Security White Paper for the Pearl Family of video encoders

Document number: EPN110-1

Author: Yusupha Touray

Publication Date: 12/3/18

Table of Contents

Introduction

IT friendly live streaming gear provides security features to make it less vulnerable to viruses and malware, as well as make it a lot easier to deploy. Secure gear should make sure that its firmware has the latest security patches, provides mechanisms to encrypt content that it exchanges with other network devices, validates the identity of servers it interacts with to ensure they are not rogue devices, and can also identify itself as a trusted device on the network. Epiphan Video's Pearl family of all-in-one video production switching, recording, and streaming encoders comply with network security requirements, making them safer for your network. Our encoders support security features including 802.1x, HTTPS, RTMPS and SFTP.

Overview of 802.1x

Although firewalls and VPNs protect networks from attacks launched from outside a Local Area Network, they do nothing to protect against someone walking into an office building, connecting their computer to a network port and launching an attack from within. The IEEE 802.1x network security standard defines an authentication protocol that ensures that only known and authorized clients can connect to the Local Area Network (LAN) and use the services offered by the LAN.

The 802.1x standard defines three entities:

- a supplicant, which is a device attempting to connect to a network. For example, Epiphan Video's Pearl video encoder
- an authenticator, which is typically a network device such as a network switch
- an Authentication Server (AS)

In an 802.1x enabled network, the authenticator by default blocks all traffic from the supplicant except EAPOL packets, which is required to authenticate a client attempting to connect to a network port.

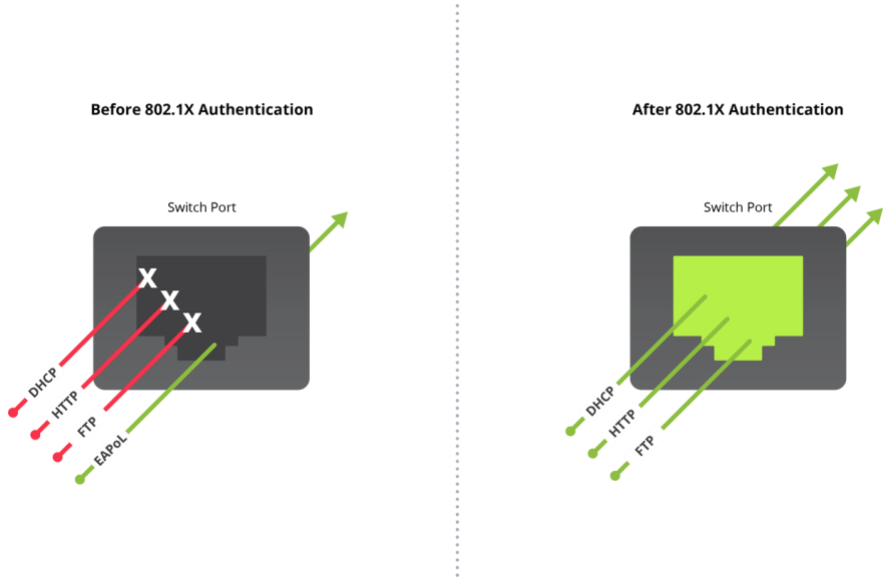

Figure 1 Before and after successful 802.1x authentication

Supported EAP methods

The authentication process is mutual in that the network authenticates Pearl in its role as the supplicant and Pearl authenticates the network. The latter ensures that Pearl doesn't connect to a rogue network. The process starts by Pearl and the AS negotiating which EAP method they will use to perform the mutual authentication.

The three EAP methods supported by Pearl are PEAP, EAP-TLS and EAP-TTLS. With PEAP, Pearl is identified by a user configured username and password combination. EAP-TTLS allows Pearl to be authenticated using either a combination of username and password or a digital certificate. Whereas, EAP-TLS requires the use of a digital certificate for authentication.

