HTML5 Web XPanel

Security Reference Guide

Crestron Electronics, Inc.

Original Instructions
The U.S. English version of this document is the original instructions.
All other languages are a translation of the original instructions.

Revision History

Rev	Date	Notes	Author(s)
A	December 21, 2021	Initial version	IH, RS
B	October 6, 2023	Removed mention of software mobility license requirements	IH

Please send comments and change recommendations to:

SecurityDocs@crestron.com

Contents

Overview

This document covers the system configuration used to securely configure the HTML5 Web XPanel functionality on a Crestron 3-Series® or 4-Series™ control system.

> **NOTE:** The term "control system" is used in this document to refer to all applicable 3-Series and 4-Series control system models unless specified otherwise.

Ports and Protocols

The following ports and protocols may be used by the HTML5 Web XPanel functionality depending on the system design and configuration.

Common Ports

Function	Destination Port	From (Sender)	To (Listener)	Notes
HTTPS	443/TCP	Admin or End User Workstation	Control system	Retrieves the HTML5 Web XPanel project.
WSS (Secure WebSocket)	49200	HTML5 Web XPanel on web browser	Control system	This is the default port for this function. It can be changed via the `securewebsocketport` command.

Prerequisites

In order to perform the configuration, the following prerequisites must be met.

Firmware Version

4-Series control systems must be running firmware version 2.500.*x* or greater. 3-Series control systems must be running firmware version 1.7000.*x* or greater.

Device Access

The administrator can access and configure the control system by using a web browser or an SSH client. This document describes configuration of the device using an SSH client, which provides access to console commands. Some configuration capabilities can only be performed by issuing console commands.

> **NOTE:** The SSH client that is used must be capable of connecting to the device using SSHv2 and must be compatible with FIPS 140-2 validated algorithms.

As an alternative to using an SSH client, the same console commands can be executed through the USB port.

Required Configuration

The following sections describe the configuration changes required for running an HTML5 Web XPanel project on a control system.

> **NOTE:** Ensure that the control system is running the minimum required firmware version or higher as described in Prerequisites on page 3.

Secure Web Port

HTML5 XPanel requires the use of the HTTPS protocol on the control system. The service port for the HTTPS protocol can be changed from the web standard (443) to another port number using the `securewebport` command.

Syntax: `securewebport [portnumber]`

- `portnumber` - Sets the desired port number (in decimal notation)
- No parameter - Displays the current value

You must append the port to the URL to reference the project on the control system if the port was changed from the default. For example, if the port was changed to 8443, the following example URL would change from "https://W.X.Y.Z/shell-template/index.html" to "https://W.X.Y.Z:8443/shell-template/index.html".

Secure WebSocket Port

HTML5 Web XPanel is hosted in a web browser and, as a result, cannot communicate to the control system via TLS sockets like touch screens. A new service port is made available on the control system to support secure WebSocket communications that are supported in browsers and by the CH5 WebXPanel library. The `securewebsocketport` console command can change the default port number.

Syntax: `securewebsocketport [portnumber]`

- `portnumber` - Sets the desired port number (in decimal notation)
- No parameter - Displays the current value

You will need to change the WebXPanel library configuration if you change the WebSocket port from the default value. For more information, refer to Add or Remove HTML5 Web XPanel Support and Add XPanel to Custom (Non-Template) Project on the Crestron HTML5 User Interface Developer Microsite.

Authentication Commands

Crestron has set high security standards for using the HTML5 Web XPanel feature on a control system.

The control system must require authentication to access resources and must enable secure, encrypted communications. The three commands to control these settings are `ssl`, `authentication`, and optionally `userpageauth`.

> **NOTE:** The syntax for each command is provided at the end of this section.

For 4-Series control systems and 3-Series control systems that have been recently purchased or restored to factory defaults, the `ssl` and `authentication` commands will be properly configured to use the HTML5 Web XPanel feature. For older 3-Series control systems, these settings may need to be updated.

> **NOTE:** Issue the `ver -v` command to view information about your control system. If the `FORCED_AUTH_MODE` setting is false, then you will need to update the SSL and authentication settings for the control system.

