logs         | /opt/wso2am/repository/logs/wso2carbon.log  |
| Restart command   | <wso2-root>/bin/api-manager.sh restart      |