ONVIF Advanced Security Client Test Specification

Version 16.07 July, 2016

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

Table of Contents

Introduction

The goal of the ONVIF Test Specification set is to make it possible to realize fully interoperable IP physical security implementations from different vendors. This specification also acts as an input document to the development of a test tool which will be used to test the ONVIF Client implementation conformance towards ONVIF standard. This Client Test Tool analyzes network communications between ONVIF Devices and Clients being tested and determines whether a specific Client is ONVIF conformant (see ONVIF Conformance Process Specification [http://www.onvif.org/Documents/Specifications.aspx]).

This particular document defines test cases required for testing Imaging Service features of a Client application e.g. Get Imaging Capabilities, Video Sources List, Get Imaging Settings, Imaging Settings Configuration, Focus Control. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

Scope

This ONVIF Advanced Security Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Advanced Security Service features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Advanced Security Service features according to ONVIF Advanced Security Service Specification.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Advanced Security Service features.

This specification **does not** address the following:

- 3rd parties Client use cases
- Non-functional (performance and regression) testing and analysis.
- SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
- Network protocol implementation Conformance test for HTTPS and HTTP protocols.

The following sections cover test cases needed for the verification of relevant features as mentioned in the ONVIF Profile Specifications.

Get Imaging Capabilities

Get Imaging Capabilities section specifies Client ability to request imaging capabilities from Device.

Normative references

ONVIF Conformance Process Specification:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Profile Policy:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Core Specifications:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Core Client Test Specification:

http://www.onvif.org/Documents/Specifications.aspx

ONVIF Advanced Security Specification:

http://www.onvif.org/Documents/Specifications.aspx

ISO/IEC Directives, Part 2:

http://www.iso.org/directives

ISO 16484-5:2014-09 Annex P:

https://www.iso.org/obp/ui/#!iso:std:63753:en

WS-BaseNotification:

http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

W3C SOAP 1.2, Part 1, Messaging Framework:

http://www.w3.org/TR/soap12-part1/

W3C XML Schema Part 1: Structures Second Edition:

http://www.w3.org/TR/xmlschema-1/

W3C XML Schema Part 2: Datatypes Second Edition:

"http://www.w3.org/TR/xmlschema-2/ [http://www.w3.org/TR/xmlschema-2/]

W3C Web Services Addressing 1.0 – Core:

http://www.w3.org/TR/ws-addr-core/

Rules for the structure and drafting of International Standards, Annex H: Verbal forms for the expression of provisions.

Terms and Definitions

Conventions

The key words "shall", "shall not", "should", "should not", "may", "need not", "can", "cannot" in this specification are to be interpreted as described in [ISO/IEC Directives Part 2].

Definitions

This section describes terms and definitions used in this document.

Profile
See ONVIF Profile Policy.

ONVIF Device
Computer appliance or software program that exposes one or multiple ONVIF Web Services.

ONVIF Client
Computer appliance or software program that uses ONVIF Web Services.

Conversation
A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.

Network
A network is an interconnected group of devices communicating using the Internet protocol.

Network Trace Capture file
Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.

SOAP
SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.

Client Test Tool
ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.

Advanced Security Service
Service for keystore and a TLS server on an ONVIF device.

Valid Device Response
Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.

Abbreviations

This section describes abbreviations used in this document.

HTTP	Hyper Text Transport Protocol.
HTTPS	Hyper Text Transport Protocol over Secure Socket Layer.
URI	Uniform Resource Identifier.
WSDL	Web Services Description Language.
XML	eXtensible Markup Language.

Namespaces

Prefix and namespaces used in this test specification are listed in Table 1. These prefixes are not part of the standard and an implementation can use any prefix.

Table 1. Defined namespaces in this specification

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML schema instance namespace
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
tds	http://www.onvif.org/ver10/device/wsdl	The namespace for the WSDL device service
trt	http://www.onvif.org/ver10/media/wsdl	The namespace for the WSDL media service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
tptz	http://www.onvif.org/ver20/ptz/wsdl	The namespace for the WSDL PTZ service
trv	http://www.onvif.org/ver10/receiver/wsdl	The namespace for the WSDL receiver service
trc	http://www.onvif.org/ver10/recording/wsdl	The namespace for the WSDL recording service
tse	http://www.onvif.org/ver10/search/wsdl	The namespace for the WSDL search service
trp	http://www.onvif.org/ver10/replay/wsdl	The namespace for the WSDL replay service
tac	http://www.onvif.org/ver10/accesscontrol/wsdl	The namespace for the WSDL access control service
tdc	http://www.onvif.org/ver10/doorcontrol/wsdl	The namespace for the WSDL door control service
tas	http://www.onvif.org/ver10/advancedsecurity/wsdl	The namespace for the WSDL advanced security service
tar	http://www.onvif.org/ver10/accessrules/wsdl	The namespace for the WSDL access rules service
tcr	http://www.onvif.org/ver10/credential/wsdl	The namespace for the WSDL credential service
tsc	http://www.onvif.org/ver10/schedule/wsdl	The namespace for the WSDL schedule service
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.
timg	http://www.onvif.org/ver20/imaging/wsd	The namespace for the WSDL imaging service

Test Overview

This section provides information for the test setup procedure and required prerequisites that should be followed during test case execution.

