Driving IP-based physical security through global standardization

ONVIF™

Profile Q Test Specification

Version 17.01

January 2017

Driving IP-based physical security through global standardization

Revision History

Ver.	Date	Description
14.12		First issue
15.06	June, 2015	The following test cases were updated: Factory Default state verification Default access policy - Operator Default access policy - Administrator and Anonymous Default access policy - Administrator and User/Operator Processor Usage event Last Reboot event (Status Change) Firmware Upload
16.01	January, 2016	The parameter value "UTCDateTime" has been changed in step 7 of QUICK_INSTALL-3-1-7 and step 10 of QUICK_INSTALL-3-1-6. Clarifications have been added to steps 6 7 of QUICK_INSTALL-3-1-4. The FAIL condition:"The DUT did not send UnsubscribeResponse message." has been removed from test QUICK_INSTALL-3-1-4. The steps 32-34 have been added to section 4.1.1 (QUICK_INSTALL-1-1-1) The steps 8-10 have been added to section 4.3.2 (QUICK_INSTALL-4-1-2)
17.01	January, 2017	Section 4.3 Monitoring Events was removed (moved to ONVIF Base Test specification)

Driving IP-based physical security through global standardization

Table of Contents

1 Introduction

The goal of the ONVIF test specification set is to make it possible to realize fully interoperable IP physical security implementation from different vendors. The set of ONVIF test specification describes the test cases needed to verify the [ONVIF Core Specs] and [ONVIF Conformance] requirements. In addition, the test cases are to be basic inputs for some Profile specification requirements. It also describes the test framework, test setup, pre-requisites, test policies needed for the execution of the described test cases.

This ONVIF Profile Q Test Specification acts as a supplementary document to the [ONVIF Core Specs], illustrating test cases need to be executed and passed. In addition, this specification acts as an input document to the development of test tool that will be used to test the ONVIF device implementation conformance towards ONVIF standard. This test tool is referred as ONVIF Client hereafter.

1.1 Scope

This ONVIF Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant devices. Conformance testing is meant to be functional black-box testing. The objective of this specification is to provide the test cases to test individual requirements of ONVIF devices according to ONVIF core services which are defined in [ONVIF Network Interface Specs].

The principal intended purposes are:

1. To provide self-assessment tool for implementations.
2. To provide comprehensive test suite coverage for [ONVIF Network Interface Specs].

This specification does not address the following.

1. Product use cases and non-functional (performance and regression) testing.
2. SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
3. Network protocol implementation Conformance test for HTTPS, HTTP, RTP and RTSP protocols.
4. Wi-Fi Conformance test.

The set of ONVIF Test Specification will not cover the complete set of requirements as defined in [ONVIF Network Interface Specs]; instead it will cover its subset.

This ONVIF Profile Q Test Specification covers the Profile Q conformant transition from a device (unsecure) status after a factory reset command has been executed up to ONVIF Default Access Policy is applied. The following sections describe the brief overview and scope of each functional block.

1.1.1 General

The General section covers the test cases needed for checking of Factory Default state and Operational state of the device.

The scope of this specification section is to cover the following functions:

- Turn device to Factory Default state with SetSystemFactoryDefault command
- Turn device to Operational state with creation of user with Administrator user level by CreateUsers and SetUser command

- Check scopes Factory Default state and Operational state

1.1.2 Default Access Policy

The Default Access Policy section covers the test cases needed for checking of oprations in the PRE_AUTH, READ_SYSTEM, READ_MEDIA, ACTUATE, UNRECOVERABLE, WRITE_SYSTEM and READ_SYSTEM_SECRET access classes.

1.1.3 System

The System section covers the test cases needed for ONVIF device firmware upgrade via HTTP.

The scope of this specification section is to cover the following functions:

- Upgrade of firmware with **StartFirmwareUpgrade** command
- Check ONVIF device state after firmware upgrade.

2 Terms and Definitions

2.1 Definitions

This section defines terms that are specific to the ONVIF Profile Q and tests. For a list of applicable general terms and definitions, please see [ONVIF Base Test].

2.2 Abbreviations

This section describes abbreviations used in this document.

DUT	Device Under Test
HTTP	Hyper Text Transport Protocol
RTCP	RTP Control Protocol
RTSP	Real Time Streaming Protocol
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
NTP	Network Time Protocol
UTC	Coordinated Universal Time

3 Test Overview

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

3.1 Test Setup

3.1.1 Network Configuration for DUT

The generic test configuration for the execution of test cases defined in this document is as shown below (Figure 1).

Based on the individual test case requirements, some of the entities in the below setup may not be needed for the execution of those corresponding test cases.

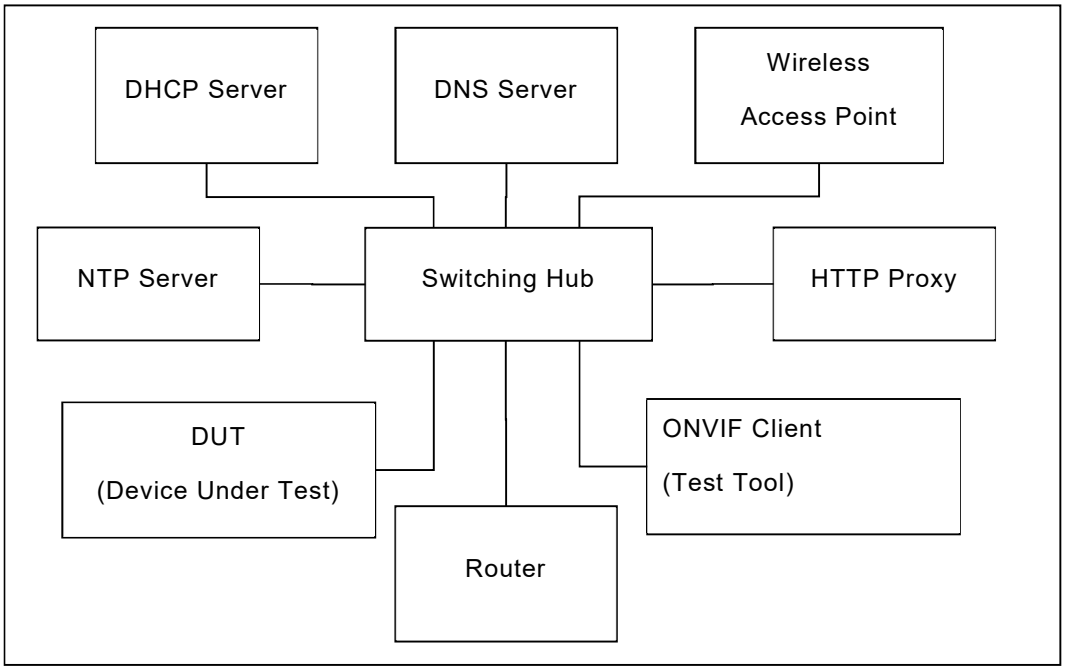


Figure 1: Test Configuration for DUT

DUT: ONVIF device to be tested. Hereafter, this is referred to as DUT (Device Under Test).

ONVIF Client (Test Tool): Tests are executed by this system, and it controls the behaviour of the DUT. It handles both expected and unexpected behaviour.

HTTP Proxy: provides facilitation in case of RTP and RTSP tunnelling over HTTP.

Wireless Access Point: provides wireless connectivity to the devices that support wireless connection.

DNS Server: provides DNS related information to the connected devices.

DHCP Server: provides IPv4 Address to the connected devices.

NTP Server: provides time synchronization between ONVIF Client and DUT.

3.2 Prerequisites

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

- The DUT shall be in out-of-the-box state.

Test Operator shall configure Reboot Timeout properly so that it would have enough time to reboot device for the following test cases for ONVIF Device Test Tool (see test description for more details):

  - 4.1.1 Factory default state verification
  - 4.3.1 Firmware Upload

3.3 Test Policy

This section describes the test policies specific to the test case execution of each functional block.

The DUT shall adhere to the test policies defined in this section.

3.3.1 General

The DUT shall give the DeviceService entry point and EventService entry point by GetServices command. Otherwise these test cases will be skipped.

- DUT shall support the following commands:
  - GetServices
  - GetServiceCapabilities
- DUT shall support HTTP Digest authentication
- DUT shall support set of user name with user name length equals to MaxUsernameLength
- DUT shall support set of user password with user password length equals to MaxPasswordLength
- DUT shall allow full anonymous access after hard SystemFactoryDefault
- DUT shall have onvif://www.onvif.org/Profile/Q/FactoryDefault scope after hard SystemFactoryDefault
- DUT shall not have onvif://www.onvif.org/Profile/Q/Operational scope after hard SystemFactoryDefault
- DUT shall have IPv4 network interface on DHCP after hard SystemFactoryDefault
- ZeroConfiguration shall be set to true after hard SystemFactoryDefault
- DUT shall conform to the access policy which is specified by Default Access Policy after create or set user with Administrator user level
- DUT shall have onvif://www.onvif.org/Profile/Q/Operational scope create or set user with Administrator user level
- DUT shall not have onvif://www.onvif.org/Profile/Q/FactoryDefault scope after create or

set user with Administrator user level

- The following tests are performed
    - Quick Install SetSystemFactoryDefault state verification

Please refer to Section 4.1 for General Test Cases.

3.3.2 Default Access Policy

The Default Access Policy section covers the test cases needed for check of access policy on the DUT.

- The DUT shall support Default Access Policy
- DUT shall conform to the access policy which is specified by Default Access Policy
- The DUT shall support HTTP Digest Authentication
- The following tests are performed
    - Default Access Policy – Anonymous
    - Default Access Policy – User
    - Default Access Policy – Operator
    - Default access policy - Administrator and Anonymous
    - Default access policy - Administrator and User/Operator

Please refer to Section 4.2 for Default Access Policy Test Cases.

3.3.3 System

The System section covers the test cases with firmware upgrade via HTTP.

- DUT shall support HttpFirmwareUpgrade.
- DUT shall returns HTTP 415 message in case invalid firmware file.
- The following tests are performed
    - Firmware Upgrade via HTTP with valid firmware file
    - Firmware Upgrade via HTTP with invalid firmware file

Please refer to Section 4.3 for System Test Cases.

4 Quick Install Test Cases

4.1 General

4.1.1 Factory default state verification

Test Label: Quick Install SetSystemFactoryDefault state verification

Test Case ID: QUICK_INSTALL-1-1-1

ONVIF Core Specification Coverage: Factory Default, CreateUsers (ONVIF Core Specification), SetUser (ONVIF Core Specification)

