Network Authentication - 802.1x

Secure the Edge of the Network

Table of contents

1 Secure the edge of the network

Security devices are mostly located at the physical edge of the network. Especially detection devices, such as cameras, are installed in places that are accessible by the public. As these devices are connected to the network, this also increases the risk of unwanted access to the network: people could try to disconnect the security device and connect their own equipment to try to gain access to the network, or attach pass-through equipment to try a so-called a man-in-the-middle attack.

There are several ways of mitigating such attempts:

- Ensure the device meets the requirements related to physical strength and cabling management: Bosch devices that have an IP66 or IP67 rating have this network connection point inside their housing. This means they need to be physically disassembled before the network connection point can be accessed. This can be further secured by using tamper-proof screws.
- Authenticate the device to the network before allowing it to access the network's resources: there are several ways to ensure that only authenticated devices can access the network. Bosch devices support authentication based on user-name and password (802.1x). In addition to 802.1x EAP-TLS can be used, which secures the whole authentication process.

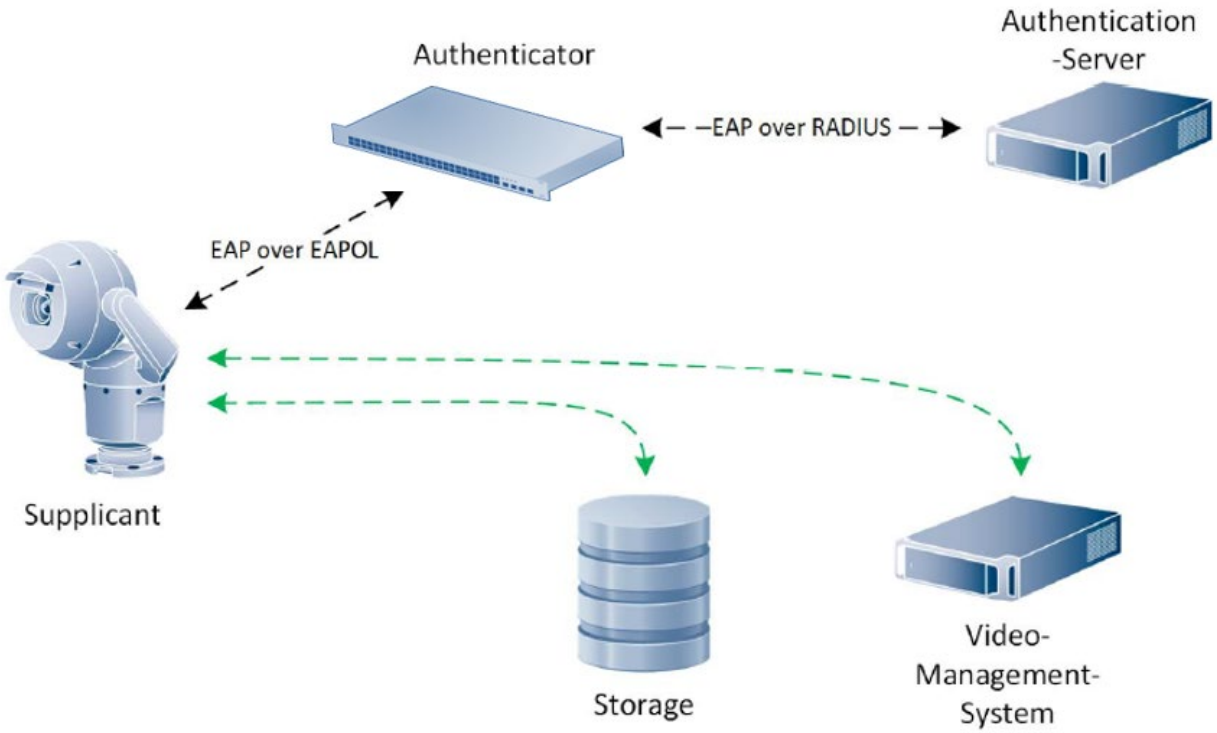

Figure 1: EAP (Extensible Authentication Protocol) data is first encapsulated in EAPOL frames between the Supplicant and Authenticator, then re-encapsulated between the Authenticator and the Authentication Server using RADIUS or Diameter.