An ONVIF client with Advanced Security features support can provide key configuration, certificate configuration and TLS server configuration.

An ONVIF Profile is described by a fixed set of functionalities through a number of services that are provided by the ONVIF standard. A number of services and functionalities are mandatory for each type of ONVIF Profile. An ONVIF Device and ONVIF Client may support any combination of Profiles and other optional services and functionalities.

Test Setup

Collect Network Traces files required by the test cases.

Collect Feature List XML files for Devices detected in the Network Trace files.

Client shall support all mandatory and conditional features listed in the Device Feature List XML file supplied for the Profiles supported by the Client.

For ONVIF compatibility, the ONVIF Client shall follow the requirements of the conformance process. For details please see the latest ONVIF Conformance Process Specification.

Prerequisites

The pre-requisites for executing the test cases described in this Test Specification include:

The Device shall be configured with an IPv4 address.

The Device shall be able to be discovered by the Client.

Advanced Security Test Cases

Validated Feature: advanced_security

Profile A Requirement: None

Profile C Requirement: None

Profile G Requirement: None

Profile Q Requirement: Conditional

Profile S Requirement: None

Expected Scenarios Under Test: [TODO]

1. Client connects to Device to upload a passphrase to the keystore.

2. Client is considered as supporting Upload Passphrase if the following conditions are met:

   - Client is able to upload a passphrase to the keystore of the Device using **UploadPassphrase** operation.

3. Client is considered as NOT supporting Upload Passphrase if ANY of the following is TRUE:

   - No valid responses for **UploadPassphrase** request.

UPLOAD PASSPHRASE

Test Label: Upload Passphrase

Test Case ID: ADVANCEDSECURITY-1

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Upload Passphrase

Test Purpose: To verify that Client is able to upload a passphrase to the keystore of the Device using **UploadPassphrase** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **UploadPassphrase** operation present.
- Device supports Advanced Security Service.
- Device supports Passphrase handling.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **UploadPassphrase** request message to upload a passphrase to the Device.
2. Device responses with code HTTP 200 OK and **UploadPassphraseResponse** message.

Test Result:

PASS -

- Client **UploadPassphrase** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **UploadPassphrase** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:UploadPassphrase** AND
- Device response on the **UploadPassphrase** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:UploadPassphraseResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.upload_passphrase

DELETE PASSPHRASE

Test Label: Delete Passphrase

Test Case ID: ADVANCEDSECURITY-2

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Passphrase

Test Purpose: To verify that Client is able to delete a passphrase from the keystore of the Device using **DeletePassphrase** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **DeletePassphrase** operation present.
- Device supports Advanced Security Service.
- Device supports Passphrase handling.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeletePassphrase** request message to delete a passphrase from the Device.
2. Device responses with code HTTP 200 OK and **DeletePassphraseResponse** message.

Test Result:

PASS -

- Client **DeletePassphrase** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeletePassphrase** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:DeletePassphrase** AND
- Device response on the **DeletePassphrase** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:DeletePassphraseResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.delete_passphrase

CREATE PKCS#10 CERTIFICATION

Test Label: Create PKCS#10 Certification

Test Case ID: ADVANCEDSECURITY-3

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Create PKCS#10 Certification

Test Purpose: To verify that Client is able to generates a DER-encoded PKCS#10 using **CreatePKCS10CSR** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreatePKCS10CSR** operation present.

- Device supports Advanced Security Service.

- Device supports PKCS10ExternalCertificationWithRSA.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **CreatePKCS10CSR** request message to generate PKCS#10 on the Device.

2. Device responses with code HTTP 200 OK and **CreatePKCS10CSRResponse** message.

Test Result:

PASS -

- Client **CreatePKCS10CSR** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **CreatePKCS10CSR** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:CreatePKCS10CSR** AND

- Device response on the **CreatePKCS10CSR** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:CreatePKCS10CSRResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.create_pkcs10

UPLOAD CERTIFICATE

Test Label: Upload Certificate

Test Case ID: ADVANCEDSECURITY-4

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Upload Certificate

Test Purpose: To verify that Client is able to upload a certificate using **UploadCertificate** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **UploadCertificate** operation present.
- Device supports Advanced Security Service.
- Device supports PKCS10ExternalCertificationWithRSA.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **UploadCertificate** request message to upload a certificate on the Device.
2. Device responses with code HTTP 200 OK and **UploadCertificateResponse** message.