802.1x authentication process

Regardless of which EAP method is agreed on, the actual authentication process starts by the AS sending its digital certificate to the supplicant (i.e. Pearl) via the authenticator. Pearl validates the received certificate using the public key of a trusted Certificate Authority that was preinstalled on Pearl. Once Pearl validates the digital certificate from the AS, the AS's public key (that was received from the server along with the certificate) is used to establish an encrypted communication channel between Pearl and the AS.

Within this encrypted communication channel, the AS requests that Pearl sends its authentication credentials (protected by the encrypted channel) to the AS for validation. The encrypted channel keeps Pearl's credentials safe from prying eyes. The supplicant's credentials can be in the form of a username and password or digital certificate.

However, Pearl may reject the network's certificate and fail the authentication if Pearl is configured to verify the network's credentials but Pearl was not pre-configured with the digital certificate of trusted CA to verify this certificate. Or the network may reject and fail the authentication if the username and password combination configured on Pearl does not match what was configured on the network. In either case, the network port is blocked and Pearl is prevented from acquiring an IP address, which blocks admin access to Pearl or streaming from Pearl.

Once the AS has validated Pearl's credentials, it will notify the Authenticator to open the switch port to which Pearl is connected for all network traffic to and from Pearl.

Using the 802.1x feature on Pearl allows you to deploy Pearl in security conscious networks without having to use workaround methods such as MAB (MAC Address Authentication Bypass), which compromise the security of the network.

Overview of HTTPS

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, which is the protocol used to exchange data between a web browser and a web server, like the web server that is running on the Pearl family of video encoders. The secure part of the protocol provides two things: first, that the browser is communicating with a trusted and verifiable entity and second, that information such as username, passwords and channel previews, that are exchanged between the web browser and the server is protected from prying eyes.

Pearl's implementation of HTTPS uses Transport Layer Security to secure the communication between itself and a web browser that is connected to the Admin interface of Pearl. The HTTPS implementation employs TLS version 1.2, which is the latest and most secure version of the protocol at the time writing this paper. The TLS handshake that is described below is responsible for the authentication, negotiation encryption algorithm, and security key negotiation required to secure the communication.

TLS Handshake

The TLS handshake takes place between a TLS client and TLS server. The TLS client and server use the handshake to:

- Mutually authenticate the client and server
- Negotiate the suite of ciphers to use for message authentication and encryption
- Exchange session keys to used to protect communication between the client and server

Mutual authentication

One of the first things the TLS server does as part of establishing a communication session with the client is the server sends its digital certificate, public key, and set of encryption and hashing algorithms it supports to the TLS client. The server's certificate is either signed by a Certificate Authority that is mutually trusted by the server and client or by the TLS server itself if the server is using a self-signed certificate. The client uses the server's certificate and the trusted Certificate Authority certificate and public key to authenticate the TLS server. Optionally, the server may request a digital certificate from the client to authenticate the client.

Cipher suite negotiation and key exchange

After authenticating the server and the server optionally authenticating the client, the client randomly generates a pre-master secrete, which it encrypts with the server's public key and sends to the server. The server and client derive a master key and session key based on the pre-master key. The client then sends notification to the server indicating which cipher suites to use for encrypting and authenticating future messages between the two. The client and server can now exchange application data that is encrypted and authenticated by the derived session keys.