2 IEEE 802.1x

IEEE 802.1x [1] is a standard published by the Institute of Electrical and Electronics Engineers Standards Association. This organization within the IEEE develops global standards in a broad range of industries, including: power and energy, biomedical and health care, information technology, telecommunication, transportation, nanotechnology, information assurance and many more. This particular standard is part of the IEEE 802.1 group of networking protocols and provides an authentication mechanism to prevent unauthorized devices to access network resources.

This protocol involves three kinds of main elements:

- The element that wants to be able to access the network resources is named the supplicant, for example a video surveillance camera.

- The element that verifies if the supplicant may access the network resources is named the authenticator. Mostly this is a manageable switch, router, or wireless access point.

- The element that actually steers the authentication process is named the authentication server. The authentication server contains the information that is used to decide if a supplicant may or may not access the network resources. Typically, this is a server that supports the RADIUS [3] protocol, which is a networking protocol that provides centralized authentication, authorization and accounting. The RADIUS protocol is part of the Internet Engineering Task Force (IETF) standards.

2.1 Extensible Authentication Protocol

The Extensible Authentication Protocol [2] is an authentication framework which supports multiple authentication methods. EAP typically runs directly over data link layers such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring IP. EAP provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees.

A typical EAP authentication procedure using RADIUS consists of four steps:

1. Initialization:
   After the authenticator detects that a device is connected to its port, this port is set to the "unauthorized" state and will only allow 802.1X traffic. Other traffic, such as UDP or TCP is not allowed and dropped.

2. Initiation:
   The authenticator will request the identity of the supplicant. When the authenticator receives this information it will forward it to the authentication server by means of the RADIUS protocol.

3. Negotiation:
   The authentication server verifies the supplicant identity and sends a challenge back to the supplicant via the authenticator. This challenge also contains the authentication method, which could be based on a user-name and password.

4. Authentication:
   The authentication server and supplicant agree on an authentication method and the supplicant will respond with the appropriate method by providing its configured credentials. If authentication is successful, the authenticator allows the supplicant access to the defined network resources.