For systems that have `FORCED_AUTH_MODE` set to true, the `userpageauth` command regulates whether web page resources require credentials to be accessed.

- For 3-Series control systems that have the setting available because `FORCED_AUTH_MODE` is true, `userpageauth` must be set to `on`.
- For 4-Series control systems, it is recommended to set `userpageauth` to `on` to improve the experience for the end user. The user will be prompted for credentials as they load the project initially instead of when the project's WebXPanel library tries to access the control system via the WebSocket.

The following table summarizes the different authentication permutations.

Control System Type	Forced Auth Mode Setting	SSL Setting	Authentication Setting	User Page Auth Setting
3-Series	false	Must be set to self or CA	Must be set to on	N/A
3-Series	true	Always self or CA	Always on	Must be set to on.
4-Series	Always true	Always self or CA	Always on	Recommended set to on.

SSL

Syntax: `ssl [OFF|SELF|CA] {TLS1.2ONLY}`

- `OFF` - Turns SSL off for the control system
- `SELF` - Sets SSL to use self-signed certificates
- `CA` - Sets SSL to use CA (Certificate Authorities) issued certificates
    - `-P` - Used to supply a password for opening the private key file when SSL is set to `CA`

- `TLS1.2ONLY` - Sets TLS 1.2 support exclusively for client and server connections
- No parameter - Displays the current value

Authentication

Syntax: `authentication [ON|OFF]`

- `ON` - Turns authentication on for the control system
- `OFF` - Turns authentication off for the control system
- No parameter - Displays the current value

User Page Authentication

Syntax: `userpageauth [ON|OFF]`

- `ON` - Turns user page authentication on for the control system
- `OFF` - Turns user page authentication off for the control system
- No parameter - Displays the current value

Default Web Endpoint

When a control system is received from the factory or restored to factory defaults, entering the control system hostname or IP address into a web browser with an "https://" prefix will redirect you to the "/setup" endpoint to allow for device configuration.

If the **Web Pages and Mobility Projects** function in Crestron Toolbox™ software is used at least once to deploy an HTML5 Web XPanel project, the default endpoint will change to the last deployed HTML5 Web XPanel project each time a project is deployed.

However, if the **ch5-cli deploy** utility is always used to deploy the project, the default will not change from the "/setup" endpoint. To set the last deployed project to become the default without using the **Web Pages and Mobility Projects** function, use the `webinit` console command. This command must only be sent to the control system once to take effect.

Allow Shared Session for Web Server

4-Series control systems use session management and session cookies that are sent via a web browser to keep track of a given user's login status. If you host an HTML5 Web XPanel project on an independent web server instead of the one provided by the control system, or if you are using web development tools to host a web server on your workstation during project development, the web browser cannot access the 4 Series control system CIP protocol by default.

For maximum security, the session cookies provided by a 4-Series control system are accessible only by web pages served up by the control system web server and are not accessible by web pages served up by independent web servers. The control system, however, provides controls to turn on a shared session with an independent web server to bypass this restriction.