Command Under Test: SetSystemFactoryDefault, CreateUsers, SetUsers

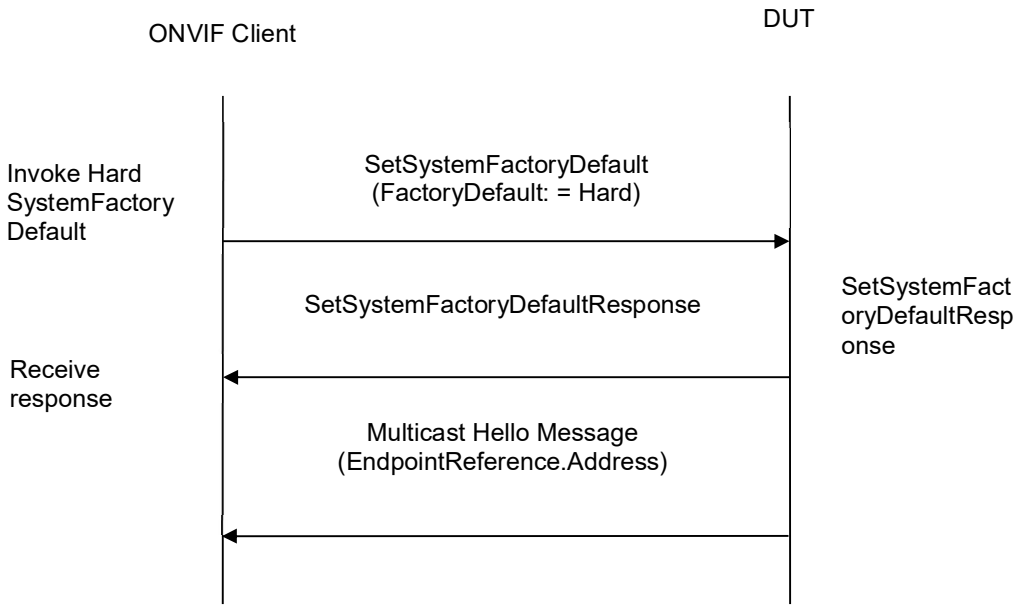
WSDL Reference: devicemgmt.wsdl

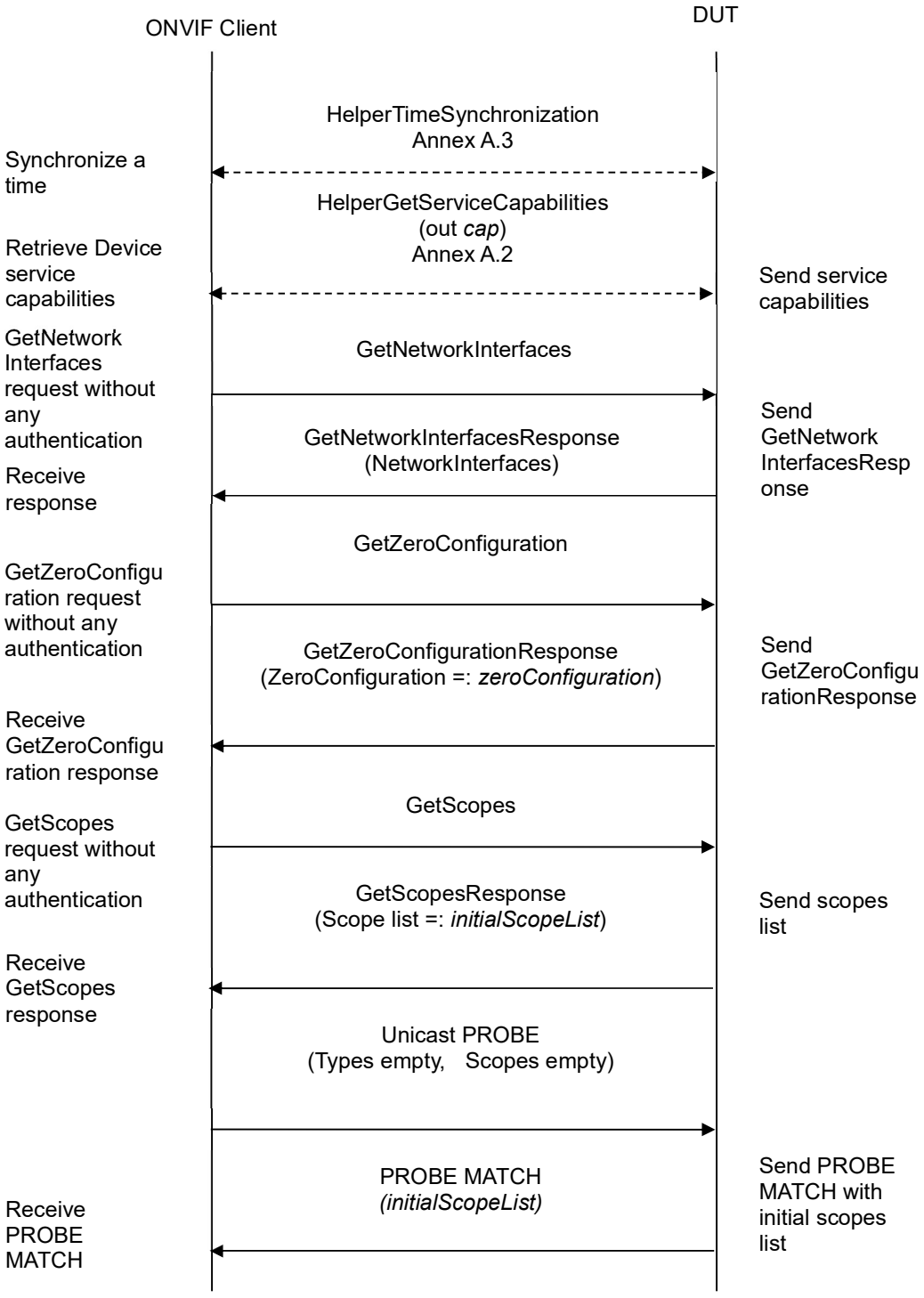
Test Purpose: To verify transition from FactoryDefault state to Operational state.

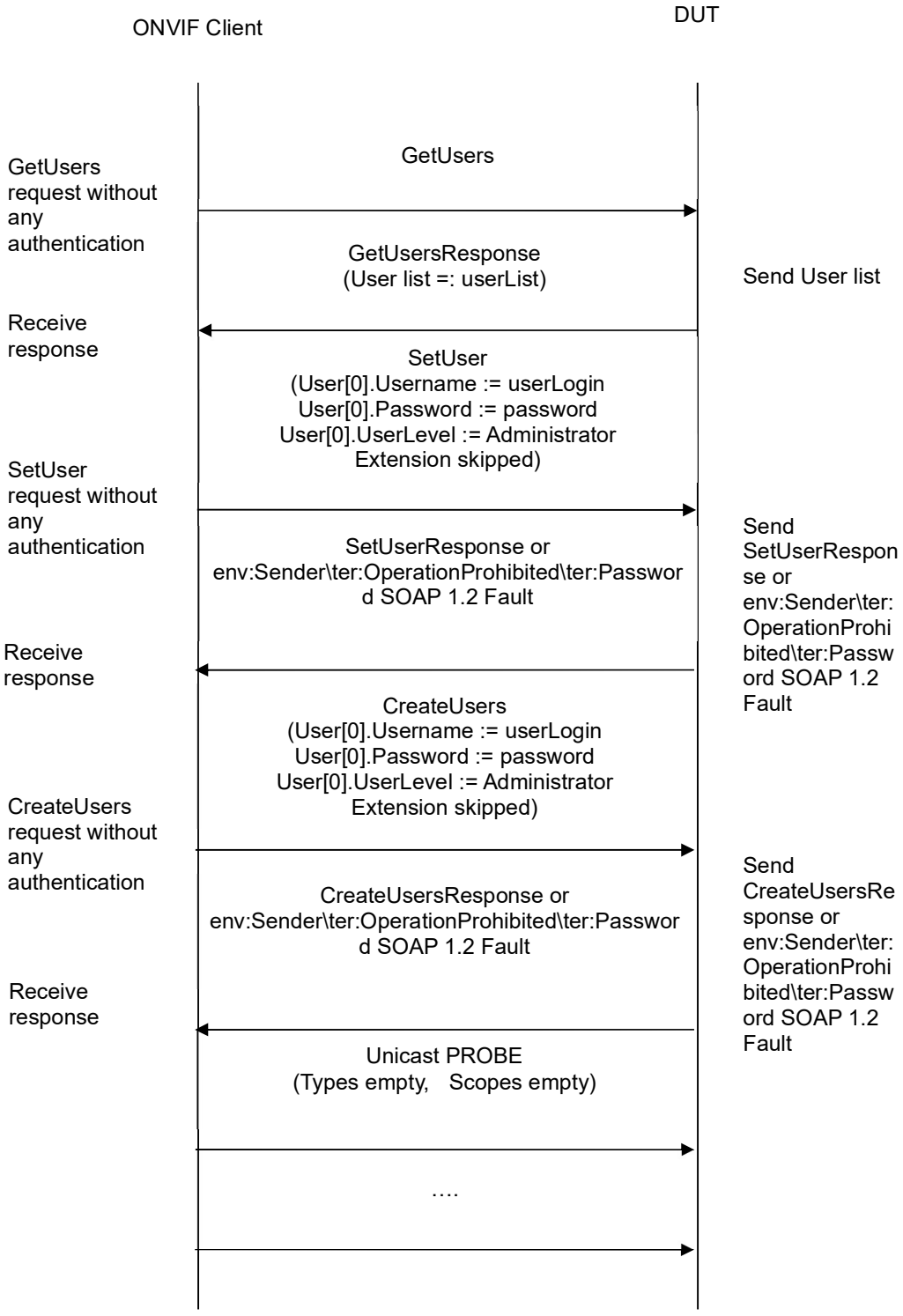
Pre-requisite: GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. ZeroConfiguration is supported by the DUT as indicated by the Network.ZeroConfiguration capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength.

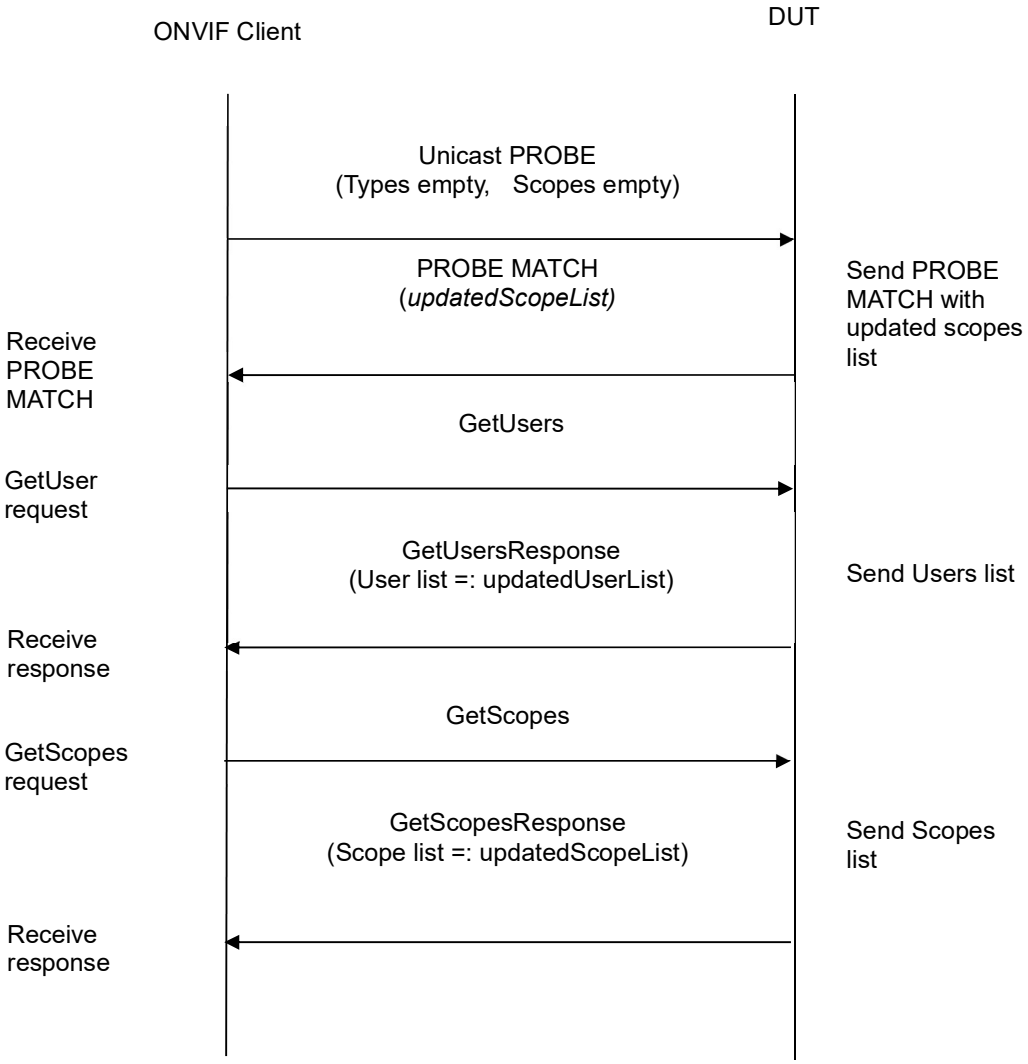
Test Configuration: ONVIF Client and DUT

Test Sequence:




ONVIF Client
DUT
HelperTimeSynchronization
Annex A.3
Synchronize a time
HelperGetServiceCapabilities
(out *cap*)
Annex A.2
Retrieve Device service capabilities
Send service capabilities
GetNetwork Interfaces request without any authentication
GetNetworkInterfaces
Send GetNetwork InterfacesResponse
GetNetworkInterfacesResponse
(NetworkInterfaces)
Receive response
GetZeroConfiguration
GetZeroConfiguration request without any authentication
Send GetZeroConfigurationResponse
GetZeroConfigurationResponse
(ZeroConfiguration =: *zeroConfiguration*)
Receive GetZeroConfiguration response
GetScopes
GetScopes request without any authentication
Send scopes list
GetScopesResponse
(Scope list =: *initialScopeList*)
Receive GetScopes response
Unicast PROBE
(Types empty, Scopes empty)
Send PROBE MATCH with initial scopes list
PROBE MATCH
(initialScopeList)
Receive PROBE MATCH


ONVIF Client
DUT
GetUsers request without any authentication
GetUsers
GetUsersResponse (User list =: userList)
Send User list
Receive response
SetUser (User[0].Username := userLogin User[0].Password := password User[0].UserLevel := Administrator Extension skipped)
SetUser request without any authentication
SetUserResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Send SetUserResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Receive response
CreateUsers (User[0].Username := userLogin User[0].Password := password User[0].UserLevel := Administrator Extension skipped)
CreateUsers request without any authentication
CreateUsersResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Send CreateUsersResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Receive response
Unicast PROBE (Types empty, Scopes empty)
....



