RBS-2016-003

Crestron DM-TXRX-100-STR
Web Interface Multiple Vulnerabilities

Table of Contents

Vendor / Product Information

Crestron Electronics is a US-based company with offices in every major market and country around the globe. The company manufactures automation and control solutions for buildings and homes, integrating systems such as A/V, lighting, shading, IT, security, BMS, and HVAC. They deliver solutions to universities, the hospitality and entertainment industry including casinos, and corporations in general.

A small subset of customers[1] include the Senate of Virginia, Twitter, Microsoft, The World Bank, American Water Corporate Headquarters, Johnson & Johnson, NYPD Emergency Operations Center, and the Chicago Police Department. Furthermore, "every branch of the U.S. Armed Services, and scores of government agencies around the world, rely on Crestron enterprise solutions to control and manage the highest levels of AV network security"[2].

The DM-TXRX-100-STR device[3] is an H.264 streaming transmitter/receiver that allows distribution of high-definition AV signals over an IP network. The device is used to stream from any media source e.g. IP cameras to various systems and devices like digital signage displays or projectors. The product received The Sound & Video Contractor Best of Show Awards at InfoComm 2015.

Vulnerable Program Details

Details for tested products and versions:

Vendor:	Crestron
Product:	DM-TXRX-100-STR
Version:	1.2834.00024

NOTE: Other versions are likely also affected.

Credits

Carsten Eiram, Risk Based Security
Twitter: @CarstenEiram
Twitter: @RiskBased

[1] https://www.crestron.com/about/press-news/case-studies-installations
[2] http://www.crestron.com/solutions/market/government-eoc-noc-ccc-courtroom-automation
[3] http://www.crestron.com/products/model/dm-txrx-100-str

Impact

DM-TXRX-100-STR provides a web-based management interface for monitoring and configuring it. Access to this interface is password-protected, but has major flaws in the authentication process.

Password verification is performed client-side using JavaScript and is only performed for the main index.html page; other pages can be accessed directly without authentication. Furthermore, while the provided JSON API does support user authentication, the status is not actually checked when using its functionality. These vulnerabilities trivially allow a remote attacker to bypass authentication and gain administrative access to the device i.e. monitor and configure it as well as view the stream.

Other discovered issues included the use of a known, unsafe, hard-coded X.509 certificate for securing HTTPS traffic, Cross-Site Request Forgery (CSRF) for all web pages, and use of default admin credentials that were not easily changeable.

Vulnerability Details

Client-side Authentication Remote Bypass (CVE-2016-5666)

Password authentication to the web-based management interface is implemented client-side using JavaScript. When accessing index.html, the following code runs from /js/device/authenable.js:

```
$(document).ready( function ()
{
        try
        {
                // while(true)
                // {
                // }
            isautenticationenable();


        }
        catch(ex)
        {
          logErrorToConsole (ex);
        }
}
);
```

The `isautenticationenable()` method determines if authentication is enabled by sending a JSON request to the device:

```
function isautenticationenable()
{
        try
        {
                $.ajax({
                                type: "GET",
                                dataType: "json",
                                url: "/uri/auth/settings?isAuthEnabled",
                                success: function (objResponse)
                                {
                                        logResponseToConsole(objResponse);
                                        dispalyloginpage(objResponse);
                                }
                                ,error: function (ex)
                                {
                                        //logErrorToConsole (ex);
                                        handleResponseMessage(ex,"","");
                                }
                        });
        }
        catch(ex)
        {
                logErrorToConsole (ex);
        }
}
```

The response is supplied as argument to the `dispalyloginpage()` [sic] method that checks if authentication is enabled and based on that displays the login or status page:

```
function dispalyloginpage(objresp)
{

        if(objresp.authenabled == '1')
        {
                //window.location="/index.html";
                console.log("authentication enabled");
        }
        else
                window.location="/status.html";
}
```

By intercepting the server response, an attacker can easily have the client-side code consider authentication as being disabled, granting administrative access to the web interface.

Direct Requests Remote Authentication Bypass (CVE-2016-5667)

If intercepting server responses is too much effort in order to bypass the client-side authentication, there is a much simpler approach.

