TCG Compliance TNC IF-MAP Metadata for Network Security Compliance Test Plan

Version 1.00
Revision 11
10 March 2011
Published

Contact:
admin@trustedcomputinggroup.org

TCG PUBLISHED

Disclaimer

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to provide specific requirements for the compliance tests for IF-MAP Metadata for Network Security, an open standard for sharing information among network security devices and systems. IF-MAP defines a standard interface between the Metadata Access Point and other elements of the TNC architecture. This document defines and specifies the Test Cases that need to be designed to verify compliance of a device (e.g. Sensor, Flow Controller, Policy Decision Point) with the IF-MAP Metadata for Network Security specification Version 1.0 Revision 24 [1].

1.2 Scope and Audience

The intended audience for this document includes compliance test designers and implementers, as well as product developers and customers who need to understand the IF-MAP Metadata for Network Security compliance tests. Readers should be familiar with the TNC Architecture [5]with the Compliance_TNC Compliance and Interoperability Principles specification [4]and with IF-MAP Binding for SOAP v2.0 [2]

1.3 Terminology

The following terms will be used in this document:

NS Client – A Client that implements IF-MAP Metadata for Network Security.

NS Server – A Server that implements IF-MAP Metadata for Network Security.

Policy Decision Point, or *PDP* – A client operating as a PDP that implements IF-MAP Metadata for Network Security.

Sensor - A client operating as a Sensor that implements IF-MAP Metadata for Network Security.

Flow Controller - A client operating as a Flow Controller that implements IF-MAP Metadata for Network Security.

2 Specifications and Components

2.1 Specifications

This document is based on the IF-MAP Metadata for Network Security specification [1], the IF-MAP Binding for SOAP 2.0 specification [2], and the Compliance TNC Compliance and Interoperability Principles document [4]. The Compliance TNC Compliance and Interoperability Principles document provides an overview of the Compliance TNC testing. The IF-MAP Binding for SOAP v2.0 specification defines the IF-MAP protocol elements and rules. The IF-MAP Metadata for Network Security specification describes the specific metadata types and links for their use in a Network Security environment to control and manage access to a network infrastructure.

2.2 Components

Devices such as Policy Decision Points, Sensors and Flow Controllers publish and/or query or subscribe to changes to IF-MAP Metadata on an IF-MAP server. A Policy Decision Point (PDP) such as a RADIUS server can determine that action should be taken on a device that violates network policy, and update the metadata accordingly. A Flow Controller such as a firewall could then take remedial action, which typically might be to isolate the device to a quarantined remediation state until appropriate steps can be taken to bring the device into compliance.

Note that the tests described here will be done to verify that a specific IF-MAP device complies with the basic protocol rules for Network Security operations. The operation of a particular device in a multi-vendor network as it applies those rules for Network Security is covered in a separate Interoperability Test plan [5], which is run after these compliance tests are successfully executed. This Interoperability Testing, informally called a “Plugfest,” is held once or twice a year, and is out of the scope of this test plan. In order to complete certification, attending a Plugfest is the next step after passing the tests described here.

2.2.1 IF-MAP Servers

Generally, IF-MAP servers are metadata independent. They operate the same way, regardless of whether the metadata concerns Network Security or endangered birds or any other topic. However, IF-MAP servers that support both IF-MAP 2.0 and IF-MAP 1.0 clients must do some metadata-specific translation to make this work. Therefore, the only test cases in this document for IF-MAP servers are test cases servers need to run if they provide such backward compatibility, i.e. only if they support both IF-MAP 2.0 and IF-MAP 1.0 clients.

The IF-MAP Compliance tests for devices operating as an IF-MAP server in a Network Security environment test that a server properly implements the old and new namespace rules for device identifiers. The Device Under Test (DUT) for these tests is an IF-MAP server.

To test a Server’s compliance with IF-MAP Metadata for Network Security namespace and device identifiers, a sequence of IF-MAP requests and responses must be exchanged between the Server and the IF-MAP client. Note that some NS Servers may not support backward compatibility metadata, and therefore would not need to run these tests.

2.2.2 IF-MAP Clients

These test cases are IF-MAP Metadata for Network Security compliance tests that all NS Clients must run.

The IF-MAP Compliance tests for devices operating as IF-MAP clients in a Network Security environment test that a client properly implements IF-MAP Metadata for Network Security. The Device Under Test (DUT) for these tests is a PDP, Sensor, or Flow Controller. A DUT may include one or more of these functions.

To test a NS Client's compliance with IF-MAP Metadata for Network Security, a sequence of IF-MAP requests and responses must be exchanged between the NS Client and the IF-MAP server. The tests will verify that the NS Client can connect to the IF-MAP server and perform operations as appropriate for the client. Note that some NS Clients may only publish metadata, while others may only subscribe to notifications.

2.2.2.1 Policy Decision Points (PDPs)

IF-MAP Clients operating as PDPs in Network Security must run the NS Client tests and additional test cases specific to PDP functionality. The PDP tests verify that a PDP Client properly implements IF-MAP Metadata for Network Security as it applies to PDP operation. The Device Under Test (DUT) for this test is a PDP operating as an IF-MAP Client.