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **SetSystemFactoryDefault** with parameters
   - FactoryDefault := Hard
4. DUT responds with a **SetSystemFactoryDefaultResponse** message.
5. Until *timeout1* timeout expires, repeat the following steps:

   5.1. The DUT will send Multicast **Hello** message after it is successfully rebooted with parameters:

- EndpointReference.Address equal to unique endpoint reference of the DUT
- Types list
- Scopes list := *scopesList*
- XAddrs list := *xaddrsList*
- MetadataVersion

5.2. If *xaddrsList* contains URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, go to step 7.

6. If *timeout1* timeout expires for step 5 without Hello with URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, FAIL the test and skip other steps.
7. ONVIF client waits for 5 seconds after Hello was received.
8. ONVIF client checks the following:
   - If *scopesList* does not contain " onvif://www.onvif.org/Profile/Q/FactoryDefault" scope, FAIL the test and skip other steps.
   - If *scopesList* contains " onvif://www.onvif.org/Profile/Q/Operational" scope, FAIL the test and skip other steps.
9. ONVIF Client synchronize a time between ONVIF Client and DUT by following the procedure mentioned in Annex A.3.
10. ONVIF Client gets the service capabilities without any authentication (out *cap*) by following the procedure mentioned in Annex A.2.
11. ONVIF client invokes **GetNetworkInterfaces** without any authentication.
12. The DUT responds with **GetNetworkInterfacesResponse** message with parameters
    - NetworkInterfaces.token1 =: *currentNetInt*, where token1 is token of the currently used network interface
13. If *currentNetInt*.IPv4.DHCP is not equal to true, FAIL the test and skip other steps.
14. If *cap* contains Network.IPVersion6 item and *cap*.Network.IPVersion6 equals to true:
    - If *currentNetInt*.IPv6.Enabled is not equal to true, FAIL the test and skip other steps.
    - If *currentNetInt*.IPv6.Config does not contain at least one LinkLocal item with not empty Address item, FAIL the test and skip other steps.
15. ONVIF client invokes **GetZeroConfiguration** without any authentication.
16. The DUT responds with **GetZeroConfigurationResponse** message with parameters
    - ZeroConfiguration =: *zeroConfiguration*
17. If *zeroConfiguration*.Enabled is not equal to true, FAIL the test.
18. ONVIF Client invokes **GetScopes** without any authentication.

19. DUT responds with a **GetScopesResponse** message with parameters
    - Scope list =: *initialScopeList*
20. If *initialScopeList* does not contain “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test.
21. If *initialScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test.
22. ONVIF Client invokes Unicast **PROBE** message with the following parameters
    - Types empty
    - Scopes empty
23. DUT responds with a **PROBE MATCH** message with parameters:
    - ProbeMatch.Scopes
24. Set the following:
    - *initialScopeList* := ProbeMatch.Scopes
25. If *initialScopeList* does not contain “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.
26. If *initialScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test and skip other steps.
27. If *cap* does not contain Security.MaxPasswordLength or Security.MaxUserNameLength, FAIL the test and skip other steps.
28. ONVIF Client invokes **GetUsers** without any authentication.
29. DUT responds with a **GetUsersResponse** message with parameters.
    - User list =: *userList*
30. If *userList* contains user with user level Administrator:

    30.1. Set the following:
    - passwordLength := *cap*.Security.MaxPasswordLength
    - *userLogin* := Username of user with user level equal to Administrator from *userList*
    - *password* := random string, contains *passwordLength* ASCII characters

    30.2. ONVIF Client invokes **SetUser** with parameters
    - User[0].Username := *userLogin*
    - User[0].Password := *password*
    - User[0].UserLevel := Administrator
    - Extension skipped

30.3. If the DUT responds with **SetUserResponse** message, go to 32.

30.4. If DUT returns env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 fault:

30.4.1. Set the following:

- *password* := random string, contains *passwordLength* ASCII characters

30.4.2. Go to the step 30.2.

30.5. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.

31. If *userList* does not contain user with user level Administrator:

31.1. Set the following:

- *userLoginLength* := *cap*.Security.MaxUserNameLength
- *passwordLength* := *cap*.Security.MaxPasswordLength
- *userLogin* := random string, contains *userLoginLength* low case alphbet characters, differs from usernames listed in *userList*
- *password* := random string, contains *passwordLength* ASCII characters

31.2. ONVIF Client invokes **CreateUsers** with parameters

- User[0].Username := *userLogin*
- User[0].Password := *password*
- User[0].UserLevel := Administrator
- Extension skipped

31.3. If the DUT responds with **CreateUsersResponse** message, go to 32.

31.4. If the DUT returns env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 fault:

31.4.1. Set the following:

- *password* := random string, contains *passwordLength* ASCII characters

31.4.2. Go to the step 31.2.

31.4.3. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.

32. ONVIF Client waits the Reboot Timeout.

33. ONVIF Client sends PROBE message and if DUT responds with PROBE MATCH mesage then go to the step 35

34. ONVIF Client waits for Hello message sent from newly configured address by the DUT. Then ONVIF Client starts using this newly configured address for further communications with DUT.

35. Until *timeout1* expires, repeat the following steps:

35.1. ONVIF Client invokes Unicast **PROBE** message with the following parameters

- Types empty
- Scopes empty

35.2. If the DUT responds with **PROBE MATCH** message:

35.2.1. Set the following:

- *updatedScopeList* := ProbeMatch.Scopes

35.2.2. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/Operational” scope, go to the step 35.

35.3. If *timeout1* timeout expires for step 35 without **PROBE MATCH** message with “onvif://www.onvif.org/Profile/Q/Operational” scope in *updatedScopeList,* FAIL the test and skip other steps.

36. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.

37. ONVIF Client invokes **GetUsers** with user with the user level Administrator credantials (*userLogin* and *password*).

38. DUT responds with a **GetUsersResponse** message with parameters.

- User list =: *updatedUserList*

39. If *updatedUserList* does not contain Username = *userLogin* with UserLevel = Administrator, FAIL the test and skip other steps.

40. ONVIF Client invokes **GetScopes** with the user level Administrator credantials (*userLoginUser* and *passwordUser*)

41. The DUT responds with a **GetScopesResponse** message with parameters

- Scope list =: *updatedScopeList*

42. If *updatedScopeList* contains “onvif://www.onvif.org/Profile/Q/FactoryDefault” scope, FAIL the test and skip other steps.

43. If *updatedScopeList* does not contain “onvif://www.onvif.org/Profile/Q/Operational” scope, FAIL the test and skip other steps.

Procedure Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **SetSystemFactoryDefaultResponse** message.

The DUT did not send **GetUsersResponse** message.

The DUT did not send **SetUsersResponse** message.

The DUT did not send **CreateUsersResponse** message.

The DUT did not send **GetZeroConfigurationResponse** message.

The DUT did not send **GetNetworkInterfacesResponse** message.

The DUT did not send **GetScopesResponse** message.

The DUT did not allow Anonymous access to the **GetNetworkInterfaces** command.

The DUT did not allow Anonymous access to the **GetZeroConfiguration** command.

The DUT did not allow Anonymous access to the **GetUsers** command before changing the user with the user level Administrator credantials.

The DUT did not allow Anonymous access to the **SetUsers** command before changing the user with the user level Administrator credantials.

The DUT did not allow Anonymous access to the **CreateUsers** command before changing the user with the user level Administrator credantials.

The DUT did not allow Anonymous access to the **GetScopes** command before changing the user with the user level Administrator credantials.

The DUT allowed Anonymous access to the **GetUsers** command after changing the user with the user level Administrator credantials.

The DUT allowed Anonymous access to the **GetScopes** command after changing the user with the user level Administrator credantials.

Note: User with username *userLogin* and password *password* shall be used for further test cases.

Note: *timeout1* will be taken from Reboot Timeout field of ONVIF Device Test Tool.

Note: IPv4 address from Hello shall be used for further test cases.

4.2 Default access policy

4.2.1 Default access policy - Anonymous

Test Label: Default Access Policy - Anonymous

Test Case ID: QUICK_INSTALL-2-1-1

ONVIF Core Specification Coverage: Default access policy

Command Under Test: GetServices, GetServiceCapabilities, GetHostname, GetSystemDateAndTime

WSDL Reference: devicemgmt.wsdl

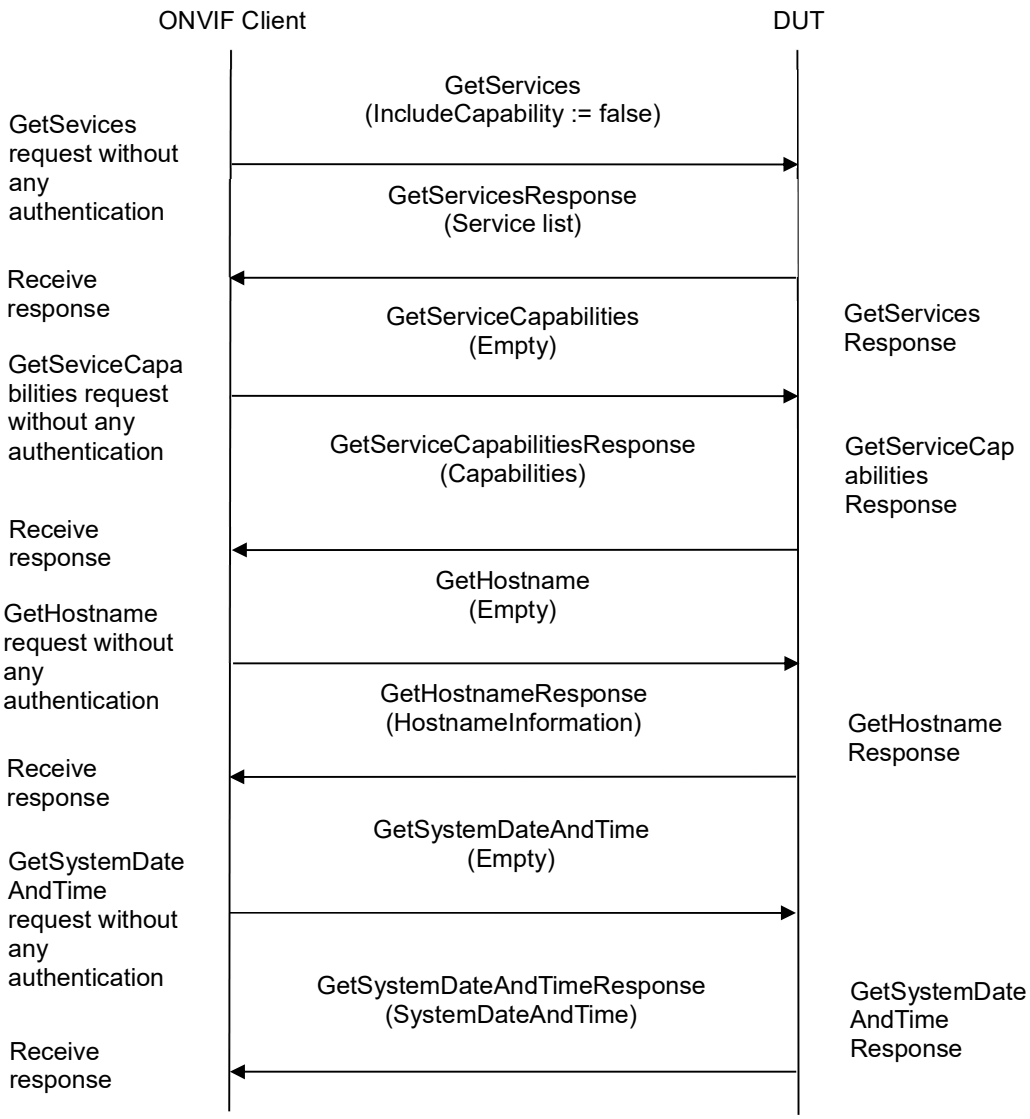
Test Purpose: To verify that operations in the PRE_AUTH access class can be accessed without authentication being required.

Pre-requisite: GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication

is supported by the DUT as indicated by the Security.HttpDigest capability. Default Access Policy is not modified.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF client invokes **GetServices** for Device service without any authentication with parameters

- IncludeCapability := false

4. The DUT responds with **GetServicesResponse** message with parameters
   - Service list
5. ONVIF client invokes **GetServiceCapabilities** for Device Service without any authentication.
6. The DUT responds with **GetServiceCapabilitiesResponse** message with parameters
   - Capabilities
7. ONVIF client invokes **GetHostname** without any authentication.
8. The DUT responds with **GetHostnameResponse** message with parameters
   - HostnameInformation
9. ONVIF client invokes **GetSystemDateAndTime** without any authentication.
10. The DUT responds with **GetSystemDateAndTimeResponse** message with parameters
    - SystemDateAndTime

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not allow Anonymous access to the **GetServices** command.

The DUT did not allow Anonymous access to the **GetServiceCapabilities** command.

The DUT did not allow Anonymous access to the **GetHostname** command.

The DUT did not allow Anonymous access to the **GetSystemDateAndTime** command.

The DUT did not send **GetServicesResponse** message.

The DUT did not send **GetServiceCapabilitiesResponse** message.

The DUT did not send **GetHostnameResponse** message.

The DUT did not send **GetSystemDateAndTimeResponse** message.