Reviewing the source code of web pages served by the device, it became apparent that only `/index.html` loads the previously described client-side authentication code in `/js/device/authenable.js`. All other pages can be accessed directly and do not check the user's authentication status.

A remote attacker can, therefore, trivially bypass the authentication by e.g. just accessing the `/status.html` page directly.

JSON API Remote Authentication Bypass (CVE-2016-5668)

The web-based management interface also provides a JSON API. This is used to perform the actions taken by administrative users via the web pages including e.g. checking authentication status as previously described by calling `/uri/auth/settings?isAuthEnabled`. However, while authentication is provided by the JSON API via the `/uri/auth/settings` endpoint, calls to other API methods never actually check if authentication is enabled or the authentication status of the user. Using direct requests to these features provided by the JSON API, a remote attacker can completely bypass authentication and access any administrative functionality.

Analyzing the JSON API, we managed to identify the following supported API methods:

API Method	Description
/uri/debug/debuginfo	Display all debug info about system
/uri/edid/edid	Display all edid information
/uri/edid/edid?edidlist	List edid configuration
/uri/edid/edid?edidadd	Upload defined edid file
/uri/edid/edid?applyedid	Apply specified edid
/uri/edid/edid?ediddelete	Delete specified edid
/uri/auth/settings	List authentication settings
/uri/auth/settings?logout	Log out... if logged in, which isn't required

/uri/auth/settings?isAuthEnabled	Query if authentication is enabled
/uri/auth/settings?getAuthStatus	Get authentication status
/uri/auth/settings?username=[username]&password=[password]	Log in... not that it's needed as no API calls check if authenticated
/uri/ethernet/deviceSettings	List device settings e.g. internal IP address gateway
/uri/ethernet/deviceSettings?ipaddress	Get / set IP address
/uri/ethernet/deviceSettings?IPT	View or add/remove iptable
/uri/ethernet/deviceSettings?mask	Get / set subnet mask
/uri/ethernet/deviceSettings?gateway	Get / set gateway
/uri/ethernet/deviceSettings?dhcp	Get / set DHCP options
/uri/ethernet/deviceSettings?hostname	Get / set host name
/uri/ethernet/deviceSettings?domainname	Get / set domain name
/uri/ethernet/deviceSettings?primarydns	Get / set primary DNS
/uri/ethernet/deviceSettings?secondarydns	Get / set secondary DNS
/uri/ethernet/deviceSettings?macaddress	Get / set MAC address
/uri/ethernet/deviceSettings?reboot	Reboot device
/uri/ethernet/deviceSettings?encryptconn	Get / set CSIP usage
/uri/system/join	Unclear what this and sub-commands do
/uri/system/sysinfo	Query system info e.g. firmware version and serial
/uri/system/sysinfo?fwupgrade	Upgrade firmware
/uri/system/sysinfo?reboot	Send command to reboot device
/uri/system/sysinfo?restore	Restore device settings
/uri/txrx/device/HDMIIN	Get HDMI input configuration
/uri/txrx/device/HDMIIN?Resolution=	Change resolution
/uri/txrx/device/HDMIIN?ediddelete	Delete edid
/uri/txrx/device/HDMIIN?Tx_CEC_Msg=	Set value of transmit message box text

/uri/txrx/device/HDMIOUT	HDMI output configuration
/uri/txrx/device/OSD	Query OSD information

As evident from the above table, unauthenticated, remote attackers may perform any administrative actions desired on the device.

Hardcoded X.509 Certificate (CVE-2016-5669)

The device offers access to the web-based management interface both over HTTP and HTTPS. Ideally, the latter option should offer secure communication, but the device uses a known, unsafe X.509 certificate intended for testing purposes only. As a result, it does not provide any additional security, allowing an attacker to e.g. conduct spoofing attacks or passively decrypt traffic.

Details of the X.509 certificate can be found here.
https://www.censys.io/certificates/51ab293c9fe391eeeb1a2739de15cd8029e3033142962c6c386f2da78d03a945

Default Credentials (CVE-2016-5670)

The default username and password to the web-based management interface is `admin:admin` without encouraging users to change this when setting up the device or providing a straight-forward way of doing so.

