ANGLER

File System remote sensor

User manual

2019-03-03

V1.0

yourAngler appliance exposes an API that can be used to extend the monitoring service to additional computers within your enterprise. This manual describes the installation and configuration of a software remote sensor that monitors a number of file system locations for signs of modification. Once a create, delete or modification event is detected, the sensor will connect to the configured yourAngler appliance and send the event details, to be distributed via all configured channels (cloud console, SMS, Email, Syslog).

Installation

Perquisites

The application is written in Java using Spring Boot framework. To run it, the computer must have a Java virtual machine (JVM), version 1.7 (7) or 1.8 (8), installed and available on the PATH. If you do not have one installed, the following links will provide you with the necessary java installation packages. Please ensure that you select one download matching your OS and CPU architecture:

https://www.java.com/en/download/
https://openjdk.java.net/install/

Additional java virtual machine implementations can be found here:

https://en.wikipedia.org/wiki/List_of_Java_virtual_machines

Once the JVM is installed, you can verify if its availability by running the following command in a shell window:

```
java -version
```

An output similar to the following should be expected:

```
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

Please correct any errors that you might encounter before you continue the installation. The above links contain sections that will assist you in diagnosing and fixing the error conditions.

Installation package structure

You must expand the downloaded package in a folder of your choice. The following folders will be created

- bin - contains start.sh or start.bat shell scripts that will be used to start the application
- lib - all application dependency libraries are stored in this folder.
- logs - application logs will be generated in this location.
- config - application configuration files that need to be adjusted prior to starting the application.
    - application.properties – this is the main configuration file that controls the locations to be monitored, and how to send the detected events to the yourAngler appliance.
    - logback.xml – this configuration file specifies the logging parameters and the location of the file log. The default output location is in the 'logs' folder. This configuration file is not usually modified, but if you need to adjust it, please consult the documentation of the 'logback' framework:
      https://logback.qos.ch/documentation.html

application.properties

This configuration file follows the java properties file format described at the end of this section. It contains three main sections

- Connectivity

    - net.protocol=UDP or TCP. It controls the socket type to be used when connecting to yourAngler appliance. Must match the configured value in the API section.
    - net.port=1234. Integer, representing the port that was configured in the API section of the appliance.
    - net.ip=ddd.ddd.ddd.ddd – IP address of yourAngler appliance. This IP can be found in the 'Info' section of the web console of the appliance or displayed as 'Ethernet IP' on the OLED display of the appliance.
- Authentication
    - auth.key=xxxxxx – this is a key that must match the value configured in the appliance API section. If the keys do not match, the events will be silently discarded.
    - auth.name=yyyyy – a string value that will be copied into every event sent to the appliance using the key 'name'.
- Monitored locations configuration
    - fs.locations=/path1,/path2 – comma separated list of absolute path locations pointing to the folders to be monitored for create/modify/delete activities. To minimize the application performance impact, please configure folders with less than 1000 child files (Example: do not add the root of the filesystem to the list)

      Important: *on Windows, the backslash character need to be escaped as \\. For example, the following pats are equivalent:*
      C:\\tmp\\sampleFile.txt
      C:/tmp/sampleFile.txt
    - fs.createLocations=true or false – optional field, with the default value of 'true' if not specified. When the application is first executed, it will create all specified locations if missing.
    - fs.ignoreDirModifications=true or false - optional field, with the default value of 'true' if not specified. If set to true, the events associated with the modification of the directories (not contents) will be discarded.

Application.properties file format:

- Entries are generally expected to be a single line of the form, one of the following:

- *propertyName=propertyValue*
- *propertyName:propertyValue*

• White space that appears between the property name and property value is ignored, so the following are equivalent.

```
name=Stephen
name = Stephen
```

White space at the beginning of the line is also ignored.

• Lines that start with the comment characters ! or # are ignored. Blank lines are also ignored.

• The property value is generally terminated by the end of the line. White space following the property value is not ignored, and is treated as part of the property value.

• A property value can span several lines if each line is terminated by a backslash (‘\’) character. For example:

```
targetCities=\
    Detroit,\
    Chicago,\
    Los Angeles
```

This is equivalent to targetCities=Detroit,Chicago,Los Angeles (white space at the beginning of lines is ignored).

• The characters newline, carriage return, and tab can be inserted with characters \n, \r, and \t, respectively.

• The backslash character must be escaped as a double backslash. For example:

```
path=c:\\docs\\doc1
```

• UNICODE characters can be entered as they are in a Java program, using the \u prefix. For example, \u002c.

Remote - API

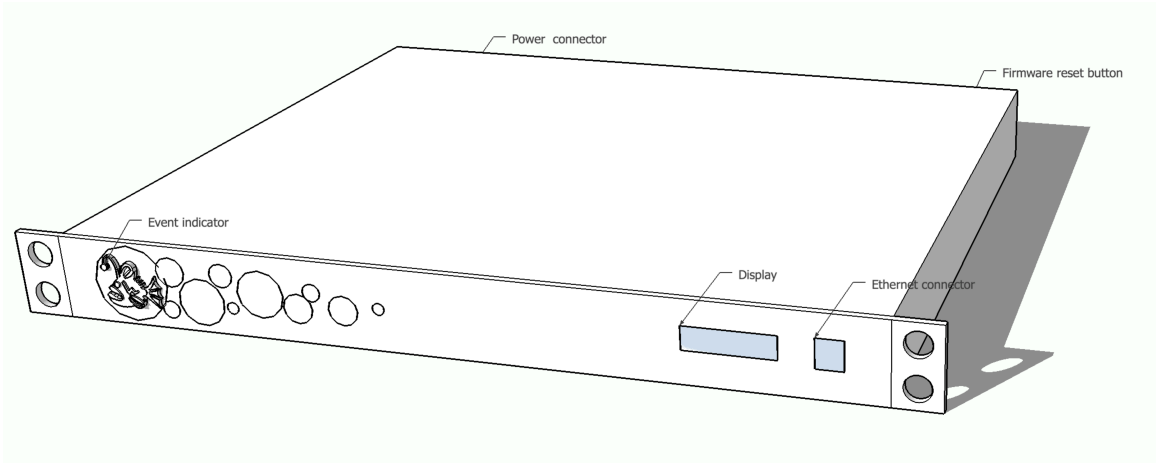
Deploying remote sensors on additional servers or desktops within your enterprise will extend the appliance's monitoring area

The API uses a TCP or UDP listener, on a port that defaults to 1234. It is recommended that you reconfigure the API listener to use a different IP port. The sender process must send the data using JSON encoding using a TCP or UDP connection to the above port. For details regarding JSON format please see the following links:

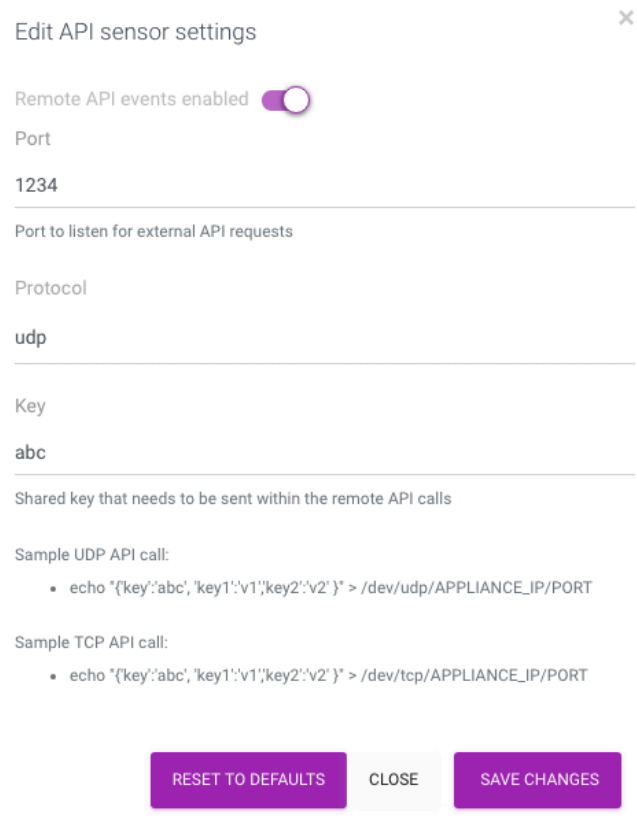
https://www.json.org/
https://tools.ietf.org/html/rfc7159

To process an inbound event, the received data must contain a key called 'key' with a value identical to the one configured in the appliance's API configuration page. If these values are not matching, the appliance will silently discard the event. All remaining keys and source IP related information will be collected and used to generate a notification message distributed via the configured notification channels.

The appliance's IP address can be found in the 'info' section of the web console or displayed on the built in OLED display (Ethernet IP).



The API function can be enabled or disabled by toggling ‘Remote API events’ button.
On the configuration screen, clicking ‘Reset to defaults’ button, port will be reset to 1234, the protocol will be set to UDP and the key will be regenerated.



Technical support

Than you for your purchase. If you require assistance setting up the software, please do not hesitate to contact us via email support@yourangler.com. For quick answers, please visit htto://faq.yourangler.com