4.2.2 Default access policy - User

Test Label: Default Access Policy - User

Test Case ID: QUICK_INSTALL-2-1-2

ONVIF Core Specification Coverage: Default access policy

Command Under Test: GetNTP, GetNetworkInterfaces, GetScopes, GetDiscoveryMode, GetEventProperties

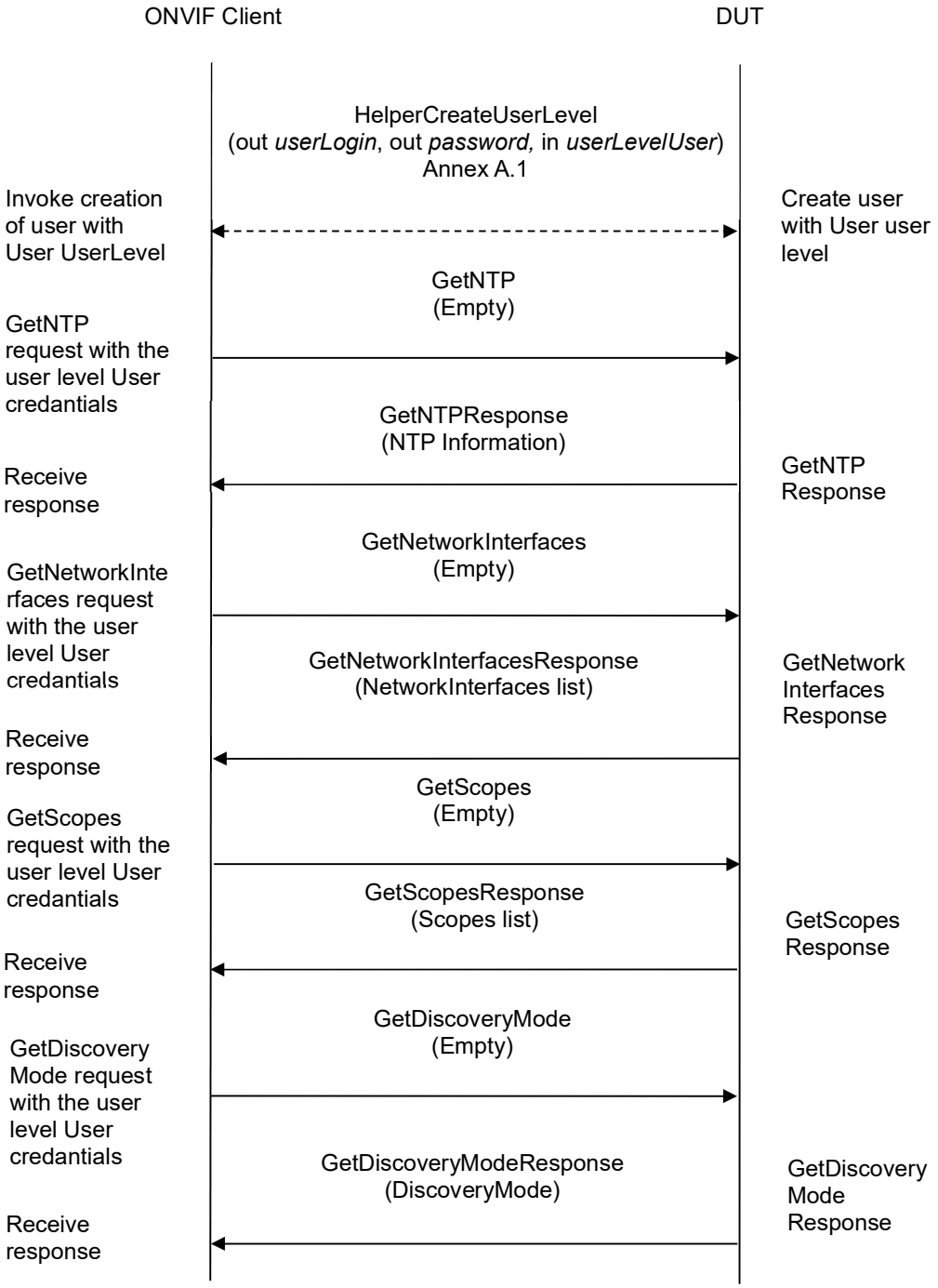
WSDL Reference: devicemgmt.wsdl

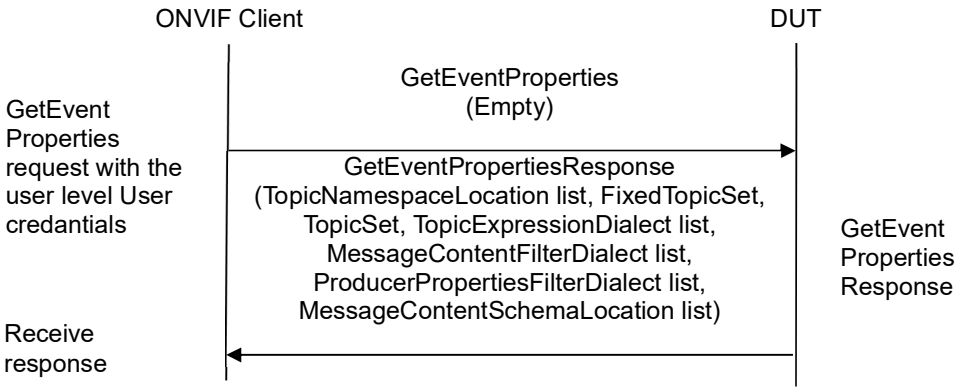
Test Purpose: To verify that operations in the READ_SYSTEM and READ_MEDIA access classes can be accessed with authentication level User.

Pre-requisite: GetServices command is supported by the DUT. Event Service was received from the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security**.**MaxUsernameLength**,** Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security**.**MaxPasswordLength. Default Access Policy is not modified.

Test Configuration: ONVIF Client and DUT

Test Sequence:


ONVIF Client
DUT
HelperCreateUserLevel
(out userLogin, out password, in userLevelUser)
Annex A.1
Invoke creation of user with User UserLevel
Create user with User user level
GetNTP
(Empty)
GetNTP request with the user level User credantials
GetNTPResponse
(NTP Information)
Receive response
GetNTP Response
GetNetworkInterfaces
(Empty)
GetNetworkInterfaces request with the user level User credantials
GetNetworkInterfacesResponse
(NetworkInterfaces list)
GetNetwork Interfaces Response
Receive response
GetScopes
(Empty)
GetScopes request with the user level User credantials
GetScopesResponse
(Scopes list)
GetScopes Response
Receive response
GetDiscoveryMode
(Empty)
GetDiscovery Mode request with the user level User credantials
GetDiscoveryModeResponse
(DiscoveryMode)
GetDiscovery Mode Response
Receive response



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. Set the following:
   - *userLevel* := User
4. ONVIF Client generates creates user with predefined user level (in *userLevel*) and user login (out *userLogin*) and password (out *password*) by following the procedure mentioned in Annex A.1.
5. If the DUT supports NTP as indicated by Network.NTP capability:
   5.1. ONVIF client invokes **GetNTP** without any authentication.
   5.2. The DUT responds with **HTTP 401 Unauthorized** error.
   5.3. ONVIF client invokes **GetNTP** with user with the user level User credantials (*userLogin* and *password*).
   5.4. The DUT responds with **GetNTPResponse** message with parameters
      - NTPInformation
6. ONVIF client invokes **GetNetworkInterfaces** without any authentication.
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. ONVIF client invokes **GetNetworkInterfaces** with user with the user level User credantials (*userLogin* and *password*).
9. The DUT responds with **GetNetworkInterfacesResponse** message with parameters
   - NetworkInterfaces list

10. ONVIF client invokes **GetScopes** without any authentication.
11. The DUT responds with **HTTP 401 Unauthorized** error.
12. ONVIF client invokes **GetScopes** with user with the user level User credantials (*userLogin* and *password*).
13. The DUT responds with **GetScopesResponse** message with parameters
    - Scopes list
14. ONVIF client invokes **GetDiscoveryMode** without any authentication.
15. The DUT responds with **HTTP 401 Unauthorized** error.
16. ONVIF client invokes **GetDiscoveryMode** with user with the user level User credantials (*userLogin* and *password*).
17. The DUT responds with **GetDiscoveryModeResponse** message with parameters
    - DiscoveryMode
18. ONVIF client invokes **GetEventProperties** without any authentication.
19. The DUT responds with **HTTP 401 Unauthorized** error.
20. ONVIF client invokes **GetEventProperties** with user with the user level User credantials (*userLogin* and *password*).
21. The DUT responds with **GetEventPropertiesResponse** message with parameters
    - TopicNamespaceLocation list
    - FixedTopicSet
    - TopicSet
    - TopicExpressionDialect list
    - MessageContentFilterDialect list
    - ProducerPropertiesFilterDialect list
    - MessageContentSchemaLocation list
    - Other additional fields if any

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT allow Anonymous access to the **GetNTP** command.

The DUT allow Anonymous access to the **GetNetworkInterfaces** command.

The DUT allow Anonymous access to the **GetScopes** command.

The DUT allow Anonymous access to the **GetDiscoveryMode** command.

The DUT allow Anonymous access to the **GetEventProperties** command.

The DUT did not allow user with the user level User access to the **GetNTP** command.

The DUT did not allow user with the user level User access to the **GetNetworkInterfaces** command.

The DUT did not allow user with the user level User access to the **GetScopes** command.

The DUT did not allow user with the user level User access to the **GetDiscoveryMode** command.

The DUT did not allow user with the user level User access to the **GetEventProperties** command.

The DUT did not send **GetNTPResponse** message.

The DUT did not send **GetNetworkInterfacesResponse** message.

The DUT did not send **GetScopesResponse** message.

The DUT did not send **GetDiscoveryModeResponse** message.

The DUT did not send **GetEventPropertiesResponse** message.

4.2.3 Default access policy - Administrator and Anonymous

Test Label: Default Access Policy - Administrator And Anonymous

Test Case ID: QUICK_INSTALL-2-1-4

ONVIF Core Specification Coverage: Default access policy

Command Under Test: SetScopes, SetDiscoveryMode, GetAccessPolicy, CreateUsers, SetSystemDateAndTime

