Events Pane in the Node View - enabling websocket when using SSL or TLS

This page discusses how to enable the websocket when using SSL/TLS only on your servers. The websocket is used to display a continuously updating list of the node's events when viewing a node context page in opCharts.

If you do not need SSL enabled for your events stream in opCharts, then the default websocket settings will simply use an un-encrypted websocket to the server on port 8042. See notes below about apache forcing WSS (SSL) when HTTPS is used .

To confirm your websocket is working look out for the following in the events pane. Green shows it is working or Yellow means the websocket has failed.

There are two parts to making this work

1. having the client connect on wss: port 443
2. having the proxy forward the events URL to the websocket for OMK.

Request the clients browser to use 443 for the websocket

To tell the clients browser to use 443 (and hence wss: with SSL/TLS) instead of the default port 8042 (ws:) change the following in opCommon.nmis.

opCommon.nmis set websocket port

```
  'websocket_proxy_port' => '443'

or in opCommon.json:
  "websocket_proxy_port": "443",
```

After updating the opCommon configuration, restart the omkd daemon.

Proxy the connection from your SSL/TLS termination to the OMK webservice

The client's websocket will now be coming through the same SSL/TLS Transport as your HTTPS traffic on port 443. We need the proxy server to forward the websocket to the OMK webservice much like the HTTP traffic.

This configuration will vary based on how you are handling your SSL/TLS termination. As such it can only be an example.

For the purposes of this article we assume the following:

1. You are using apache webproxy with no additional proxies / WAFs in front of the apache server.
2. You are using Apache's ssl module to terminate your SSL/TLS transport
3. Apache is already proxying your HTTPS traffic to the OMK webservice as HTTP on port 8042
4. You are using a similar apache configuration style (there are other methods of achieving the same)
5. "proxy_wstunnel_module" is already loaded (see information panel with example)

We need to find you apache configuration which is handling port 443 connections and hence the SSL engine. Within that VirtualHost we need to request that URL **"/en/omk/opCharts/events/log"** be proxied to **"ws://localhost:8042/en/omk/opCharts/events/log"** note the ws:

In our example this looks as follows:

Apache .conf file snippet

```
<VirtualHost *:443>
SSLEngine on
SSLProxyEngine on
ServerName some.example.com

ProxyPass "/en/omk/opCharts/events/log" "ws://localhost:8042/en/omk/opCharts/events/log"


ProxyRequests off
RequestHeader set X-Forwarded-Proto "https"
```

After updating the Apache configuration, restart the Apache daemon.

Additional Apache Configuration items and their impact

Proxy WS Tunnel module
The Apache WS proxy tunnel Module needs to be enabled for ProxyPass to work for ws.
On most systems this can be found as "proxy_wstunnel.load" you will probably find this file in mods-available and simply need to link it mods-enabled. Your Apache setup might use other configuration methods to load modules so please do check.

WSS HTTPS interaction

Please note apache will force the use of WSS (SSL websocket) if you are using HTTPS. This is because the apache configuration:

```
RequestHeader set X-Forwarded-Proto "https"
As the X-Forwarded-Proto HTTP header is seen by the OMK webservice, it therefore sets the connection URLs to be wss not ws.
```

See Also

- opEvents Realtime Events