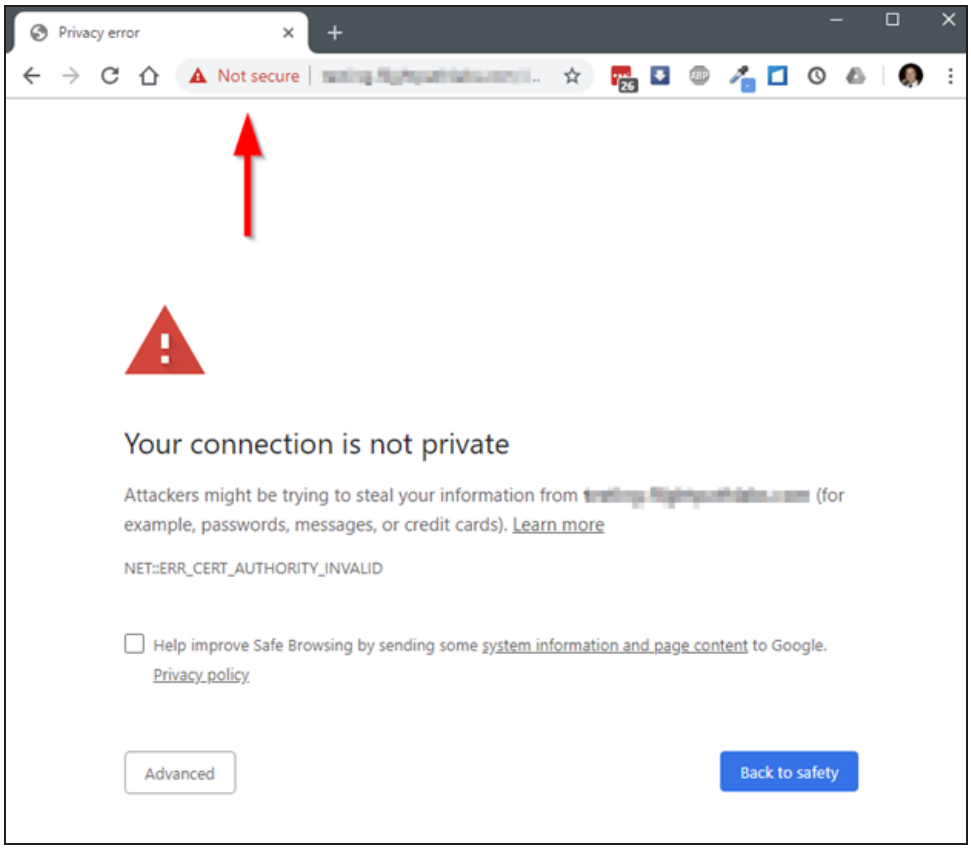
Therefore, if you plan to deploy your HTML5 Web XPanel project on an independent web server, or if you plan to develop your project using a web server hosted on your workstation, you must manually turn on a shared session by issuing the `webserver allowsharedsession on` command.

Issue `webserver allowsharedsession` without a parameter to view the current value for this setting.

Self-Signed Certificate Warnings

All browsers supported by HTML5 Web XPanel will provide warning messages upon connecting to the browser if the control system has a self-signed certificate.

An example from the Chrome® browser is shown below:



This behavior is expected. There are three solutions that can be used to address this.

- Use a CA-signed certificate on your control system (preferred method).
- Your IT Administrator extracts the self-signed certificate and installs onto your workstation as a trusted certificate.
- The end user accepts the self-signed certificate for that browser on their workstation to prevent the message from showing again.

Additional Instructions

The instructions in this section are not specific to the HTML5 Web XPanel. However, they may be useful to an administrator when setting up and configuring the HTML5 Web XPanel on a control system.

Use OpenSSL to Create a Certificate Signing Request (CSR)

In most cases, a CSR must be provided to a certificate signing authority to receive a signed certificate. When requesting a signed certificate for the control system, you may not want to or be able to generate the CSR on the device itself. In these cases, OpenSSL may be used to create the CSR.

This process can be accomplished by following these instructions on any Windows® or Linux® OS-based computer with OpenSSL version 1.0.2 or newer installed.

> **NOTE:** In the following instructions, the example file names include a generic *name* descriptor. It is recommended to replace *name* with a string that identifies the control system that will receive the requested certificate so you can more easily match the certificate files with the appropriate control system.

Create a Configuration File

First, a configuration file that will be used to generate the CSR must be created. This file will contain information about the CSR and any information that should be included in the CSR.

Create a text file called *name*-csr-openssl.cnf with the following contents:

```
# OpenSSL configuration file for CSR generation

# CSR configuration - Change sha256 to alternate hash function if desired
[ req ]
default_md          = sha256
distinguished_name  = req_distinguished_name
string_mask         = utf8only
utf8                = yes
prompt              = no
req_extensions      = req_ext

# Extensions to be included - Currently SAN only
[req_ext]
subjectAltName = @alt_names

# Information to put in certificate Subject field - fill in desired values
# Comment out any items not desired (only commonName is required)
[ req_distinguished_name ]
commonName                      = Device.Fully.Qualified.Domain.Name
```

```
countryName                     = optional
stateOrProvinceName             = optional
localityName                    = optional
0.organizationName              = optional
organizationalUnitName          = optional
emailAddress                    = optional

# List of information to put in SAN extension - fill in desired values
# Additional names or IP addresses can be added if necessary
[ alt_names ]
DNS.1 = Device.Fully.Qualified.Domain.Name
```

Modify the text file to include the information specific to the device and the network site. This information will be put into the Subject field of the certificate and is specified in the `[ req_distinguished_name ]` section of the text file. The `commonName` entry must be filled in and should be the FQDN of the control system.