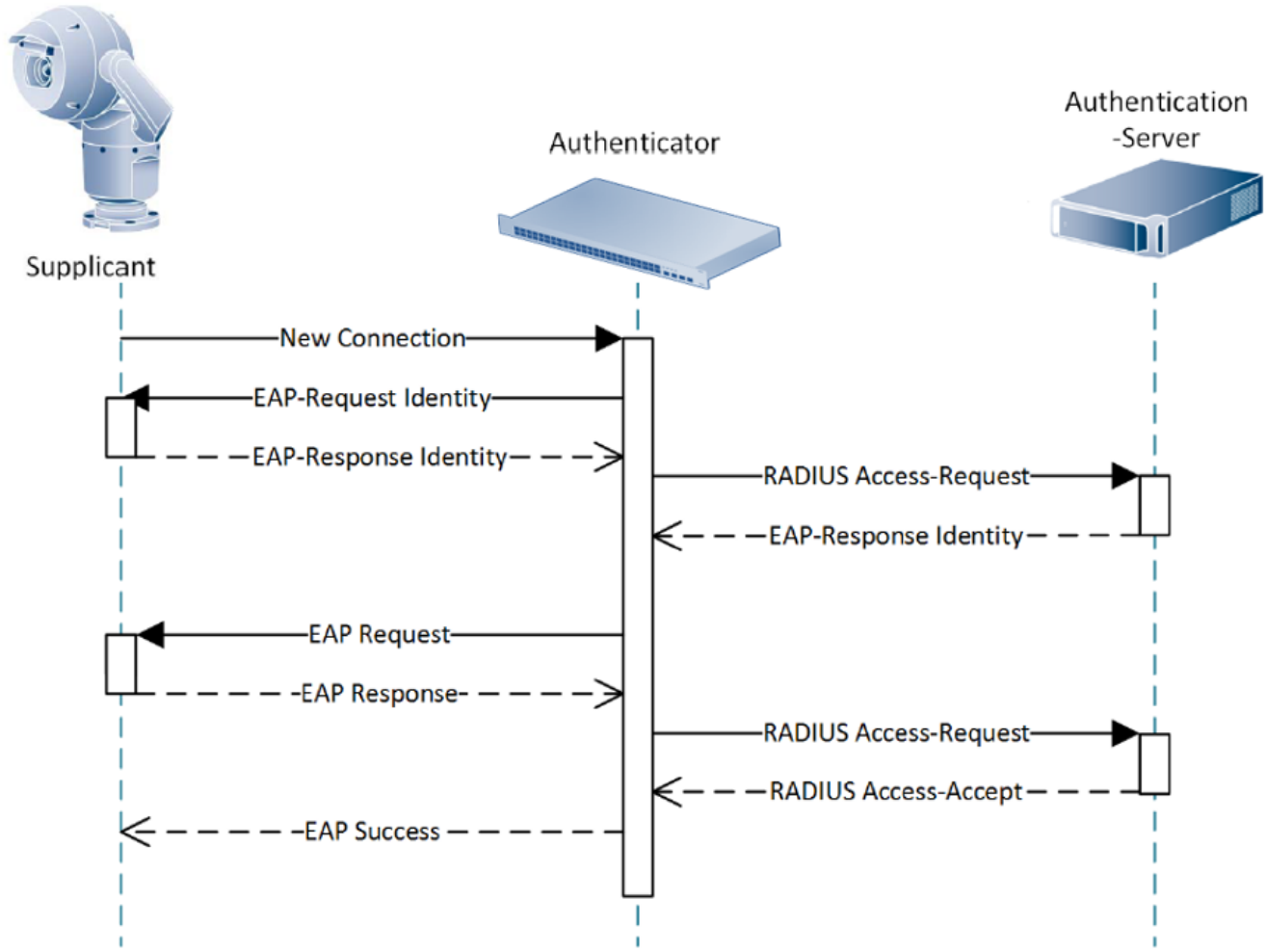

Figure 2: 802.1x Authentication Sequence Diagram

IEEE 802.1x itself does not provide a secure communication between the supplicant and authentication server. As a result, the user-name and password could be "sniffed" from the network.

To ensure a secure communication 802.1x can use EAP-TLS.

2.2 Extensible Authentication Protocol - Transport Layer Security

The Extensible Authentication Protocol (EAP), provides support for multiple authentication methods. Transport Layer Security (TLS) provides for mutual authentication, integrity-protected cipher suite negotiation, and key exchange between two endpoints. EAP-TLS [4] includes support for certificate-based mutual authentication and key derivation. In other words, EAP-TLS encapsulates the process in which both the server and client send each other a certificate.

2.2.1 Certificates

Digital certificates are used to verify that a public key belongs to a specific user or device, in other words, to verify that a user or device is actually telling the truth about its identity. These certificates are generated by a Certificate Authority (CA) based on specific details of the user or device. This Certificate Authority needs to be a trusted entity within an entire infrastructure and ensures that the certificates that are used in the infrastructure can be verified. A compromised Certificate Authority cannot be trusted any more, and can therefore also not verify the identity of a user or device.

The management of certificates, assertion, extension and revocation, is typically handled within a Public Key Infrastructure (PKI) [5].

2.2.2 EAP-TLS client certificate

The EAP-TLS client certificate binds the client's identity to a public key. This public key (as part of the certificate) is sent to the server and used to encrypt the communication between the client and server.

The requirement for a client-side certificate is what gives EAP-TLS its authentication strength. With a client-side certificate, a compromised password is not enough to break into EAP-TLS enabled systems because the intruder still needs to have the client-side certificate. The highest security available is when the "private keys" of client-side certificate are housed in SmartCards, or in embedded hardware key vaults in devices, e.g. secure crypto-coprocessors, Secure Elements, or Trusted Platform Modules, like in Bosch's security cameras. This is because there is no way to steal a client-side certificate's corresponding private key from a SmartCard or Secure Element without stealing the card itself – or the security camera. In the latter case, still no one could make use of it other than trying to connect the device to the network it is dedicated for.

2.2.3 EAP-TLS trusted certificate

When a client is presented with a server's certificate, the client tries to match the server's Certificate Authority (which is part of the certificate) against the client's list of trusted Certificate Authorities. If the issuing Certificate Authority is trusted, the client will verify that the certificate is authentic and has not been tampered with. Finally, the client will accept the certificate as proof of identity of the server.

3 Certificates in Bosch cameras

All Bosch cameras (FW 6.10 or newer) use a certificate store, which can be found in the ***Service*** section of the camera configuration. Both the EAP-TLS client certificate and the EAP-TLS trusted certificate need to be added to the store by using the ***Add*** button in the ***File list*** section. After the upload is completed the certificates can be selected in the ***Usage list***. To activate the use of the certificates the camera must be rebooted, which happens automatically after pressing the ***Set*** button, and 802.1x must be activated in ***Network->Advanced*** with credentials entered.