Cross-Site Request Forgery (CVE-2016-5671)

The web-based management interface does not require multiple steps, explicit confirmation, or a unique token when performing sensitive actions. By tricking a user into visiting a malicious website or following a specially crafted link, an attacker can e.g. change device settings with the permissions of the user i.e. conduct a Cross-Site Request Forgery (CSRF) attack. This works both against all web pages served by the device and JSON API.

Solution

Upgrade to firmware version 1.3039.00040 or later. This addresses the authentication bypass vulnerabilities and use of the insecure hard-coded X.509 certificate. While the device still comes with default admin credentials, the new firmware version makes it easy for system administrators to change this via the web-based management interface.

The CSRF issues are still unpatched, but will be addressed in a future version.

References

RBS:	RBS-2016-003[4]
VulnDB:	142310, 142311, 142312, 142313, 142314, 142315
US-CERT:	VU#974424
CVE:	CVE-2016-5666, CVE-2016-5667, CVE-2016-5668, CVE-2016-5669, CVE-2016-5670, CVE-2016-5671

Timeline

2016-04-03	Vulnerabilities discovered.
2016-04-19	Vulnerabilities reported to US-CERT.
2016-04-21	Vulnerabilities reported to the vendor by US-CERT.
2016-04-25	Vulnerabilities acknowledged by the vendor.
2016-05-10	Conference call with the vendor to discuss vulnerabilities and solutions.
2016-05-13	Device for further testing received by courier from the vendor.
2016-05-19	Feedback provided to the vendor based on additional testing of proposed fixes in latest development firmware version.
2016-05-25	The vendor asks for an extension to July 1st, 2016 before public disclosure.
2016-06-14	The vendor confirms being on track for release on July 1st, 2016.
2016-06-17	The vendor asks for an additional 30-day extension due to problems uncovered during testing. Disclosure pushed to August 1st, 2016.
2016-08-01	Alerts sent to RBS VulnDB clients. Blog posted[5]. US-CERT publishes VU#974424.
2016-08-31	Publication of this vulnerability report.

[4] https://www.riskbasedsecurity.com/research/RBS-2016-003.pdf
[5] https://www.riskbasedsecurity.com/2016/08/risk-based-security-discovers-critical-vulnerabilities-in-crestron-product/

About Risk Based Security

Risk Based Security offers clients fully integrated security solutions, combining real-time vulnerability and threat data, as well as the analytical resources to understand the implications of the data, resulting in not just security, but the right security.

Company History

Risk Based Security, Inc. (RBS) was established to support organizations with the technology to turn security data into actionable information and a competitive advantage. We do so by enhancing the research available and providing a first of its kind risk identification and evidence-based security management service.

As a data driven and vendor neutral organization, RBS is able to deliver focused security solutions that are timely, cost effective, and built to address the specific threats and vulnerabilities most relevant to the organizations we serve. We not only maintain vulnerability and data breach databases, we also use this information to inform our entire practice.

Solutions

VulnDB - Vulnerability intelligence, alerting, and third party library tracking based on the largest and most comprehensive vulnerability database. Available as feature-rich SaaS portal or powerful API. Vendor evaluations including our Vulnerability Timeline and Exposure Metrics (VTEM) as well as Cost of Ownership ratings.

Cyber Risk Analytics - Extensive data breach database including interactive dashboards and breach analytics. Clients are able to gather and analyze security threat and data breach information on businesses, industries, geographies, and causes of loss. It also allows monitoring of domains for data breaches and leaked credentials as well as implementing a continuous vendor management program with our PreBreach data.

YourCISO - Revolutionary service that provides organizations an affordable security solution including policies, vulnerability scans, awareness material, incident response, and access to high quality information security resources and consulting services.

Vulnerability Assessments (VA) and Pentesting - Regularly scheduled VAs and pentests help an organization identify weaknesses before the bad guys do. Managing the most comprehensive VDB puts us in a unique position to offer comprehensive assessments, combining the latest in scanning technology and our own data. Detailed and actionable reports are provided in a clear and easy to understand language.

Security Development Lifecycle (SDL) - Consulting, auditing, and verification specialized in breaking code, which in turn greatly increases the security of products.