All other fields are optional and should be filled in or commented out (if not commented out, the certificate will contain "optional" as the value of that field). Note that the `countryName` field is only allowed to be 2 characters.

The following example shows a sample of this section containing filled and empty fields:

```
[ req_distinguished_name ]
commonName                      = deviceName.crestron.com
countryName                     = US
stateOrProvinceName             = NJ
localityName                    = Rockleigh
0.organizationName              = Crestron Electronics
#organizationalUnitName          = optional
#emailAddress                    = optional
```

This CSR will also request the standard Subject Alternate Name (SAN) extension to be included in the certificate. The information to include in this extension is specified in the `[ alt_names ]` section of the text file. At least one entry is required, and that entry should match the FQDN specified in the `commonName` field above.

Add additional names that may be used when connecting to the control system. Each additional name must use an incremented number in the suffix for the "DNS" identifier. IP addresses are also supported if needed.

The following example shows a sample of this section filled out for a control system with three names and two IP addresses:

```
[ alt_names ]
DNS.1 = deviceName.crestron.com
DNS.2 = alternateName.crestron.com
DNS.3 = thirdname.crestron.com
IP.1 = 192.168.0.10
IP.2 = 10.0.0.5
```

Finally, if your certificate signing authority requires the CSR to be signed with a stronger hash than SHA256, the `default_md` field in the `[ req ]` section can be changed. Change `sha256` to `sha384` or `sha512` as needed.

Generate the Private Key

Generate a 2048 bit RSA key by issuing the following command:

```
openssl genrsa -out name.key.pem 2048
```

If desired, replace the `2048` parameter with `3092` or `4096` to generate a longer key of that length.

Create the CSR

Create the CSR using the key and information in the configuration file:

```
openssl req -config name-csr-openssl.cnf -key name.key.pem -new -out name.csr.pem
```

If you wish to view the CSR in text form to confirm it contained the expected information, use the following command:

```
openssl req -noout -text -in name.csr.pem
```

Create and Sign the Certificate

The certificate must be created and signed by the trusted signing authority for the network the device will be used on. Provide the CSR file (name.csr.pem) to your signing authority to create and sign the certificate. The signing authority should return the signed certificate along with the signing chain for that certificate.

Load the Certificate

To load the certificate as the Default Server Certificate, use the *name*.key.pem file that was created, along with the server certificate and signing chain from the signing authority, and follow the instructions provided in the "Certificate Management" and "Default Server Certificate" topics within the 4-Series Control Systems Security Reference Guide.

If you wish to load the certificate as the Web Server certificate, the certificate and key must be placed into a PKCS #12 file. Ensure that the certificate provided by the signing authority is in PEM format, and then issue the following command, where *name*.cert.pem is the file from the signing authority with the certificate in PEM format:

```
openssl pkcs12 -export -out name.certandkey.pfx -inkey name.key.pem -in
name.cert.pem
```

OpenSSL will ask for an "Export Password". Enter a password which will be used to protect the PKCS #12 file. It will then ask you to confirm that password.

Next, follow the instructions provided in the "Certificate Management" topic within the 4-Series Control Systems Security Reference Guide for loading a Web Server certificate. Make sure to provide the Export Password that was entered above when loading the certificate file into the device.

Clean Up

Once successfully loaded onto the control system, wipe the local copy of the private key (in the file *name*.key.pem) on the computer used to generate the CSR, as this contains the secret information specific to that certificate for that device.

Crestron Electronics, Inc.
15 Volvo Drive, Rockleigh, NJ 07647
Tel: 888.CRESTRON
Fax: 201.767.7656
www.crestron.com

Security Reference Guide — Doc. 9215B
10/06/23
Specifications subject to
change without notice.