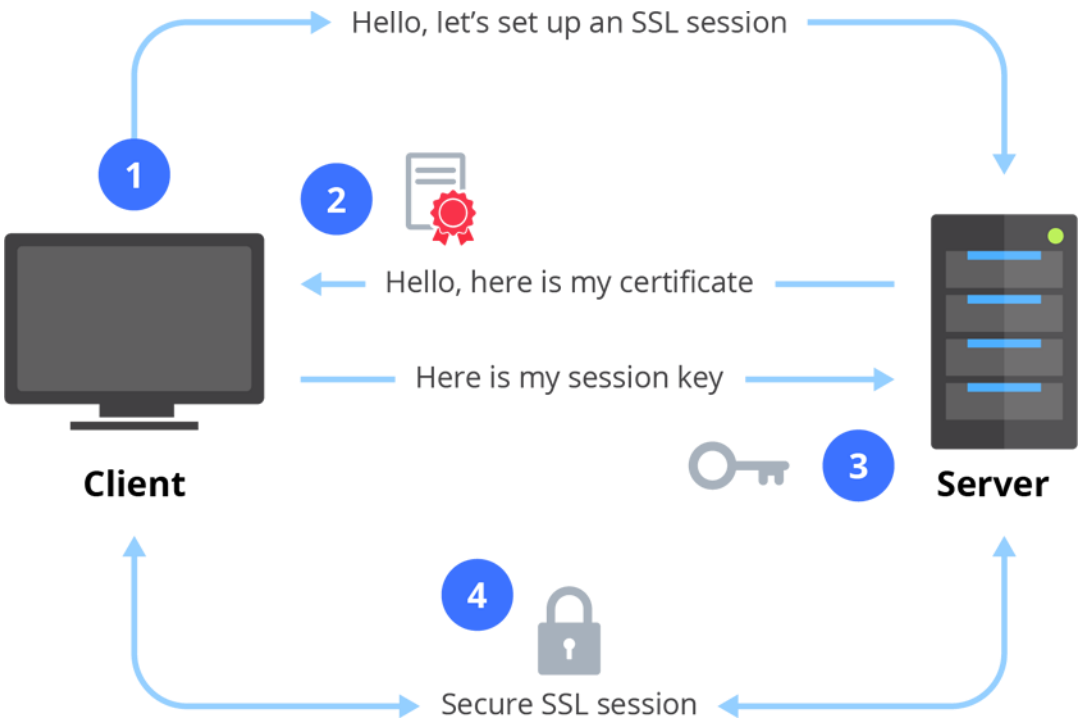

Figure 2 TLS handshake

How does HTTPS work on Pearl

The implementation of HTTPS on Pearl relies on the TLS handshake described above. When HTTPS is enabled on Pearl and the administrator points their web browser to the IP address of Pearl, the web browser takes on the role of TLS client and Pearl plays the role of TLS server.

- As described in the TLS handshake protocol, Pearl sends its public key and either the built-in self-signed digital certificate or a digital certificate that the Pearl administrator had previously uploaded to Pearl. The following cipher suites that Pearl supports in the current firmware release are also sent to the administrator's web browser. This is shown as step 2 in Figure 2. Symmetric encryption methods:
  - AES-GCM(128)
  - AES-GCM(256)
  - AES-CBC(128)
  - AES-CBC(256)
- Message authentication code methods:

    - SHA256
    - SHA384
- Key exchange protocol
    - ECDH

The web browser uses Pearl's certificate to validate the identity of Pearl. However, Pearl does not require the browser to be authenticated by requiring the latter to send its digital certificate to Pearl. Instead, authentication of the user of the web browser is done when the TLS handshake is complete and the administrator logs into Pearl Admin UI with a username and password.

After completing the authentication phase of the TLS handshake, the browser and Pearl begin the cipher suite negotiation and key exchange phase of the protocol. The negotiated cipher suites and key are used to protect all future exchanges of messages between the browser and Pearl, which is represented by Step 3 in Figure 2. The key and cipher suite negotiation ends when a session key has been established as depicted in step 4. From then onwards, any message that is exchanged between the web browser and Pearl is encrypted and authenticated using the negotiated cipher suites and key.

Connecting to Pearl using HTTPS ensures that all information including passwords and streaming keys are kept secure. In addition, snapshots and live previews of the audio and video content being recorded/streamed on Pearl is also kept secure over the HTTPS connection.