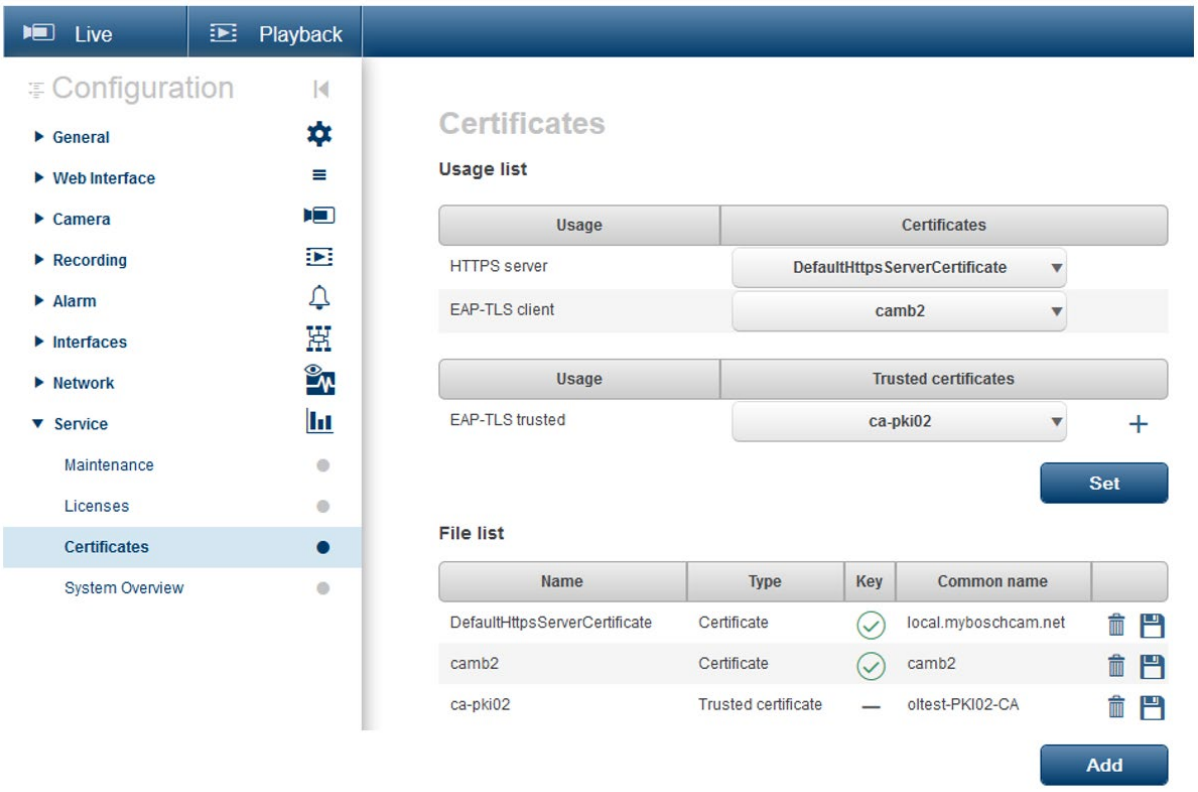

Figure 3: Example EAP/TLS certificates stored in a Bosch camera (FW6.11)

Since firmware version 6.40, the certificates are displayed as a combined list.