To test a PDP's compliance with IF-MAP Metadata for Network Security, a sequence of IF-MAP requests and responses must be exchanged between the PDP and the IF-MAP server. The PDP will conform by connecting to the IF-MAP server, subscribing to events from the MAP server, and publishing events to the MAP server, as appropriate for the device. After each exchange, test traffic shall be sent to ensure that the test criteria are met and the PDP has properly implemented the IF-MAP Metadata for Network Security specification.

2.2.2.2 Sensors

IF-MAP Clients operating as Sensors in Network Security must run the NS Client tests and additional test cases specific to sensor functionality. The IF-MAP Compliance tests for Sensors operating as IF-MAP clients in a Network Security environment test that a Sensor properly implements IF-MAP Metadata for Network Security as it applies to sensor operation. The Device Under Test (DUT) for test is a Sensor operating as an IF-MAP Client.

To test a Sensor's compliance with IF-MAP Metadata for Network Security, a sequence of IF-MAP requests and responses must be exchanged between the Sensor and the IF-MAP server. The Sensor will conform by connecting to the IF-MAP server, publishing events as appropriate for the sensor, and optionally subscribing to events from the MAP server

2.2.2.3 Flow Controllers

IF-MAP Clients operating as Flow Controllers in Network Security must run the NS Client tests and additional test cases specific to flow controller functionality. The IF-MAP Compliance tests for Flow Controllers operating as IF-MAP clients in a Network Security environment test that a Flow Controller properly implements IF-MAP Metadata for Network Security as it applies to flow controller operation. The Device Under Test (DUT)for this test is a flow controller operating as an IF-MAP Client.

To test a flow controller's compliance with IF-MAP Metadata for Network Security, a sequence of IF-MAP requests and responses must be exchanged between the flow controller and the IF-MAP server. Flow Controllers will typically subscribe to notifications, and might publish metadata.

2.2.2.4 Other Clients

There may be other devices which operate as NS Clients, and they would need to run the NS Client test cases specified in this test plan. Additional device-specific cases are as-yet undefined, and therefore beyond the scope of this test plan.

Other clients that may support IF-MAP Metadata but not specifically for Network Security would also be beyond the scope of this test plan.

3 Requirements and Recommendations

The IF-MAP Metadata for Network Security v1.0 specification includes many requirements and recommendations for NS Clients, and a few for NS Servers. This section lists only the mandatory requirements since the compliance tests for IF-MAP Metadata for Network Security only test normative requirements (not recommendations).

This section has five subsections. The first section lists mandatory requirements upon all NS Clients, which are run for all clients. The second section lists mandatory requirements upon Policy Decision Points (PDPs) which are specific to their functionality. The third section applies only to Sensors, and the fourth to Flow Controllers. The fifth section contains specific Backwards Compatibility requirements for IF-MAP servers.

As required by the TCG Compliance and Interoperability Guidelines, each requirement listed below has a unique name composed of the string “CTNC” (for Compliance_TNC), “MAP-NS” (indicating that these are requirements for MAP Servers and Clients operating in a Network Security environment), “Client” or “Server” depending on which component the requirement applies to, “GEN” (for General requirements to be run against all Clients), “PDP” (to be run against PDPs), “SEN” (to be run against Sensors, or “FC” (to be run against Flow Controllers), “REQ” indicating it is a requirement, and a compliance classifier (“M” for MUST or MUST NOT). Usage classifiers are not included in requirement names at this time.

3.1 Requirements for NS Clients

[CTNC-MAP-NS-CLNT-REQ-1-M] These metadata types [access-request-device access-request-mac, access-request-ip, authenticated-as, authenticated-by, device-ip, discovered-by and enforcement-report] MUST be attached only to prescribed kinds of identifiers or to links adjacent to prescribed kinds of identifiers, as specified in the type definition, so that match-links filters will work as intended. (Section 3.1)

[CTNC-MAP-NS-CLNT-REQ-2-M] This requirement is obsolete and no longer used.

[CTNC-MAP-NS-CLNT-REQ-3-M] The MAP Client MUST NOT delete metadata with any other ifmap-publisher-id (other than its own). (Sec 3.1)

[CTNC-MAP-NS-CLNT-REQ-4-M] If a MAP Client is publishing metadata from multiple sources (which may or may not include itself), it MUST generate a unique discoverer-id for each source using the form “ifmap-publisher-id:UID” where UID may be a simple ordinal value. (Sec 3.1)

[CTNC-MAP-NS-CLNT-REQ-5-M] Clients MUST publish access-request-device (Link) metadata only between: access-request and device (Sec 3.1.1)

[CTNC-MAP-NS-CLNT-REQ-6-M] Clients MUST publish access-request-ip link metadata only between: access-request and ip-address (Sec 3.1.2)

[CTNC-MAP-NS-CLNT-REQ-7-M] Clients MUST publish access-request-mac (Link) only between: access-request and mac-address (Sec 3.1.3)

[CTNC-MAP-NS-CLNT-REQ-8-M] Clients MUST publish authenticated-as (Link) metadata only between: access-request and identity (Sec 3.1.4)

[CTNC-MAP-NS-CLNT-REQ-9-M] Clients MUST publish authenticated-by (Link) metadata only between: access-request and device (Sec 3.1.5)

[CTNC-MAP-NS-CLNT-REQ-10-M] If device-characteristic data is used, manufacturer, model, os, os-version, device-type, and discovery-method elements, the type field’s value MUST take the form of a vendor-defined type or a TCG-defined type as specified in section 3.1 (Sec 3.1.8).