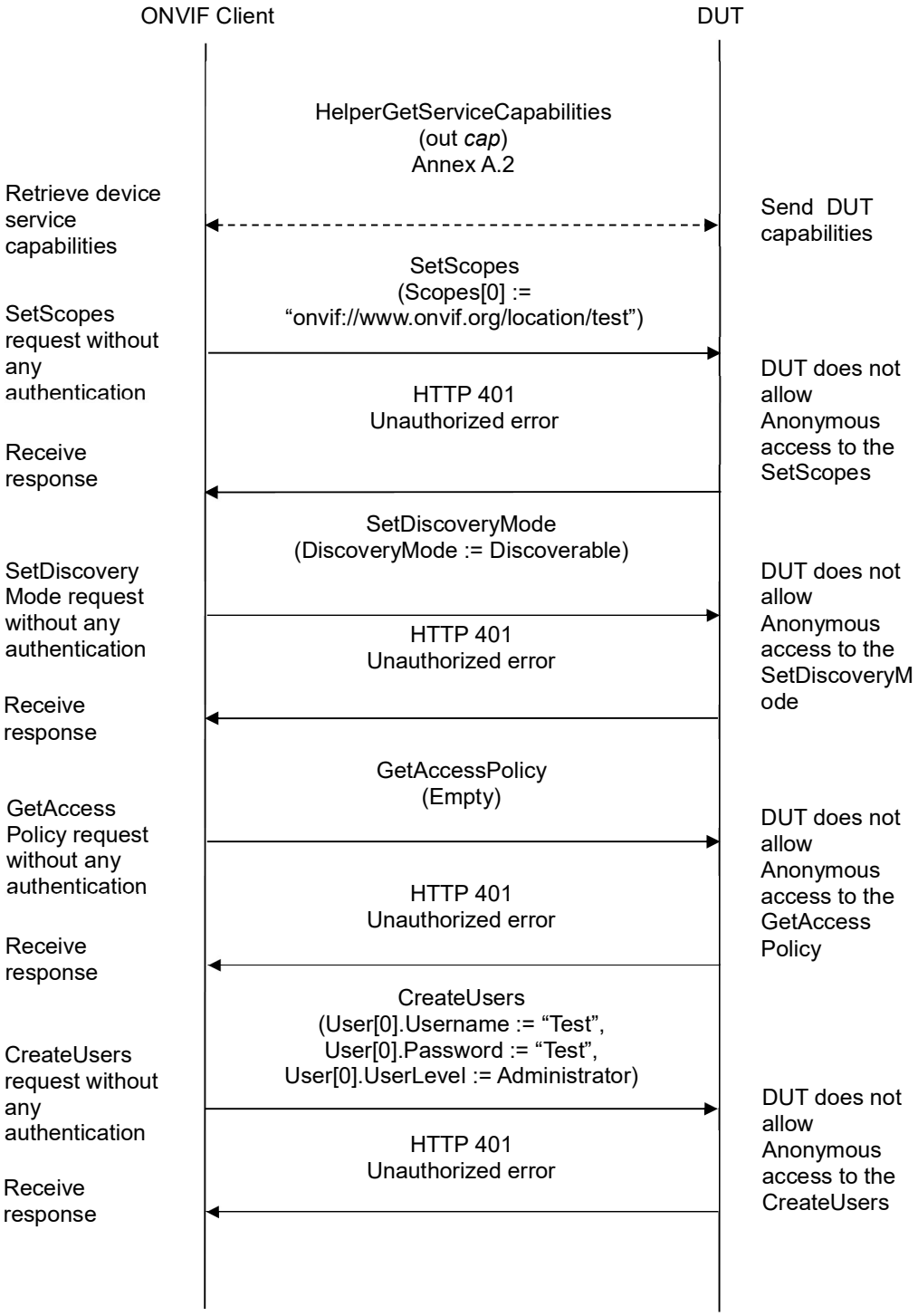
WSDL Reference: devicemgmt.wsdl

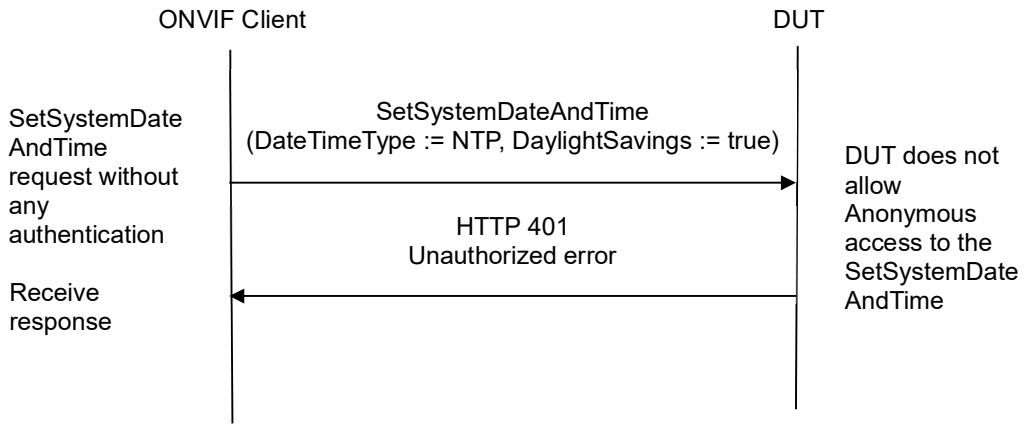
Test Purpose: To verify that operations in the UNRECOVERABLE, WRITE_SYSTEM and READ_SYSTEM_SECRET access classes can notbe accessed without authentication level Administrator.

Pre-requisite: GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Default Access Policy is not modified.

Test Configuration: ONVIF Client and DUT

Test Sequence:


ONVIF Client
DUT
HelperGetServiceCapabilities
(out cap)
Annex A.2
Retrieve device service capabilities
Send DUT capabilities
SetScopes
(Scopes[0] :=
“onvif://www.onvif.org/location/test”)
SetScopes request without any authentication
DUT does not allow Anonymous access to the SetScopes
HTTP 401
Unauthorized error
Receive response
SetDiscoveryMode
(DiscoveryMode := Discoverable)
SetDiscovery Mode request without any authentication
DUT does not allow Anonymous access to the SetDiscoveryMode
HTTP 401
Unauthorized error
Receive response
GetAccessPolicy
(Empty)
GetAccess Policy request without any authentication
DUT does not allow Anonymous access to the GetAccess Policy
HTTP 401
Unauthorized error
Receive response
CreateUsers
(User[0].Username := “Test”,
User[0].Password := “Test”,
User[0].UserLevel := Administrator)
CreateUsers request without any authentication
DUT does not allow Anonymous access to the CreateUsers
HTTP 401
Unauthorized error
Receive response



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.2.
4. ONVIF client invokes **SetScopes** without any authentication with parameters
   - Scopes[0] := “onvif://www.onvif.org/location/test”
5. The DUT responds with **HTTP 401 Unauthorized** error.
6. ONVIF client invokes **SetDiscoveryMode** without any authentication with parameters
   - DiscoveryMode := Discoverable
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. If *cap*.Security contains AccessPolicyConfig and *cap*.Security.AccessPolicyConfig equals to true:

   8.1. ONVIF client invokes **GetAccessPolicy** without any authentication.

   8.2. The DUT responds with **HTTP 401 Unauthorized** error.
9. ONVIF client invokes **CreateUsers** without any authentication with parameters
   - User[0].Username := “Test”
   - User[0].Password := “Test”
   - User[0].UserLevel := Administrator
   - Extension skipped
10. The DUT responds with **HTTP 401 Unauthorized** error.

11. ONVIF client invokes **SetSystemDateAndTime** without any authentication with parameters
    - DateTimeType := NTP
    - DaylightSavings := true
    - TimeZone skipped
    - UTCDateTime skipped
12. The DUT responds with **HTTP 401 Unauthorized** error.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT allowed Anonymous access to the **SetScopes** command.

The DUT allowed Anonymous access to the **SetDiscoveryMode** command.

The DUT allowed Anonymous access to the **GetAccessPolicy** command.

The DUT allowed Anonymous access to the **CreateUsers** command.

The DUT allowed Anonymous access to the **SetSystemDateAndTime** command.

4.2.4 Default access policy - Administrator and User/Operator

Test Label: Default Access Policy - Administrator And User/Operator

Test Case ID: QUICK_INSTALL-2-1-5

ONVIF Core Specification Coverage: Default access policy

Command Under Test: SetScopes, SetDiscoveryMode, GetAccessPolicy, CreateUsers, SetSystemDateAndTime

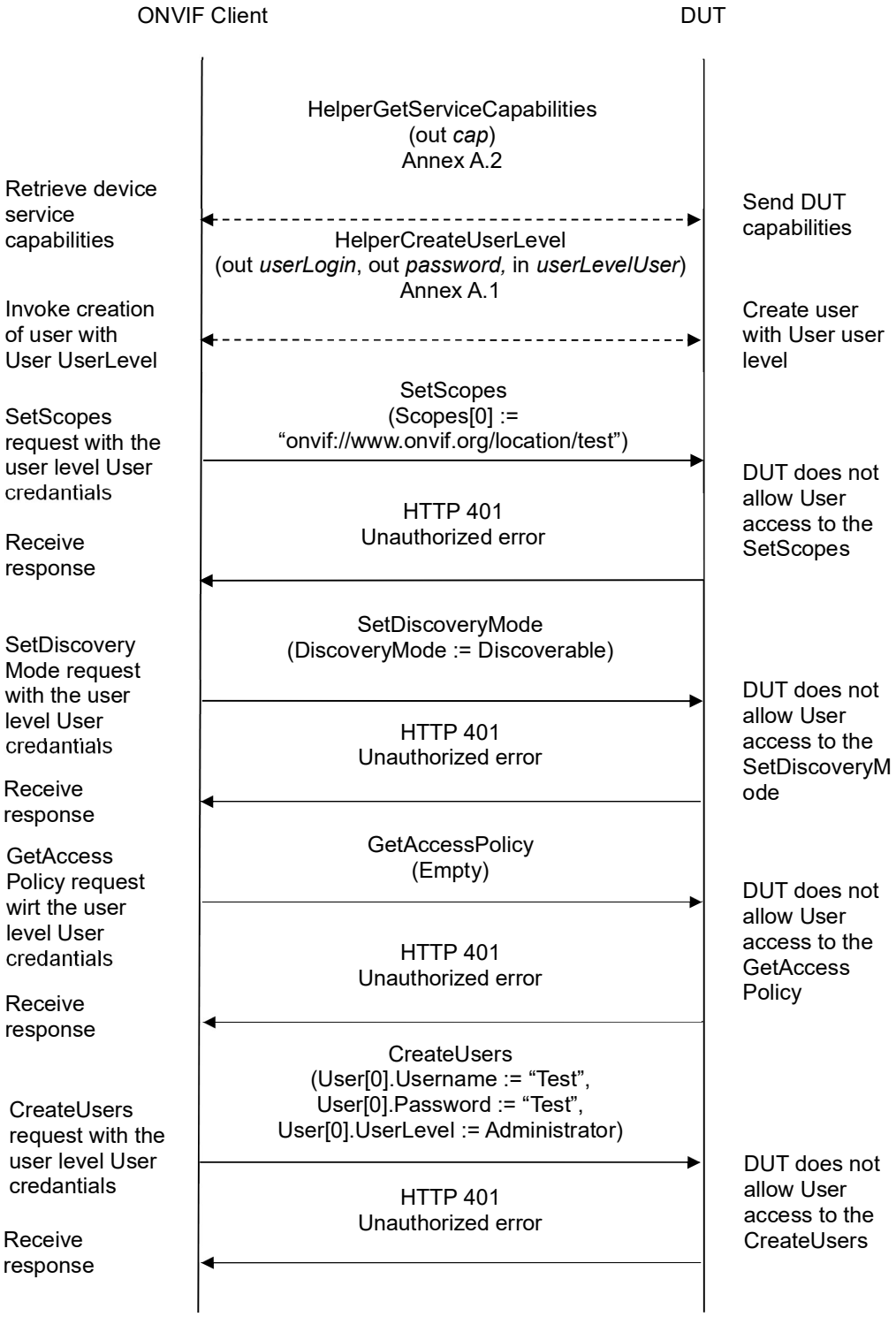
WSDL Reference: devicemgmt.wsdl

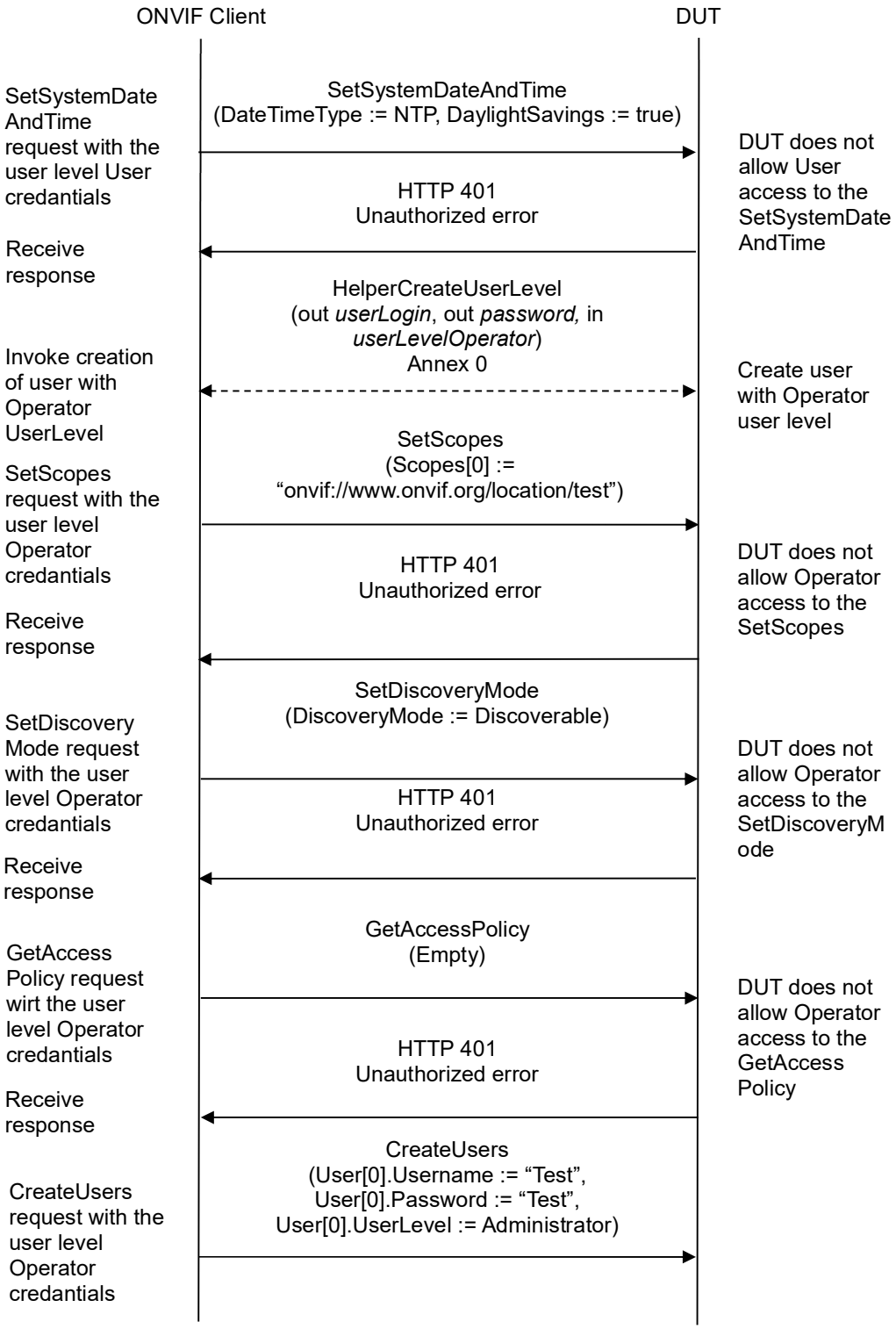
Test Purpose: To verify that operations in the UNRECOVERABLE, WRITE_SYSTEM and READ_SYSTEM_SECRET access classes can not be accessed without authentication level Administrator.

