Setting up Secure WebSockets (wss) requires a **Reverse Proxy** configuration. This allows your web server (Nginx or Apache) to handle the SSL/TLS encryption on port **443** and pass the raw traffic to your PHP WebSocket server running on port **8080**.

1. Nginx Configuration

Nginx is the most popular choice for WebSockets because of its native support for "Upgrade" headers.

Add this to your Nginx site configuration (usually in `/etc/nginx/sites-available/default`):

```
server {
    listen 443 ssl;
    listen [::]:443 ssl; # Added IPv6 support
    server_name yourdomain.com;

    # SSL Certificates
    ssl_certificate /etc/letsencrypt/live/yourdomain.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/yourdomain.com/privkey.pem;

    # Modern SSL Security (Recommended)
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1d;

    location /ws {
        # 1. Reverse Proxy to your PHP WSSocket
        proxy_pass http://127.0.0.1:8080;
        proxy_http_version 1.1;

        # 2. WebSocket Upgrade Headers
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";

        # 3. Real IP Forwarding (Critical for sys_auditlogs)
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        # 4. Buffering and Timeouts
        proxy_buffering off; # Recommended for WebSockets
        proxy_read_timeout 86400s; # Prevents Nginx from dropping idle connections
        proxy_send_timeout 86400s;

        # 5. Handle potential large payloads (like history requests)
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;

        # Optional: Disable logs for pings to save disk space
        # access_log off;
    }
}
```

2. Apache Configuration

For Apache, you must enable the `proxy_wstunnel` module.

Enable modules via terminal:

```
sudo a2enmod proxy
sudo a2enmod proxy_wstunnel
sudo a2enmod remoteip
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Update your VirtualHost file:

```
<VirtualHost *:443>
    ServerName yourdomain.com

    SSLEngine on
    SSLCertificateFile /etc/letsencrypt/live/yourdomain.com/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/yourdomain.com/privkey.pem

    # 1. Real IP Configuration
    # This ensures Apache handles the headers for sys_auditlogs
    RemoteIPHeader X-Forwarded-For
    ProxyPreserveHost On

    # 2. WebSocket Proxy Logic
    # We use RewriteEngine to handle the 'Upgrade' hop properly
    RewriteEngine on
    RewriteCond %{HTTP:Upgrade} websocket [NC]
    RewriteCond %{HTTP:Connection} upgrade [NC]
    RewriteRule ^/ws/?(.*) "ws://127.0.0.1:8080/$1" [P,L]

    # 3. Fallback ProxyPass
    ProxyPass "/ws" "ws://127.0.0.1:8080/"
    ProxyPassReverse "/ws" "ws://127.0.0.1:8080/"

    # 4. Timeout Settings
    # Prevents disconnection during long idle chat times
    ProxyTimeout 86400
</VirtualHost>
```

3. Update your `WSClient` JavaScript

Once the proxy is set up, you no longer connect directly to port `8080`. You connect to the standard HTTPS port using the `/ws` path we defined.

Update your `index.html` initialization:

```
// Connect via WSS (Secure) on the standard port 443
const ws = new WSClient('wss://yourdomain.com/ws', 'pwo_token');
ws.connect();
```