[CTNC-MAP-NS-CLNT-REQ-11-M] Clients MUST publish device-ip (Link) only between: device and ip-address (Sec 3.1.9)

[CTNC-MAP-NS-CLNT-REQ-12-M] Clients MUST publish discovered-by (Link) only between: ip-address and device or mac-address and device (Sec 3.1.10)

[CTNC-MAP-NS-CLNT-REQ-13-M] Clients MUST publish enforcement-report (Link) only between: ip-address and device or mac-address and device (Sec 3.1.11)

[CTNC-MAP-NS-CLNT-REQ-14-M] This requirement is obsolete and no longer used.

[CTNC-MAP-NS-CLNT-REQ-15-M] If a MAP Client specifies enforcement-action type as other, the client MUST specify a non-empty string for the other-type-definition field. (Sec 3.1.11)

[CTNC-MAP-NS-CLNT-REQ-16-M] If a MAP Client specifies enforcement-action type as other, the other-type-definition field's value MUST take the form of a vendor-defined type or a TCG-defined type as specified in section 3.1 (Sec 3.1.11)

[CTNC-MAP-NS-CLNT-REQ-17-M] A MAP Client MUST publish enforcement-report metadata using the update operation and delete the enforcement-report when it no longer applies (e.g. when the enforcing element is no longer taking enforcement action against the endpoint). (Sec 3.1.11)

[CTNC-MAP-NS-CLNT-REQ-18-M] MAP Clients publishing event data with type of "cve" MUST identify the vulnerability using the vulnerability-uri element. (Sec 3.1.12)

[CTNC-MAP-NS-CLNT-REQ-19-M] MAP Clients publishing event data with type of "other" MUST specify a non-empty string for the other-type-definition field. (Sec 3.1.12)

[CTNC-MAP-NS-CLNT-REQ-20-M] For MAP Clients publishing event data with type of "other," the other-type-definition field's value MUST take the form of a vendor-defined type or a TCG-defined type as specified in section 3.1 (Sec 3.1.12)

[CTNC-MAP-NS-CLNT-REQ-21-M] MAP Clients using Version 2.0 or later of the TNC IF-MAP Binding for SOAP MUST publish events using the notify operation (section 3.7.1 of [Trusted Computing Group, TNC IF-MAP Binding for SOAP, Revision 2.0, May 2010] Sec 3.1.12)

[CTNC-MAP-NS-CLNT-REQ-22-M] A MAP Client MUST publish unexpected-behavior metadata using the update operation and delete the unexpected-behavior when it no longer applies (e.g. when the behavior monitoring system has determined that the endpoint has returned to normal behavior). (Sec 3.1.19)

[CTNC-MAP-NS-CLNT-REQ-23-M] A client that publishes an access-request-ip or access-request-mac link or other metadata associated with an access-request MUST be configurable to apply lifetime="session" and that SHOULD be the default. (Sec 5)

3.2 Requirements for PDP Clients

[CTNC-MAP-PDP-REQ-1-M] When successfully authenticating an endpoint, a MAP Client in a PDP MUST use a device identifier and access-request identifier to publish the following metadata: (Sec 4.1.1)

a. access-request-device metadata on the link between the access-request identifier and the endpoint's device identifier

b. access-request-mac metadata on the link between the access-request identifier and the endpoint's mac-address identifier (when authenticated at layer 2 or otherwise available)

c. access-request-ip metadata on the link between the access-request identifier and the endpoint's ip-address identifier (when authenticated at layer 3 or otherwise available)

d. authenticated-by metadata on the link between the access-request identifier and the PDP's device identifier

[CTNC-MAP-PDP-REQ-2-M] If the PDP is aware of any of the following metadata, it MUST publish: (Sec 4.1.1)

a. authenticated-as metadata on the link between the access-request identifier and any identity identifiers associated with the user's authenticated identity

b. capability metadata on the access-request identifier

c. device-attribute metadata on the link between the access-request identifier and the endpoint's device identifier

d. device-characteristic metadata on the link between the access-request identifier and its own device identifier

e. role metadata on the link between the access-request identifier and an identity identifier

f. layer2-information metadata on the link between the access-request identifier and the device identifier of the PEP (when authenticated at layer 2 or otherwise available)

g. wlan-information metadata on the link between the access-request identifier and the device identifier of the PEP

[CTNC-MAP-PDP-REQ-3-M] The PDP MUST be configurable to create all of the metadata described in REQ-1 and REQ-2 with a lifetime attribute of "session". (Sec 4.1.1)

[CTNC-MAP-PDP-REQ-4-M] When an endpoint disconnects from the network, the PDP MUST delete any metadata associated with the endpoint if the lifetime attribute of the publish request is "session". (Sec 4.1.1)

[CTNC-MAP-PDP-REQ-5-M] A PDP which implements subscription capabilities MUST also take remediation action against endpoints based on policy definitions. (Sec 4.1.1)

[CTNC-MAP-PDP-REQ-6-M] To take remediation action against an endpoint, the PDP MUST implement subscription capabilities. (Sec 4.1.1)

3.3 Requirements for Sensor Clients

[CTNC-MAP-SEN-REQ-1-M] A Sensor acting as a MAP Client MUST publish event metadata, or device characteristic metadata, or both.(4.1.2) This requirement is ambiguous since some sensors may only publish specific metadata, e.g. ip-mac or location. This requirement is being ignored in this plan.