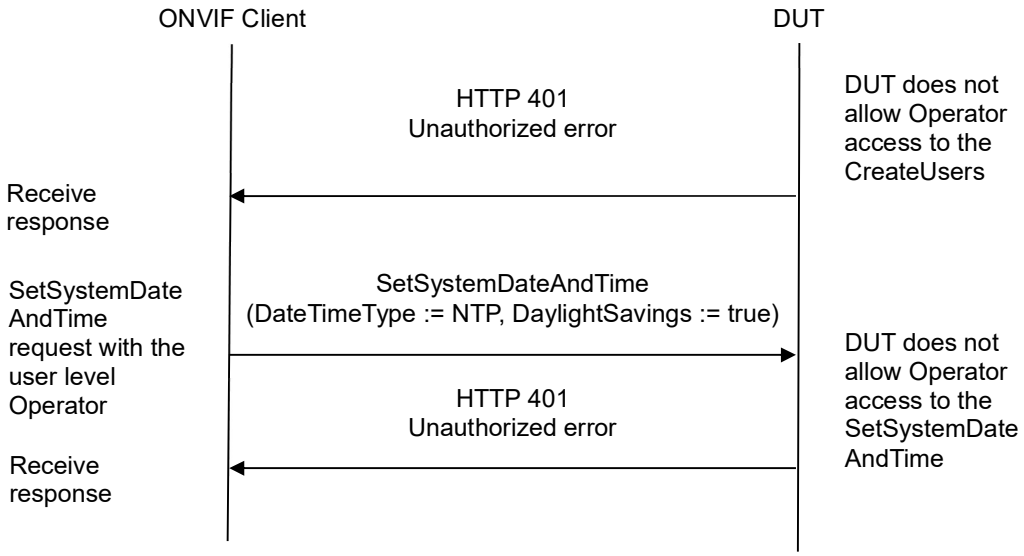
Pre-requisite: GetServices command is supported by the DUT. Default Access Policy is supported by the DUT as indicated by the Security.DefaultAccessPolicy capability. HTTP Digest Authentication is supported by the DUT as indicated by the Security.HttpDigest capability. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security**.**MaxUsernameLength**,** Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security**.**Max PasswordLength.Default Access Policy is not modified.

Test Configuration: ONVIF Client and DUT

Test Sequence:


ONVIF Client
DUT
HelperGetServiceCapabilities
(out *cap*)
Annex A.2
Retrieve device service capabilities
Send DUT capabilities
HelperCreateUserLevel
(out *userLogin*, out *password,* in *userLevelUser*)
Annex A.1
Invoke creation of user with User UserLevel
Create user with User user level
SetScopes
(Scopes[0] :=
“onvif://www.onvif.org/location/test”)
SetScopes request with the user level User credantials
DUT does not allow User access to the SetScopes
HTTP 401
Unauthorized error
Receive response
SetDiscoveryMode
(DiscoveryMode := Discoverable)
SetDiscovery Mode request with the user level User credantials
DUT does not allow User access to the SetDiscoveryMode
HTTP 401
Unauthorized error
Receive response
GetAccessPolicy
(Empty)
GetAccess Policy request wirt the user level User credantials
DUT does not allow User access to the GetAccess Policy
HTTP 401
Unauthorized error
Receive response
CreateUsers
(User[0].Username := “Test”,
User[0].Password := “Test”,
User[0].UserLevel := Administrator)
CreateUsers request with the user level User credantials
DUT does not allow User access to the CreateUsers
HTTP 401
Unauthorized error
Receive response


ONVIF Client
DUT
SetSystemDate AndTime request with the user level User credantials
SetSystemDateAndTime (DateTimeType := NTP, DaylightSavings := true)
DUT does not allow User access to the SetSystemDate AndTime
HTTP 401 Unauthorized error
Receive response
HelperCreateUserLevel (out userLogin, out password, in userLevelOperator) Annex 0
Invoke creation of user with Operator UserLevel
Create user with Operator user level
SetScopes (Scopes[0] := "onvif://www.onvif.org/location/test")
SetScopes request with the user level Operator credantials
DUT does not allow Operator access to the SetScopes
HTTP 401 Unauthorized error
Receive response
SetDiscoveryMode (DiscoveryMode := Discoverable)
SetDiscovery Mode request with the user level Operator credantials
DUT does not allow Operator access to the SetDiscoveryMode
HTTP 401 Unauthorized error
Receive response
GetAccessPolicy (Empty)
GetAccess Policy request wirt the user level Operator credantials
DUT does not allow Operator access to the GetAccess Policy
HTTP 401 Unauthorized error
Receive response
CreateUsers (User[0].Username := "Test", User[0].Password := "Test", User[0].UserLevel := Administrator)
CreateUsers request with the user level Operator credantials



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. Set the following:
   - *userLevelUser* := User
   - *userLevelOperator*:= Operator
4. ONVIF Client generates creates user with predefined user level (in *userLevelUser*) and user login (out *userLoginUser*) and password (out *passwordUser*) by following the procedure mentioned in Annex A.1.
5. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.2.
6. ONVIF client invokes **SetScopes** with user with the user level User credantials (*userLoginUser* and *passwordUser*) with parameters
   - Scopes[0] := “onvif://www.onvif.org/location/test”
7. The DUT responds with **HTTP 401 Unauthorized** error.
8. ONVIF client invokes **SetDiscoveryMode** with user with the user level User credantials (*userLoginUser* and *passwordUser*) with parameters
   - DiscoveryMode := Discoverable

9. The DUT responds with **HTTP 401 Unauthorized** error.
10. If *cap*.Security contains AccessPolicyConfig and *cap*.Security.AccessPolicyConfig equals to true:
    10.1. ONVIF client invokes **GetAccessPolicy** with user with the user level User credantials (*userLoginUser* and *passwordUser*).
    10.2. The DUT responds with **HTTP 401 Unauthorized** error.
11. ONVIF client invokes **CreateUsers** with user with the user level User credantials (*userLoginUser* and *passwordUser*) with parameters
    - User[0].Username := “Test”
    - User[0].Password := “Test”
    - User[0].UserLevel := Administrator
    - Extension skipped
12. The DUT responds with **HTTP 401 Unauthorized** error.
13. ONVIF client invokes **SetSystemDateAndTime** with user with the user level User credantials (*userLoginUser* and *passwordUser*) with parameters
    - DateTimeType := NTP
    - DaylightSavings := true
    - TimeZone skipped
    - UTCDateTime skipped
14. The DUT responds with **HTTP 401 Unauthorized** error.
15. ONVIF Client generates creates user with predefined user level (in *userLevelOperator*) and user login (out *userLoginOperator*) and password (out *passwordOperator*) by following the procedure mentioned in Annex 0.
16. ONVIF client invokes **SetScopes** with user with the user level Operator credantials (*userLoginOperator* and *passwordOperator*) with parameters
    - Scopes[0] := “onvif://www.onvif.org/location/test”
17. The DUT responds with **HTTP 401 Unauthorized** error.
18. ONVIF client invokes **SetDiscoveryMode** with user with the user level Operator credantials (*userLoginOperator* and *passwordOperator*) with parameters
    - DiscoveryMode := Discoverable
19. The DUT responds with **HTTP 401 Unauthorized** error.
20. If *cap*.Security contains AccessPolicyConfig and *cap*.Security.AccessPolicyConfig equals to true:
    20.1. ONVIF client invokes **GetAccessPolicy** with user with the user level Operator credantials (*userLoginOperator* and *passwordOperator*).

20.2. The DUT responds with **HTTP 401 Unauthorized** error.

21. ONVIF client invokes **CreateUsers** with user with the user level Operator credantials (*userLoginOperator* and *passwordOperator*) with parameters
    - User[0].Username := “Test”
    - User[0].Password := “Test”
    - User[0].UserLevel := Administrator
    - Extension skipped
22. The DUT responds with **HTTP 401 Unauthorized** error.
23. ONVIF client invokes **SetSystemDateAndTime** with user with the user level Operator credantials (*userLoginOperator* and *passwordOperator*) with parameters
    - DateTimeType := NTP
    - DaylightSavings := true
    - TimeZone skipped
    - UTCDateTime skipped
24. The DUT responds with **HTTP 401 Unauthorized** error.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT allowed User access to the **SetScopes** command.

The DUT allowed User access to the **SetDiscoveryMode** command.

The DUT allowed User access to the **GetAccessPolicy** command.

The DUT allowed User access to the **CreateUsers** command.

The DUT allowed User access to the **SetSystemDateAndTime** command.

The DUT allowed Operator access to the **SetScopes** command.

The DUT allowed Operator access to the **SetDiscoveryMode** command.

The DUT allowed Operator access to the **GetAccessPolicy** command.

The DUT allowed Operator access to the **CreateUsers** command.

The DUT allowed Operator access to the **SetSystemDateAndTime** command.