Overview of RTMPS

RTMPS (aka RTMP over TLS) is the encrypted version of RTMP, the popular live streaming protocol. RTMPS allows you to stream securely and is the right choice for events such as corporate town hall meetings, UX studies, or any other content containing confidential material. The protocol not only encrypts your content between the encoder and the CDN, it also protects against domain impersonation. To use RTMPS, both the video encoder and the CDN must support the protocol.

Pearl's implementation of RTMPS

Epiphan Video's implementation of RTMPS on their Pearl family of products is in many ways similar to HTTPS. RTMPS runs over TLS version 1.2 and uses the TLS handshake for authentication, cipher suite negotiation, and key exchange to protect video content that is exchanged between Pearl and the CDN. However, in contrast to HTTPS, Pearl plays the role of TLS client and the CDN plays the role of the server, as defined by the TLS handshake protocol.

During the connection setup, the CDN presents its digital certificate, public key, and cipher suites to Pearl for verification. When Pearl receives the CDN's digital certificate at the beginning of the RTMPS session, Pearl verifies the identity of the CDN using either a certificate from one of the Certificate Authorities that come already installed in Pearl or using a certificate that was manually uploaded to Pearl by the administrator. When verification passes, Pearl's administrator is assured that Pearl is sending the video content to the intended recipient and not to an impersonator.

Although TLS has provision for the CDN to authenticate Pearl, this is not widely implemented. After verifying the CDN's certificate, the CDN and Pearl begin the cipher suite negotiation and key exchange phase of the handshake. The derived keying materials are used to encrypt video content that is streamed from Pearl that can only be decrypted by the CDN.

Overview of SFTP

SFTP (SSH File Transfer Protocol) is a secure file transfer protocol. It allows a client to send a file securely over an otherwise insecure medium, such as a Local Area Network or over the public Internet. SFTP runs on top of another protocol called SSH (Secure Shell), which itself uses the TLS Handshake to authenticate the client and server, then establishes a secure connection between client and server.

Pearl's implementation of SFTP

SFTP implementation on Pearl products works in a similar manner as RTMPS in that it uses the TLS handshake for authentication, cipher suite negotiation, and key exchange to protect video assets transferred from the SFTP client on Pearl to an SFTP server. In contrast to RTMPS; however, both the SFTP client and SFTP server authenticate each other.

Similar to RTMPS, Pearl plays the role of the TLS client and the SFTP server plays the role of the server, as defined by the TLS handshake protocol.

During the connection setup, the SFTP server presents its digital certificate, public key, and cipher suites to Pearl for verification. When verification passes, Pearl's administrator is assured that when Pearl transfers recorded files to the SFTP server, it is sending it to a trusted entity and not an impersonator.

After verifying the SFTP server's certificate, the server and SFTP client begin the next phase of the handshake to authenticate Pearl. Pearl uses the username and password that was configured for the SFTP client on Pearl by the Pearl administrator. During the authentication phase of the SFTP client, Pearl's SFTP client's credentials are not sent unencrypted to the server, as is done if the FTP protocol is used. Rather, they are encrypted using the session key that was negotiated during the TLS handshake.

The final phase of the TLS handshake begins after successful mutual authentication between Pearl and the SFTP server. This is the cipher suite negotiation and key exchange phase of the handshake to derive the keying materials to use to encrypt recorded video files that are transferred from Pearl to a SFTP server located on a local LAN or across the Internet.

Using SFTP rather than FTP to transfer files from Pearl ensures that credentials are not sent in the clear for anyone in the middle to see and that recorded video files are also sent to the server within an encrypted connection.

Summary

Epiphan Video's Pearl family of encoders are network secure and IT friendly with built-in 802.1x network security, HTTPS, RTMPS, and SFTP support. Connecting Pearl to an enterprise network is easier, takes less time and effort to deploy, and won't require compromising your corporate network security policies. These protocols help to ensure sensitive information such as passwords and stream keys are kept secure and that the video content that is recorded and streamed is transferred over secure connections to authenticated servers.