[CTNC-MAP-SEN-REQ-2-M] Event metadata and/or device characteristic metadata MUST be attached to either the IP address or MAC address of the endpoint for which the event or device characteristic was detected. (4.1.2)

[CTNC-MAP-SEN-REQ-3-M] The Sensor MUST use notify to publish event metadata. (4.1.2)

[CTNC-MAP-SEN-REQ-4-M] A Sensor that publishes device-characteristic metadata MUST generate a unique device identifier for itself. (4.1.2)

[CTNC-MAP-SEN-REQ-5-M] A sensor publishing device-characteristic metadata MUST publish on a link between the device identifier for the Sensor and either an ip-address or mac-address identifier. (4.1.2)

[CTNC-MAP-SEN-REQ-6-M] When a sensor publishes other types of metadata such as ip-mac or location, the Sensor MUST be configurable to publish this metadata with a lifetime attribute of “session”. (4.1.2)

[CTNC-MAP-SEN-REQ-7-M] When published metadata no longer describes the state of the network, the Sensor MUST delete the metadata. (4.1.2)

[CTNC-MAP-SEN-REQ-8-M] Sensors other than DHCP servers MUST NOT publish end-time metadata on ip-mac links. (Sec 3.1.13)

[CTNC-MAP-SEN-REQ-9-M] A Sensor publishing location metadata MUST include at least one location-information sub element that provides contextual data about the location of the user or device. (Sec 3.1.15)

[CTNC-MAP-SEN-REQ-10-M] For the location-information element's type attribute, the type field’s value MUST take the form of a vendor-defined type or a TCG-defined type as specified in section 3.1. (Sec 3.1.15)

[CTNC-MAP-SEN-REQ-11-M] Any IF-MAP Client that responds to request-for-investigation metadata MUST be configurable to respond only to a particular set of qualifier strings. (3.1.16)

[CTNC-MAP-SEN-REQ-12-M] When a MAP Client notices a request-for-investigation, if the request-for-investigation contains a qualifier, the MAP Client MUST only respond if the qualifier is in the configured set. (Sec 3.1.16)

[CTNC-MAP-SEN-REQ-13-M] If the request-for-investigation contains a qualifier, the MAP Client MUST only respond if the qualifier is in the configured set. (Sec 3.1.16)

3.4 Requirements for Flow Controller Clients

[CTNC-MAP-FC-REQ-1-M] A MAP Client in a Flow Controller MUST implement the ability to subscribe to notifications. (4.1.3)

[CTNC-MAP-FC-REQ-2-M] The Flow Controller MUST apply policy based on poll results. (4.1.3)

[CTNC-MAP-FC-REQ-3-M] When an endpoint attempts to send traffic through a Flow Controller, the Flow Controller MUST subscribe to the MAP for metadata related to the endpoint. (4.1.3)

[CTNC-MAP-FC-REQ-4-M] A Flow Controller MUST provide access to endpoints based on policy definitions relative to metadata received in poll results. (4.1.3)

[CTNC-MAP-FC-REQ-5-M] If a Flow Controller publishes enforcement-report metadata, the metadata MUST be published on the link between the Flow Controller’s device identifier and the ip-address or mac-address of the endpoint. (4.1.3)

3.5 Requirements for NS Servers

The following requirements apply to NS servers supporting both IF-MAP 1.1 and IF-MAP Metadata for Network Security 1.0 protocol, in order to achieve backward compatibility.

[CTNC-MAP-NS-SERVER-REQ-1-M] For IF-MAP 2.0 clients, the authenticated-by (old namespace) link between an access-request and an ip-address SHOULD be replaced with an authenticated-by (new namespace) link between an access-request and a device along with a device-ip link (new namespace) between a device and the ip-address. The device identifier's name MUST be chosen so that it can't conflict with a valid device identifier published by any of the MAP Server's clients. (Sec 7)

[CTNC-MAP-NS-SERVER-REQ-2-M] For IF-MAP 2.0 clients, the layer2-information (old namespace) link between an access-request and an ip-address SHOULD be replaced with a layer2-information (new namespace) link between the access-request and a device along with a device-ip (new namespace) link between a device and an ip-address. The device identifier's name MUST be chosen so that it can't conflict with a valid device identifier published by any of the MAP Server's client.

4 Configurations and Topologies

4.1 NS Client Testing Common Setup

This common setup is for NS Client testing. For NS Server testing, please see section 4.2.

IF-MAP Client

In IF-MAP Client testing, only one client is required; this client is the device under test (DUT). No functional changes should be made to the DUT during the course of the testing.

IF-MAP Client must be configured with proper URI to communicate with IF-MAP Server.