4.3 System

4.3.1 Firmware Upload

Test Label: System -FirmwareUpgrade

Test Case ID: QUICK_INSTALL-4-1-1

ONVIF Core Specification Coverage: Firmware Upload via HTTP

Command Under Test: StartFirmwareUgrade

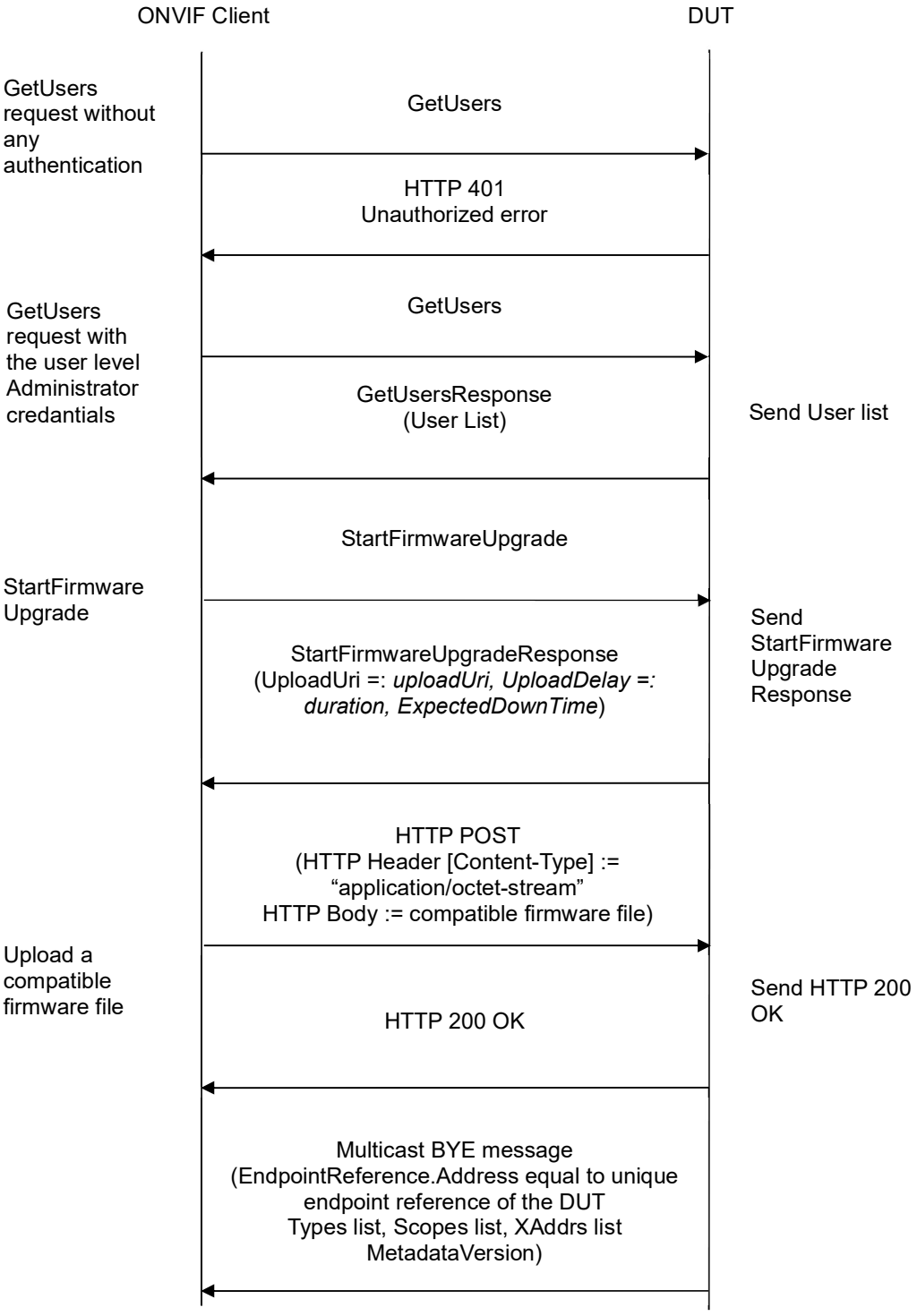
WSDL Reference: devicemgmt.wsdl

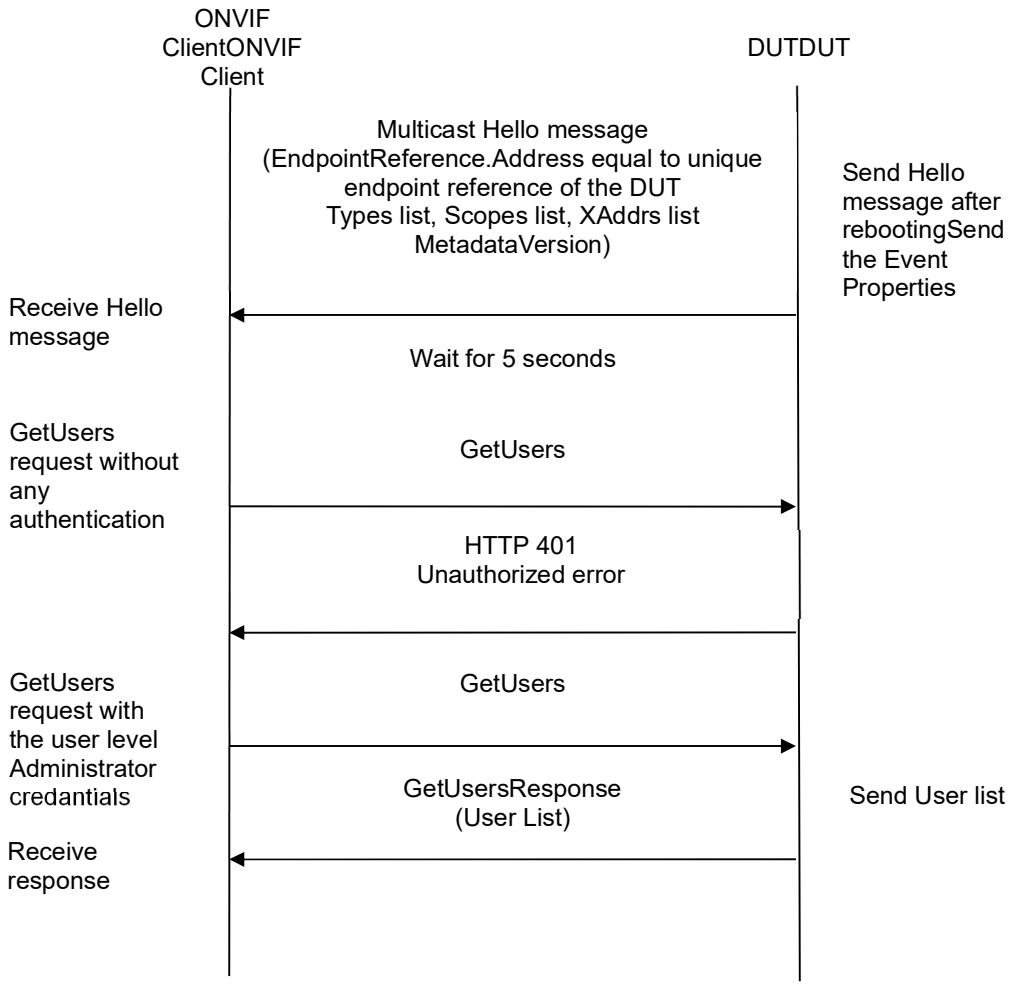
Test Purpose: To verify that the firmware upgrade is correctly executed.

Pre-requisite: GetServices command is supported by the DUT. HttpFirmwareUpgrade feature is supported by the DUT as indicated by the System.HttpFirmwareUpgrade. Location of Compatible firmware file is provided.

Test Configuration: ONVIF Client and DUT

Test Sequence:


ONVIF Client
DUT
GetUsers request without any authentication
GetUsers
HTTP 401 Unauthorized error
GetUsers request with the user level Administrator credantials
GetUsers
GetUsersResponse (User List)
Send User list
StartFirmwareUpgrade
StartFirmware Upgrade
StartFirmwareUpgradeResponse (UploadUri =: uploadUri, UploadDelay =: duration, ExpectedDownTime)
Send StartFirmware Upgrade Response
HTTP POST (HTTP Header [Content-Type] := “application/octet-stream” HTTP Body := compatible firmware file)
Upload a compatible firmware file
HTTP 200 OK
Send HTTP 200 OK
Multicast BYE message (EndpointReference.Address equal to unique endpoint reference of the DUT Types list, Scopes list, XAddrs list MetadataVersion)



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **GetUsers** without any authentication.
4. The DUT responds with **HTTP 401 Unauthorized** error.
5. ONVIF Client invokes **GetUsers** with user with the user level Administrator credantials.
6. The DUT responds with a **GetUsersResponse** message with parameters
   - UserList
7. ONVIF Client invokes **StartFirmwareUpgrade**.

8. The DUT responds with a **StartFirmwareUpgradeResponse** message with parameters
   - UploadUri =: *uploadUri*
   - UploadDelay =: *uploadDuration*
   - ExpectedDownTime =: *downTimeDuration*
9. ONVIF Client waits for time *uploadDuration*.
10. ONVIF Client invokes **HTTP POST** to *uploadUri* with parameters
    - HTTP Header [Content-Type] := “application/octet-stream”
    - HTTP Body := compatible firmware file
11. The DUT responds with **HTTP 200 OK** message.
12. If BYE message is supported by the DUT as indicated by the System.DiscoveryBye capabilities, the DUT will send Multicast **BYE** message before rebooting with parameters
    - EndpointReference.Address equal to unique endpoint reference of the DUT
    - Types list
    - Scopes list
    - XAddrs list
    - MetadataVersion
13. Until *timeout1* timeout expires, repeat the following steps:

    13.1. The DUT will send Multicast **Hello** message after it is successfully rebooted with parameters:
    - EndpointReference.Address equal to unique endpoint reference of the DUT
    - Types list
    - Scopes list
    - XAddrs list := *xaddrsList*
    - MetadataVersion

    13.2. If *xaddrsList* contains URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, go to step 15.
14. If *timeout1* timeout expires for step 5 without Hello with URI address with not a LinkLocal IPv4 address from ONVIF Client subnet, FAIL the test and skip other steps.
15. ONVIF client waits for 5 seconds after Hello was received.
16. ONVIF Client invokes **GetUsers** without any authentication.
17. The DUT responds with **HTTP 401 Unauthorized** error.
18. ONVIF Client invokes **GetUsers** with user with the user level Administrator credantials.

19. The DUT responds with a **GetUsersResponse** message with parameters

- UserList

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **GetUsersResponse** message.

The DUT allowed Anonymous access to the **GetUsers** command.

The DUT did not send **StartFirmwareUpgradeResponse** message.

The DUT did not response with **HTTP 200 OK** to HTTP POST request.

The DUT did not send **Bye** message with EndpointReference.Address equal to unique endpoint reference of the DUT in case Bye message was supported by the DUT.

Note: Reboot Timeout will be used for waiting for the Bye message from the DUT. Reboot Timeout will be taken from the Reboot Timeout field of ONVIF Device Test Tool.

Note: *timeout1* will be selected as maximum of (*downTimeDuration*, Reboot Timeout), where Reboot Timeout will be taken from the Reboot Timeout field of ONVIF Device Test Tool.

Note: IPv4 address from Hello shall be used for further test cases.

4.3.2 Invalid Firmware Upload

Test Label: System -FirmwareUpgradeInvalid

Test Case ID: QUICK_INSTALL-4-1-2

ONVIF Core Specification Coverage: Firmware Upload via HTTP

Command Under Test: StartFirmwareUgrade

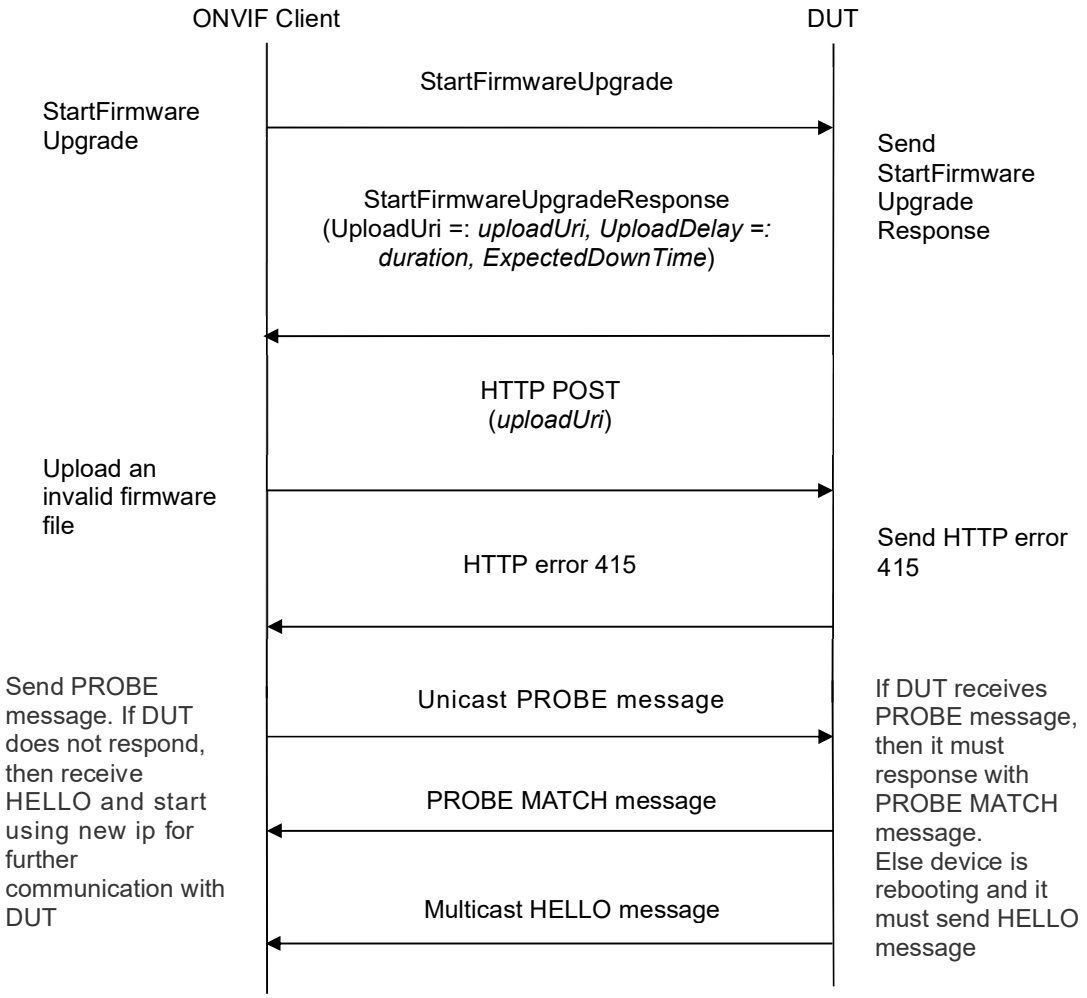
WSDL Reference: devicemgmt.wsdl

Test Purpose: To verify that the firmware upgrade fails with the expected error code.

Pre-requisite: GetServices command is supported by the DUT. HttpFirmwareUpgradefeature is supported by the DUT as indicated by the System.HttpFirmwareUpgrade.

Test Configuration: ONVIF Client and DUT

Test Sequence:



Test Procedure:

1. Start an ONVIF Client.
2. Start the DUT.
3. ONVIF Client invokes **StartFirmwareUpgrade**.
4. The DUT responds with a **StartFirmwareUpgradeResponse** message with parameters
   - UploadUri =: *uploadUri*
   - UploadDelay =: *duration*
   - ExpectedDownTime
5. ONVIF Client waits for time *duration*.

6. ONVIF Client invokes **HTTP POST** to *uploadUri* with parameters
   - HTTP Header [Content-Type] := “application/octet-stream”
   - HTTP Body := invalid firmware file
7. The DUT responds with **HTTP 415** message.
8. ONVIF client waits Reboot timeout.
9. ONVIF Client sends PROBE message and if DUT does not respond with PROBE MATCH mesage then go to the step 10. If DUT responds, then finish the test.
10. ONVIF Client waits for Hello message sent from newly configured address by the DUT. Then ONVIF Client starts using this newly configured address for further communications with DUT.

Test Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **StartFirmwareUpgradeResponse** message.

The DUT did not response with **HTTP 415** to HTTP POST request.

Annex A

This section describes the meaning of the following definitions. These definitions are used in the test case description.

