Thoughts on Streaming Video Securely

Howdy Pierce, Cardinal Peak • December 17, 2013

As the Internet of Things (IoT) continues to take off, a reasonable number of those Things involve remotely streaming video or still images. Examples include IP security cameras, baby monitors, and televisions with built-in cameras for videoconferencing.

Unfortunately, with a few exceptions, most of these devices stream the video in the clear, completely unencrypted. It’s a sad state of affairs.

To kick off this blog series, which will discuss video as it pertains to embedded devices, I thought it would be worthwhile to summarize what options are available when it comes to standards-based secure video streaming, and talk about the tradeoffs inherent in embedded devices.

First, a quick survey of the current marketplace. Most video streaming devices don’t encrypt the video they stream, because doing so is computationally expensive. So while you might connect to an embedded web server on the device via HTTPS, and you probably have to log into the device to control it, all that security is limited to the “control plane.” The video itself will almost certainly be transmitted unencrypted. If it complies with an open standard, it is probably being sent over RTP/RTSP or over HTTP Live Streaming (HLS).

Until recently, the accepted wisdom in the industry was that end users didn’t care about encrypting this type of video, as long as it was a little difficult for a naïve user to access it without permission. With a near-constant stream of news regarding more and more sophisticated network attacks, however, this logic is feeling strained.

I won’t get into the gory details here, but at a high level there are three primary standards-based means of streaming encrypted video:

- Secure RTP, or SRTP, is an IETF standard based around 128-bit AES encryption (in the default configuration). It is widely used in VoIP telephony but as far as I know it is rarely used in streaming video.
- Also in the RTP/RTSP family, video can be sent via “HTTP Tunneling”—specifically, over RTP/RTSP/HTTP, as I described in a blog post about RTP. However, in order to encrypt the content, you would replace the HTTP with HTTPS, thus using TLS. This mechanism is discussed in section 5.1.1.4 of the ONVIF Streaming Spec, but I don’t believe it is widely utilized.
- Finally, video can be sent using HLS, but again over HTTPS and TLS. This approach is more complex to implement but—if the embedded device has the capability—could offer the promise of adaptive bit rate streaming, which the other approaches don’t.
Of course, in both of the latter cases, content is being encrypted with TLS, formerly called SSL. TLS supports a list of ciphers—the actual cipher being used is negotiated at the time of connection between the two hosts involved—but common ciphers are 128-bit or 256-bit AES.

(As an aside, TLS also supports less computationally complex ciphers such as RC4. Unfortunately, RC4 is no longer considered secure, and you really shouldn’t implement it in a current product.)

Because each of the above methods involves AES, streaming video securely implies that your device is able to perform AES encryption at the same data rate it’s capable of encoding video. Now, although encryption requires significant computation, it’s much less computationally involved than what a decent h.264 encoder has to do to encode the video in the first place. The problem is that while the video encoding is done in hardware, for many current architectures the encryption would need to be done in software—with the obvious consequences in both cost and heat generation.

In the long run, though, the added burden of encryption needn’t be expensive: Just look at the industry for streaming entertainment video to consumers. In this case, devices such as Rokus, TiVos, Chromecasts, and smart TVs are performing decryption. But since the complexity of AES is roughly equivalent regardless of whether you are encrypting or decrypting, the fact that you can buy a Chromecast for $35 shows that AES is not inherently cost prohibitive. The problem is that we need silicon vendors to incorporate it.

More about the author Howdy Pierce.