IF-MAP Clients must be configured to authenticate with the IF-MAP Server via certificate-based authentication.

IF-MAP Server

The IF-MAP Server will be monitored by the test program for the requests coming from the IF-MAP Client DUT, hence the IF-MAP Server needs to be integrated with the test program and have the ability to send controlled packets. The IF-MAP Server needs to have the following capabilities:

- Must be implemented using SOAP v1.2 [4]
- Capable of capture and validation of SOAP request and response contents.
- Able to send SOAP packets via both secured (https) and non-secured (http) connection.
- Able to generate and send controlled SOAP packet and envelopes.
- Able to authenticate the MAP client via either certificate-based authentication in the TLS handshake or Basic Authentication.

The IF-MAP Server must be configured to authenticate credentials for one IF-MAP Client. This client must have full privilege to all MAP operations, identifiers, and metadata.

The IF-MAP Server must be provisioned with appropriate certificate(s) required for authentication by the IF-MAP Client.

4.2 NS Server Testing Common Setup

This common setup is for IF-MAP Server testing. For NS Client testing, please see section 4.1

IF-MAP Server

The IF-MAP Server does not send requests but only responds to requests sent by the NS Client. The MAP Server is the device under test (DUT); no functional changes should be made to the DUT during the course of the testing.

The IF-MAP Server must be configured with a valid URI to communicate with the IF-MAP Client.

The IF-MAP Server must be configured to authenticate IF-MAP Client credentials for a client that must have full privilege to all MAP operations, identifiers, and metadata.

The IF-MAP Server must be provisioned with appropriate certificate(s) required for authentication by IF-MAP Clients.

IF-MAP Client

Two devices acting as a IF-MAP clients will be needed to send requests to the IF-MAP server. One of these clients will use IF-MAP 1.X and the other will use IF-MAP 2.0 and IF-MAP Metadata for Network Security 1.0.

The IF-MAP server test cases in this plan only apply to servers supporting backward compatibility with IF-MAP Clients running the IF-MAP 1.1 specification, therefore this specification of the MAP client must be simulated by the test program, as described in the NS Server test cases. The test program must send various controlled requests while monitoring the response from the MAP server, hence the MAP client and/or test program needs to have the following capabilities:

- Must be implemented using SOAP v1.2 [4]
- Capable of capture and validation of SOAP request and response contents.
- Able to send SOAP packets via both secured (https) and non-secured (http) connection.
- Able to authenticate via either certificate-based authentication in the TLS handshake or Basic Authentication

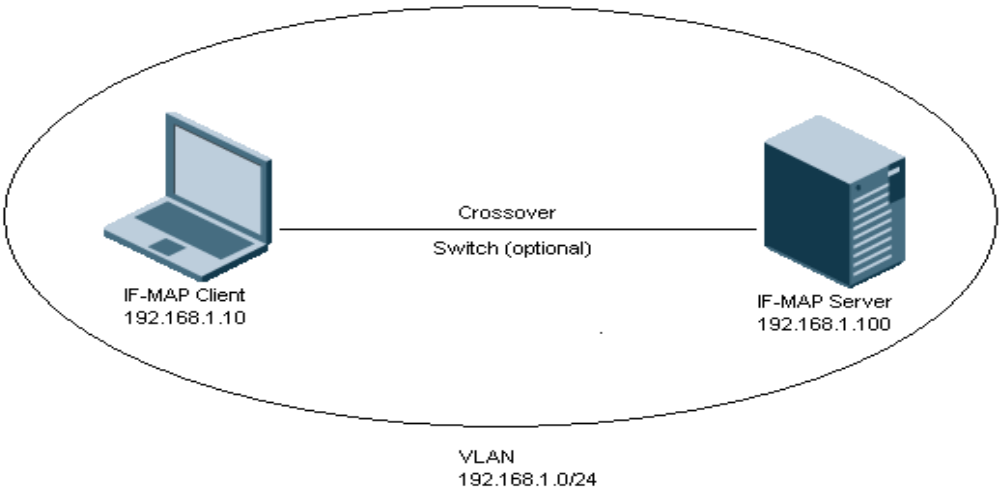
IF-MAP Clients must be configured with a valid URI to communicate with the IF-MAP Server.

IF-MAP Clients must be configured to authenticate with the IF-MAP Server via certificate-based authentication.

4.3 Test Topology

4.3.1 Client Test Topology

The test topology depicted below is used for all client test cases. The NS Client DUT is likely one of PDP, Sensor or Flow Controller devices.



Insert diagram with two clients, one IF-MAP version 1.0 the other running earlier IF-MAP version.

In all cases the results of each test must be logged (Pass, Fail, No-Run) along with a detailed description of any failure cases, including data captures of traffic to and from the DUT showing the error condition. Because data between the client and server will be encrypted on the wire, it must be captured by the NS Server or NS Client, whichever is not the DUT.

All devices MUST have consistent time and date. The test topology MUST be reset to default configuration at the start of every test case.