A.1 Create user with defined user level

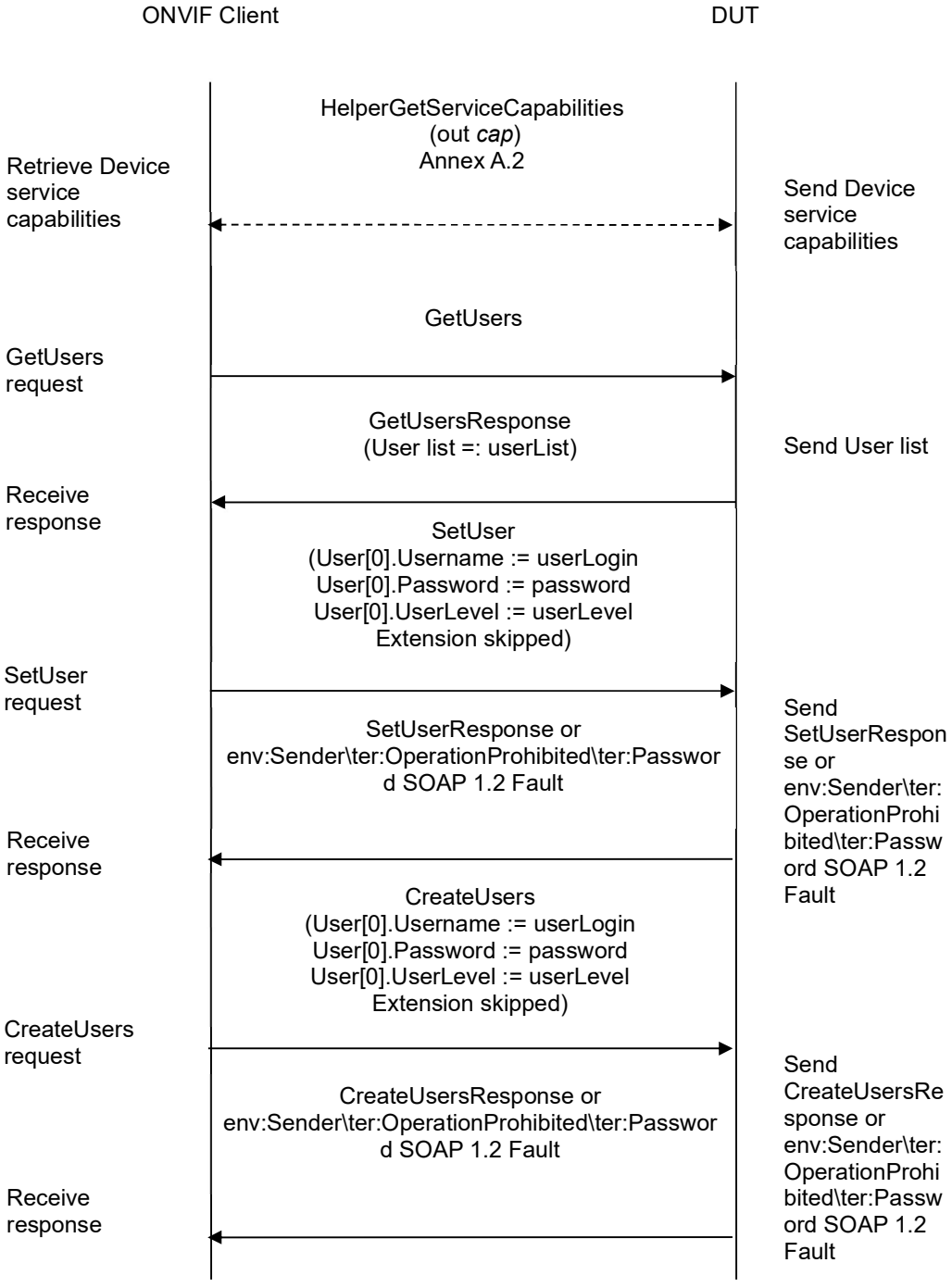
Name: HelperCreateUserLevel

Procedure Purpose: Helper procedure to create user with predefined user level or change existing with new one.

Pre-requisite: GetServices command is supported by the DUT. Maximum Username Length is supported by the DUT as indicated by the Capabilities.Security.MaxUsernameLength, Maximum Password Length is supported by the DUT as indicated by the Capabilities.Security.MaxPasswordLength.

Input: The user level (*userLevel*) of user to be created (*userLevel* shall have User or Operator value).

Returns: The user login (*userLogin*) with predefined user level and corresponding user password (*password*).


ONVIF Client
DUT
HelperGetServiceCapabilities
(out cap)
Annex A.2
Retrieve Device service capabilities
Send Device service capabilities
GetUsers
GetUsers request
GetUsersResponse
(User list =: userList)
Send User list
Receive response
SetUser
(User[0].Username := userLogin
User[0].Password := password
User[0].UserLevel := userLevel
Extension skipped)
SetUser request
SetUserResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Send SetUserResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Receive response
CreateUsers
(User[0].Username := userLogin
User[0].Password := password
User[0].UserLevel := userLevel
Extension skipped)
CreateUsers request
CreateUsersResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Send CreateUsersResponse or env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 Fault
Receive response

Procedure:

1. ONVIF Client gets the service capabilities (out *cap*) by following the procedure mentioned in Annex A.2.
2. If *cap* does not contain Security.MaxPasswordLength or Security.MaxUserNameLength, FAIL the test and skip other steps.
3. ONVIF Client invokes **GetUsers**.
4. The DUT responds with a **GetUsersResponse** message with parameters
   - User list := *userList*
5. If there is user with user level *userLevel* in *userList*:

   5.1. Set the following:
   - *passwordLength* := *cap*.Security.MaxPasswordLength
   - *userLogin* := Username of user with user level equal to *userLevel* from *userList*
   - *password* := random string, contains *passwordLength* ASCII characters

   5.2. ONVIF Client invokes **SetUser** with parameters
   - User[0].Username := *userLogin*
   - User[0].Password := *password*
   - User[0].UserLevel := *userLevel*
   - Extension skipped

   5.3. If the DUT responds with **SetUserResponse** message, skip other steps.

   5.4. If the DUT returns env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 fault:

   5.4.1. Set the following:
   - *password* := random string, contains *passwordLength* ASCII characters

   5.4.2. Go to the step 6.2.

   5.5. If DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.
6. If there are no users with user level *userLevel* in *userList*:

   6.1. Set the following:
   - *userLoginLength* := *cap*.Security.MaxUserNameLength
   - *passwordLength* := *cap*.Security.MaxPasswordLength
   - *userLogin* := random string, contains *userLoginLength* low case alphbet characters, differs from usernames listed in *userList*
   - *password* := random string, contains *passwordLength* ASCII characters

   6.2. ONVIF Client invokes **CreateUsers** with parameters

- User[0].Username := *userLogin*
- User[0].Password := *password*
- User[0].UserLevel := *userLevel*
- Extension skipped

6.3. If the DUT responds with **CreateUsersResponse** message, skip other steps.

6.4. If the DUT returns env:Sender\ter:OperationProhibited\ter:Password SOAP 1.2 fault:

6.4.1. Set the following:

- *password* := random string, contains *passwordLength* ASCII characters

6.4.2. Go to the step 7.2.

6.4.3. If the DUT returns other SOAP 1.2 fault, FAIL the test and skip other steps.

Procedure Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **GetServiceCapabilitiesResponse** message.

The DUT did not send **GetUsersResponse** message.

A.2 Get service capabilities

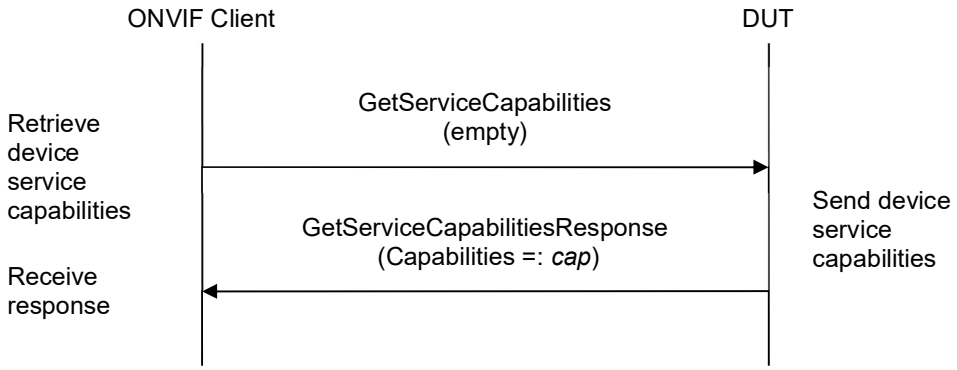
Name: HelperGetServiceCapabilities

Procedure Purpose: Helper procedure to get device service capabilities.

Pre-requisite: None.

Input: None

Returns: The service capabilities (*cap*).



Procedure:

1. ONVIF Client invokes **GetServiceCapabilities**.
2. The DUT responds with a **GetServiceCapabilitiesResponse** message with parameters
    - Capabilities =: *cap*

Procedure Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not send **GetServiceCapabilitiesResponse** message.

A.3 Time synchronization

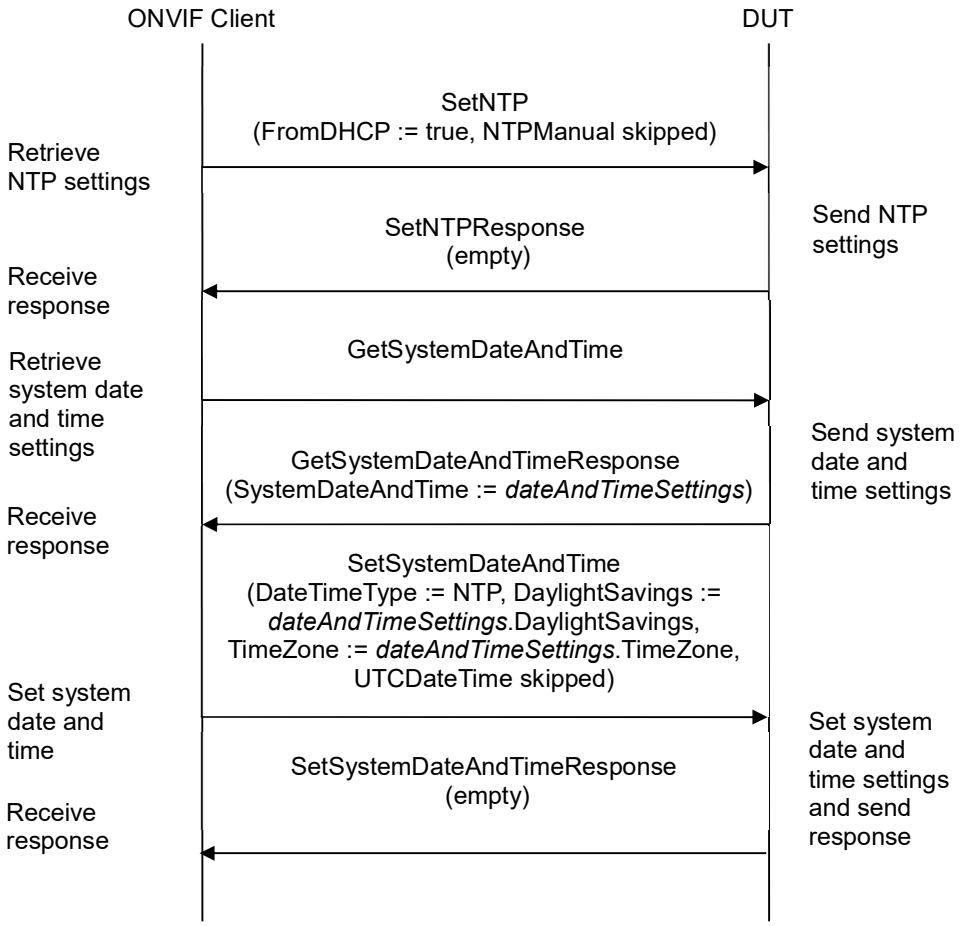
Name: HelperTimeSynchronization

Procedure Purpose: Helper procedure for time synchronization between ONVIF Client and DUT after hard system factory default.

Pre-requisite: None.

Input: None

Returns: None



Procedure:

1. ONVIF Client invokes **SetNTP** without any authentication with parameters
   - FromDHCP := true
   - NTPManual skipped

2. The DUT responds with a **SetNTPResponse** message.
3. ONVIF client invokes **GetSystemDateAndTime** without any authentication.
4. The DUT responds with **GetSystemDateAndTimeResponse** message with parameters
   - SystemDateAndTime **:=** *dateAndTimeSettings*
5. ONVIF Client invokes **SetSystemDateAndTime** without any authentication with parameters
   - DateTimeType := NTP
   - DaylightSavings := *dateAndTimeSettings*.DaylightSavings
   - TimeZone := *dateAndTimeSettings.TimeZone*
   - UTCDateTime skipped
6. The DUT responds with a **SetSystemDateAndTime** message.

Procedure Result:

PASS –

The DUT passed all assertions.

FAIL –

The DUT did not allow Anonymous access to the **SetNTP** command.

The DUT did not send **SetNTPResponse** message.

The DUT did not allow Anonymous access to the **GetSystemDateAndTime** command.

The DUT did not send **GetSystemDateAndTimeResponse** message.

The DUT did not allow Anonymous access to the **SetSystemDateAndTime** command.

The DUT did not send **SetSystemDateAndTimeResponse** message.