Test Result:

PASS -

- Client **UploadCertificate** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **UploadCertificate** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:UploadCertificate** AND
- Device response on the **UploadCertificate** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:UploadCertificateResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.upload_certificate

DELETE CERTIFICATE

Test Label: Delete Certificate

Test Case ID: ADVANCEDSECURITY-5

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Certificate

Test Purpose: To verify that Client is able to delete a certificate using **DeleteCertificate** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **DeleteCertificate** operation present.

- Device supports Advanced Security Service.

- Device supports PKCS10ExternalCertificationWithRSA or SelfSignedCertificateCreationWithRSA or PKCS12CertificateWithRSAPrivateKeyUpload.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteCertificate** request message to delete a certificate from the Device.

2. Device responses with code HTTP 200 OK and **DeleteCertificateResponse** message.

Test Result:

PASS -

- Client **DeleteCertificate** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **DeleteCertificate** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:DeleteCertificate** AND

- Device response on the **DeleteCertificate** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:DeleteCertificateResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.delete_certificate

DELETE CERTIFICATION PATH

Test Label: Delete Certification Path

Test Case ID: ADVANCEDSECURITY-6

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Certification Path

Test Purpose: To verify that Client is able to delete a certification path using **DeleteCertificationPath** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **DeleteCertificationPath** operation present.
- Device supports Advanced Security Service.
- Device supports TLSServer or PKCS12CertificateWithRSAPrivateKeyUpload.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteCertificationPath** request message to delete a certification path from the Device.
2. Device responses with code HTTP 200 OK and **DeleteCertificationPathResponse** message.

Test Result:

PASS -

- Client **DeleteCertificate** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteCertificationPath** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:DeleteCertificationPath** AND
- Device response on the **DeleteCertificationPath** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:DeleteCertificationPathResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.delete_certification_path

DELETE KEY

Test Label: DeleteKey

Test Case ID: ADVANCEDSECURITY-7

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Key

Test Purpose: To verify that Client is able to delete a key using **DeleteKey** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **DeleteKey** operation present.
- Device supports Advanced Security Service.
- Device supports MaximumNumberOfKeys.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteKey** request message to delete a key from the keystore of Device.
2. Device responses with code HTTP 200 OK and **DeleteKeyResponse** message.

Test Result:

PASS -

- Client **DeleteKey** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteKey** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:DeleteKey** AND
- Device response on the **DeleteKey** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:DeleteKeyResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.delete_key

GET KEY STATUS

Test Label: Get Key Status

Test Case ID: ADVANCEDSECURITY-8

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Key

Test Purpose: To verify that Client is able to get key status using **GetKeyStatus** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetKeyStatus** operation present.

- Device supports Advanced Security Service.

- Device supports MaximumNumberOfKeys.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetKeyStatus** request message to get a key status from the Device.

2. Device responses with code HTTP 200 OK and **GetKeyStatusResponse** message.

Test Result:

PASS -

- Client **GetKeyStatus** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetKeyStatus** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:GetKeyStatus** AND

- Device response on the **GetKeyStatus** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:GetKeyStatusResponse**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.get_key_status

UPLOAD PKCS12

Test Label: Upload PKCS12

Test Case ID: ADVANCEDSECURITY-9

Profile A Reference: None

Profile C Reference: None

Profile G Reference: None

Profile Q Reference: Conditional

Profile S Reference: None

Feature Under Test: Delete Key

Test Purpose: To verify that Client is able to uploads a certification path consisting of X.509 certificates using **UploadCertificateWithPrivateKeyInPKCS12** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **UploadCertificateWithPrivateKeyInPKCS12** operation present.

- Device supports Advanced Security Service.
- Device supports PKCS12CertificateWithRSAPrivateKeyUpload.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **UploadCertificateWithPrivateKeyInPKCS12** request message to upload a PKCS12 to the Device.
2. Device responses with code HTTP 200 OK and **UploadCertificateWithPrivateKeyInPKCS12Response** message.

Test Result:

PASS -

- Client **UploadCertificateWithPrivateKeyInPKCS12** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **UploadCertificateWithPrivateKeyInPKCS12** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tas:UploadCertificateWithPrivateKeyInPKCS12** AND
- Device response on the **UploadCertificateWithPrivateKeyInPKCS12** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tas:UploadCertificateWithPrivateKeyInPKCS12Response**.

FAIL -

- The Client failed PASS criteria.

Validated Feature List: advanced_security.upload_pkcs12

A. Revision History

December 30, 2015 Version 16.07

- Initial version:

  General parts added

  UPLOAD PASSPHRASE Test Case added

  DELETE PASSPHRASE Test Case added