4.3.2 Validate Common Setup

1.Validate Client Test Setup

Before running any tests, validate the test environment as follows:

Test Steps:

Validate IF-MAP Server/client configuration:

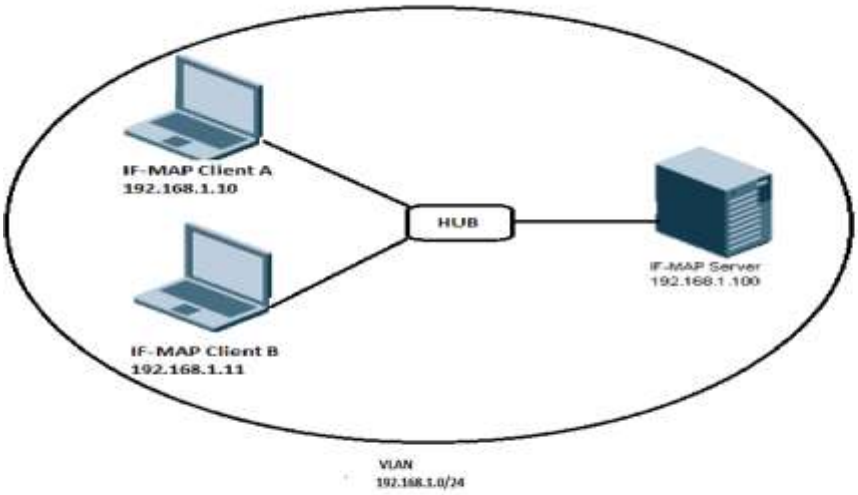
1. Disconnect IF-MAP Client from IF-MAP Server if connected
    a. Maintain disconnect state until MAP Server's httpd times out the previous session (this time may vary depending on the MAP Server configuration)
2. From IF-MAP Server side, configure IF-MAP serving URI (if configurable)
3. From IF-MAP Server side, configure proper credential for IF-MAP client.
4. From IF-MAP Client side, configure proper IF-MAP server URI.
5. From IF-MAP Client side, configure matching credential to be used.
6. From IF-MAP Client side, initiate a newSession request, validate session_id and publisher_id is returned.

Expected Outcomes:

1. Network traffic flows as expected.
2. IF-MAP Client is able to communicate and authenticate with IF-MAP server.
3. IF-MAP server recognizes the IF-MAP client and responds with publisher_id.

4.3.3 Server Test Topology

The test topology depicted below is used for all server test cases. The NS Server DUT test case is only for compatibility with IF-MAP Clients running 1.1 and backwards compatibility with earlier versions.



2.Validate Server Test Setup

Before running any tests, validate the test environment as follows:

Test Steps:

Validate IF-MAP Server/client configuration:

1. Disconnect IF-MAP Clients from IF-MAP Server if connected
    a. Maintain disconnect state until MAP Server's httpd times out the previous session (this time may vary depending on the MAP Server configuration)
2. From IF-MAP Server side, configure IF-MAP serving URI (if configurable)
3. From IF-MAP Server side, configure proper credential for IF-MAP client.
4. From IF-MAP Client A side, configure proper IF-MAP server URI.
5. From IF-MAP Client A side, configure matching credential to be used.
6. From IF-MAP Client A side, initiate a newSession request, validate session_id and publisher_id is returned.
7. Repeat Steps 4, 5, and 6 from Client B side.

Expected Outcomes:

1. Network traffic flows as expected.
2. IF-MAP Clients are able to communicate and authenticate with IF-MAP server.

3. IF-MAP server recognizes the IF-MAP clients and responds with publisher_ids.

5 Test Cases

In test cases where there are multiple expected outcomes listed, all of the expected outcomes must be met in order to pass the test. In test cases where there are multiple anticipated failures listed, any single failure results in failing the test.

5.1 IF-MAP Metadata for Network Security Compliance Test Cases, NS Clients

In the IF-MAP Metadata for Network Security Compliance Test Cases for NS Clients, the Device Under Test (DUT) is the NS Client. To verify that the NS Client correctly implements the specification, the test program will enter a test step state, and then require specific requests from the Client. Some cases will require loading the MAP server with data before running the test; these pre-conditions will be specified in the test case. Pre-loading of test data will be done by the test suite.

5.1.1 Always Available

Following are “Always Available” (AA) test cases. This behaviour must be monitored throughout the test, and any mismatching behaviour must immediately be flagged as a failure.

As such, there are no specific test steps associated with these test cases; the expected outcomes should be observed in the results of all subsequent test cases.

5.1.1.1 access-request-device link

[CTNC-MAP-NS-CLNT-AA-1]

Purpose: To verify that the access-request-device link is published only between access-request and device. Also verify that the client is configured to publish with lifetime=session, or uses lifetime=session as the default.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-5-M]

[[CTNC-MAP-NS-CLNT-REQ-23-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and device.

Anticipated Failures:

1. If client publishes access-request-device link between any other identifiers, test case fails.

2.If the link is published with lifetime=forever, the test case fails.

5.1.1.2 access-request-ip link

[CTNC-MAP-NS-CLNT-AA-2]

Purpose: To verify that the access-request-ip link is published only between access-request and ip-address, and that it is published with a lifetime=session.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-6-M]

[CTNC-MAP-NS-CLNT-REQ-23-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and ip address.

Anticipated Failures:

1. If client publishes access-request-ip link between any other identifiers, test case fails.

2.If the link is published with lifetime=forever, the test case fails.