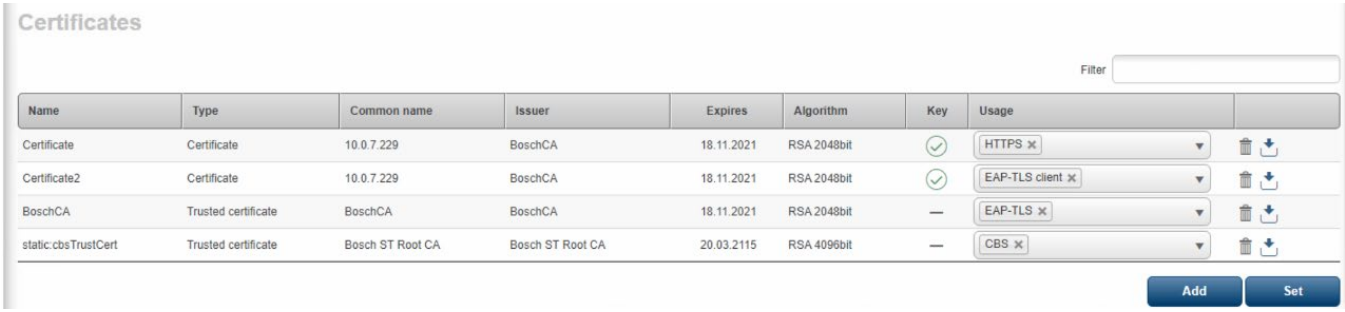

Figure 4: Example EAP/TLS certificates stored in a Bosch camera (FW6.40 and higher)

3.1 Secured in a safe

The certificates are stored in a chip like being used on SmartCards, also called a Secure Element, or “Trusted Platform Module”, or short TPM. This chip acts like a safe for critical data, a secure vault, protecting certificates, keys, licenses, etc. against unauthorized access even when the camera is broken up. More information on the Trusted Platform Module is available in a separate TechNote.

Certificates are accepted in multiple formats via the web interface: *.pem, *.crt, *.cer, *.p12, *.pfx, *.der
They may be uploaded as one combined file, or split into certificate and key parts and uploaded as separate files.

If uploaded using the Configuration Manager, the format will be automatically converted by Configuration Manager to a format accepted by the camera.

Please note that the maximum size for certificates is limited to 4096 Bytes.

4 Appendix – Example configuration

This example configuration focuses on the camera part and the special requirements for certificates used with them.
It does not focus on the environment, nor does it limit to a specific use case.

4.1 Prerequisite

- Use latest camera firmware. If not possible check the latest camera firmware release notes history for fixes you might miss with the older version to be aware of issues or restrictions you may encounter.
- Ensure the certificates you are going to use are based on 2048 bit keys.
- Ensure the ***Common Name (CN)*** on the certificates a.k.a. ***"Issued to"*** and ***"Issued by"*** are written without spaces or underscore and have a minimum length of 5 characters in all certificates.