5.1.1.3 access-request-mac link

[CTNC-MAP-NS-CLNT-AA-3]

Purpose: To verify that the access-request-mac link is published only between access-request and mac-address, and that it is published with a lifetime=session.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-7-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and mac and returns publishReceived.

Anticipated Failures:

1. If client publishes access-request-mac link between any other identifiers, test case fails.

2.If the link is published with lifetime=forever, the test case fails.

5.1.1.4 Authenticated-as link

[CTNC-MAP-NS-CLNT-AA-4]

Purpose: To verify that the authenticated-as link is published only between access-request and device identity.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-8-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and identity and returns publishReceived.

Anticipated Failures:

1. If client publishes authenticated-as link between any other identifiers, test case fails

5.1.1.5 Authenticated-by link

[CTNC-MAP-NS-CLNT-AA-5]

Purpose: To verify that the authenticated-by link is published only between access-request and device.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-9-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and device identity and returns publishReceived.

Anticipated Failures:

1. If client publishes authenticated-by link between any other identifiers, test case fails

5.1.1.6 device-characteristic link

[CTNC-MAP-NS-CLNT-AA-6]

Purpose: To verify that the device-characteristic link is published only between access-request and device, ip-address and device, or mac-address and device. Also test that, if device-characteristic metadata such as manufacturer, model, os, os-version, device-type and discovery-method elements are published, the type field must take the form of a vendor-defined type or a TCG-defined type.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-10-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between access-request and device identity and returns publishReceived.
2. Published device-characteristic link containing one or more of manufacturer, model, os, os-version, device-type and discovery-method elements has a valid vendor-defined type, as TCG-defined types are yet to be defined.

Anticipated Failures:

1. If client publishes device-characteristic link between any other identifiers, test case fails.

2. If vendor-defined type field of the form Vendor-ID:Type does not specify the Vendor-ID as a 24-bit SMI Private Enterprise Number Vendor ID,

5.1.1.7 device-ip link

[CTNC-MAP-NS-CLNT-AA-7]

Purpose: To verify that the device-ip link is published only between device and ip-address.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-11-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published device-ip link between device and ip.

Anticipated Failures:

1. If client publishes device-ip link between any other identifiers, test fails.

5.1.1.8 discovered-by link

[CTNC-MAP-NS-CLNT-AA-8]

Purpose: To verify that the discovered-by link is published between ip-address and device or mac-address and device.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-12-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Server accepts published link between device and ip or device and mac.

Anticipated Failures:

1. If client publishes discovered-by link between any other identifiers, test fails.

5.1.1.9 enforcement-report link

[CTNC-MAP-NS-CLNT-AA-9]

Purpose: To verify that the enforcement-report link, if supported by the NS Client, is published only between ip-address and device or mac-address and device, and that a supported enforcement action type is used. If a client uses “other” as the enforcement action type then a non-empty string must be specified for the other-type-definition field, and it must take the form of a vendor-defined type or a TCG-defined type. Also verify that a client deletes enforcement-report metadata when it no longer applies.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-3-M][CTNC-MAP-NS-CLNT-REQ-13-M]

Error! Reference source not found.

[CTNC-MAP-NS-CLNT-REQ-15-M]

[CTNC-MAP-NS-CLNT-REQ-16-M]

[CTNC-MAP-NS-CLNT-REQ-17-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Client sends enforcement-report link between device and ip or device and mac.

Anticipated Failures:

1. If client publishes enforcement-report link between any other identifiers, test fails

2. If client publishes enforcement-report link as anything other than an update request, test fails.

3. If enforcement-action is “other” then the client must include the other-type-definition element, otherwise the test fails.

4. If enforcement-action is “other” and the client includes the other-type-definition element, it must be of the format Vendor-ID:Type where Vendor-ID is the 24-bit SMI Private Enterprise Number Vendor ID of the client, and the Type is the type specified, otherwise the test fails.

5.1.1.10 event

[CTNC-MAP-NS-CLNT-AA-10]

Purpose: To verify that client who publishes event metadata uses notify. Events with type “cve” must identify the vulnerability-uri element, and event data with type “other” must specify a non-empty string for the other-type-definition field that takes the form of a vendor-defined type or a TCG-defined type.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-18-M]

[CTNC-MAP-NS-CLNT-REQ-19-M]

[CTNC-MAP-NS-CLNT-REQ-20-M]

[CTNC-MAP-NS-CLNT-REQ-21-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Client publishes events using notify with correct type and method.

Anticipated Failures:

1.If element-type = “other” and other-type-definition is not included, test fails.

2. If element-type = “other” and other-type-definition is included, it must be of the format Vendor-ID:Type where Vendor-ID is a 24-bit SMI Private Enterprise Number Vendor ID, and the Type is the type specified, otherwise the test fails.

3. If element-type = “cve” and vulnerability-uri element is not included, test fails.

4. A publish notify must be used, else test fails.

5.1.1.11 Delete request publisher-id validation

[CTNC-MAP-NS-CLNT-AA-11]

Purpose: To verify that the publisher-id of a delete requests matches that of the initial publish-update.

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Client publishes a delete request.