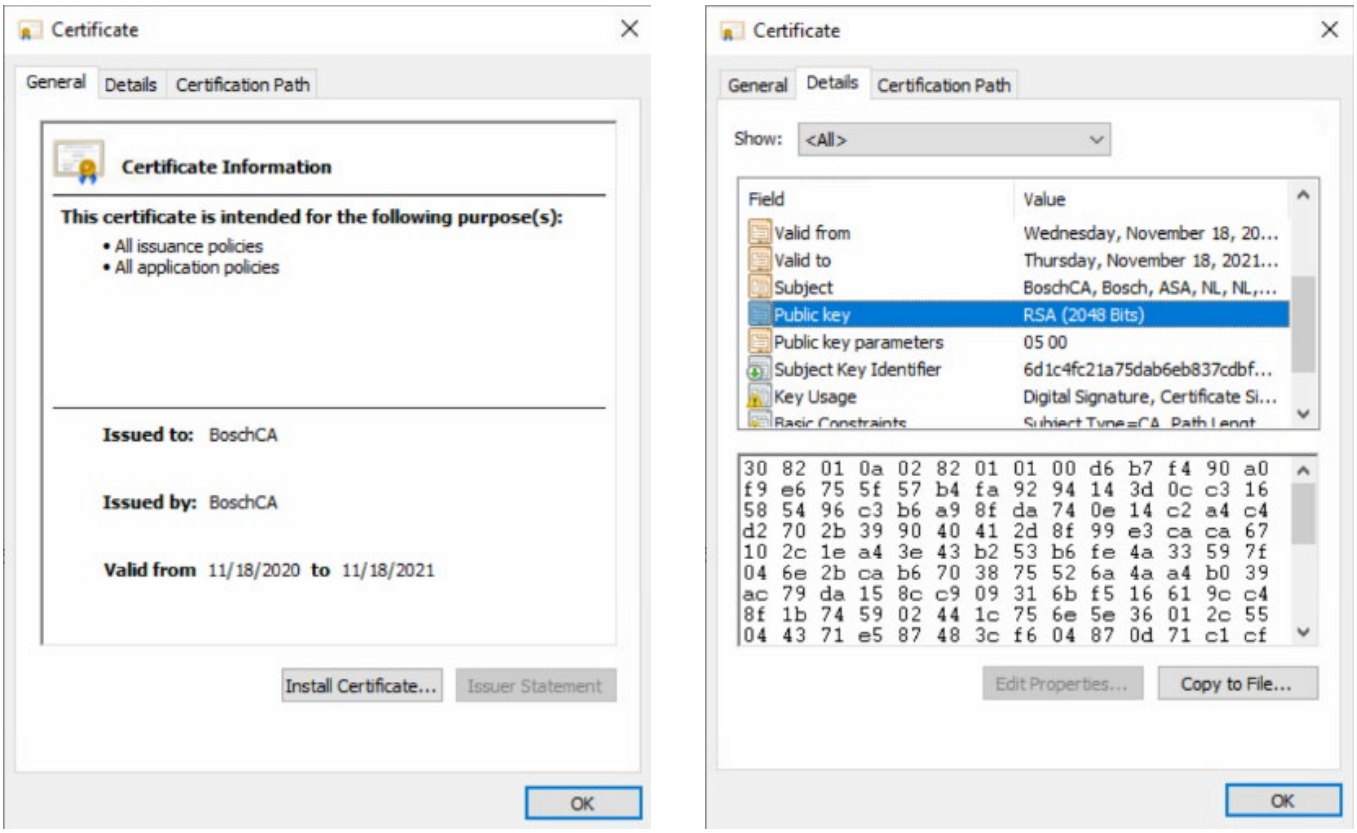

Figure4 – Example: Certificate details as shown on a Microsoft Windows workstation

4.2 How to configure the camera

1. Ensure the camera is in time-sync with its authenticating server, consider using a timeserver.
2. For the Client, upload the ***EAP-TLS client*** certificate and set the ***Usage***.
   Client certificate -> set usage: EAP-TLS client
   Note: In order to be able to select this usage the certificate must contain the “Private Key”. Once uploaded, this is recognizable by the green check-box in column ***Key***.

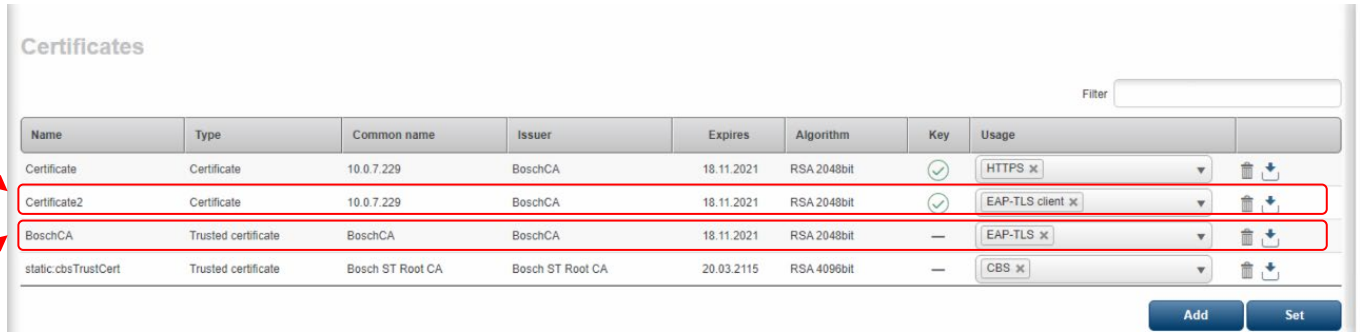

Figure 5: Example screenshot from webpage also indicating certificate Type, like e.g. “Trusted Certificate”. This is not shown in Configuration Manager (7.20).

3. For the server, upload the ***EAP-TLS*** certificate (*a.k.a. “Trusted certificate”)* and set the ***Usage***
   Server certificate -> set usage: EAP-TLS
   Note: This certificate does not contain the private key as this resides on the authenticating server.
4. Enable 802.1X on the camera (***Network->Advanced->802.1X***) by setting ***Authentication*** to ***On***.
5. ***Identity***: Typically, the user name for identifying the camera is provided via the *Common Name* field in the certificate. If not, or if the username provided via *Common Name* shall be ignored, enter the username that the authenticating server uses for identifying the camera. This is only mandatory, if the dial-in network has defined a challenge for the authentication server.
6. ***Password***: Leave empty if no username is required but taken from *Common Name* field. This is only required for EAP-MD5 or if user shall not be taken from certificate.

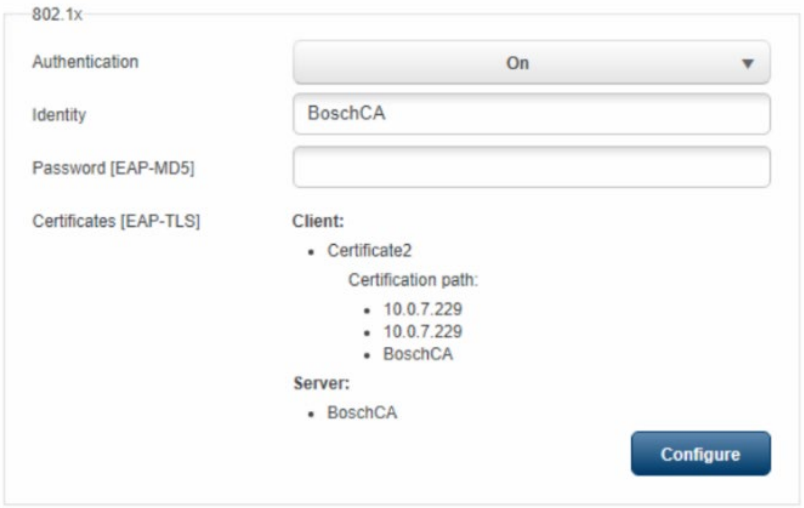

Figure 6: Configuration section for 802.1x from the webpage of the camera.

7. Reboot the camera after you are done.

4.3 Logging in case of troubleshooting

If a configuration like the above does not lead to a functioning system, some additional logging may help investigating the issue. In case of technical support being required, these loggings will also be needed to enable tech support help solving the issue.

- Take a Wireshark capture from a port mirror where the camera shall get connected to.
- Start the capture before plugging the camera to the switch.
- Gather all used certificates, usernames and passwords.
- Download the maintenance log from the camera.
- Download the camera configuration file (pull at the same time as other logs) and passwords for service and loading.
- Gather network schematics.
- Gather the configuration/settings and used certificates of the authenticating server.
- Install a syslog server on the Wireshark PC and configure printouts: *syslog_dbg eapol ssl*.
  This way the syslog server automatically starts collecting log information when the camera connects to the switch, and is in sync with the matching Wireshark capture.

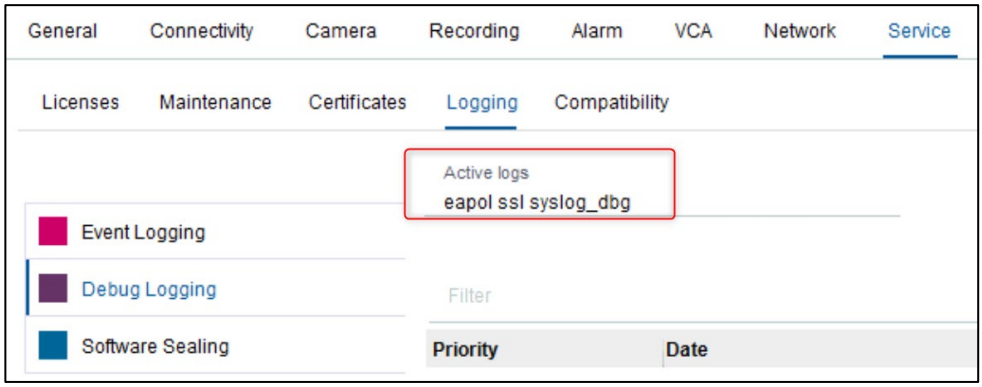

Figure 7: Enable additional logging for 802.1x troubleshooting in Configuration Manager.

5 References

1. 802.1x, IEEE standard for port-based Network Access Control
http://www.ieee802.org/1/pages/802.1x-2004.html

2. RFC 3748, Extensible Authentication Protocol (EAP),
https://tools.ietf.org/html/rfc3748

3. RFC 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines
https://tools.ietf.org/html/rfc3580

4. RFC 5216, The EAP-TLS Authentication Protocol,
http://www.ietf.org/rfc/rfc5216.txt

5. RFC 3280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
https://www.ietf.org/rfc/rfc3280.tx t

Bosch Sicherheitssysteme GmbH
Robert-Bosch-Ring 5
85630 Grasbrunn
Germany
www.boschsecurity.com