Anticipated Failures:

1. The publisher-id of the client making the delete request is different than the publisher-id of the metadata being deleted

5.1.1.12 Location Metadata Test

[CTNC-MAP-NS-CLNT-AA-12]

Purpose: To verify that a sensor publishing location metadata includes at least one location-information sub element providing contextual data about the location. Also verify that, for the location-information’s element type attribute, the type field’s value takes the form of a vendor-defined type, or a TCG-defined type. When a sensor publishes location metadata, it must be configurable to, or default to, publishing it with a lifetime attribute of “session.”

This test case is for the following requirements:

[CTNC-MAP-SEN-REQ-6-M]

[CTNC-MAP-SEN-REQ-9-M]

[CTNC-MAP-SEN-REQ-10-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. Sensor client publishes location metadata with at-least one location-information sub element.
2. Sensor client publishes location information’s Type field in the form of a vendor-defined type, or a TCG-defined type.

Anticipated Failures:

1. If sensor client publishes location metadata with no location-information sub element, the test fails.
2. If sensor client publishes location metadata and the type field value does not take a vendor-specific type or TCG-defined type, the test fails.

5.1.1.13 ip-mac link

[CTNC-MAP-NS-CLNT-AA-13]

Purpose: To verify that non-DHCP Sensor clients do not publish end-time metadata on ip-mac links. And to verify that ip-mac links are only published between ip-address and mac-address

This test case is for the following requirements:

[CTNC-MAP-SEN-REQ-8-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Expected Outcomes:

1. DHCP server sensor client publishes end-time metadata on an ip-mac link.
2. Sensor client publishes location metadata with at-least one location-information sub element.

Anticipated Failures:

1. If sensor client is not a DHCP server, test fails

2. If sensor client publishes location metadata with no location-information sub element, the test fails.

5.2 Non-AA Test Cases, NS Clients

5.2.1.1 unexpected-behavior

[CTNC-MAP-NS-CLNT-TC-1]

Purpose: To verify that client who publishes unexpected-behavior metadata publishes it using update. Also verify that client publishes metadata with its own publisher-id.

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-1-M]

[CTNC-MAP-NS-CLNT-REQ-3-M]

[CTNC-MAP-NS-CLNT-REQ-22-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements:

- Enable always available cases that satisfies the metadata validity tests
- Enable tracking of metadata to ensure that unexpected behaviour metadata is both published and deleted during a session.

Test Steps:

1. Begin tracking SOAP transactions contents from server side.
2. Client requests new session from server.
3. Client publishes unexpected-behavior metadata.
4. Clear the condition that caused the unexpected behavior.
5. Disconnect the client.

Expected Outcomes:

1. Publish succeeds.
2. Clearing the condition that caused the event results in the client deleting the metadata.

Anticipated Failures:

1.If client publishes unexpected-behavior metadata via notify, the test fails.

2.If upon client disconnect, the client has not deleted the unexpected-behavior metadata, the test fails.

3.If client sends delete request with any other ifmap-publisher-id (other than its own), the test fails.

5.2.1.2 MAP Clients that publish metadata from multiple sources

[CTNC-MAP-NS-CLNT-TC-2]

Purpose: To verify that client who publishes metadata from multiple sources, generates a unique discoverer-id for each source, in the form “ifmap-publisher-id:UID” where UID may be a simple, ordinal value. This test case only applies to NS Clients that publish metadata from multiple sources. This test case only applies to metadata that contains the discoverer-id element (device-characteristic, event, location).

This test case is for the following requirements:

[CTNC-MAP-NS-CLNT-REQ-4-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements:

- Enable AA tests to ensure validity of metadata
- Server will ensure that no two discoverer-id containing pieces of metadata contain the same discoverer-id

Test Steps:

1. Begin tracking SOAP transaction contents from server side.
2. Client requests new session from server.
3. A condition is triggered that causes the client to send a publish request containing device-characteristic, event, or location metadata from source A.
4. A condition is triggered that causes the client to send a publish request containing device-characteristic, event, or location metadata from source B.

5. Disconnect client.

Expected Outcomes:

1. Client creates unique discoverer-id for metadata based on information from Source A and for metadata based on information from Source B.

Anticipated Failures:

1. If Client uses the same discoverer-id for both pieces of metadata, the test fails.

2. If discoverer-id is not of the form “ifmap-publisher-id:UID” where UID is a simple, ordinal value, the test fails.

5.3 IF-MAP Metadata for Network Security Compliance Test Cases, PDP Clients

In the IF-MAP Metadata for Network Security Compliance Test Cases for PDP Clients, the Device Under Test (DUT) is a PDP operating as an IF-MAP Client. To verify that the IF-MAP Client correctly implements the specification, the test program will enter a test step state, and then require specific packets from the Client.

5.3.1.1 PDP successfully authenticates and deletes endpoint metadata

CTNC-MAP-PDP-TC-1

Purpose: To verify that PDP client publishes access-request-device, access-request-mac, access-request-ip and authenticated-by metadata when successfully authenticating an endpoint. Also verify that when an endpoint disconnects from the network, the PDP publishes a delete to the server of the metadata associated with that endpoint. PDP must also support lifetime of “session” which will also be verified.

This test case is for the following requirements:

[CTNC-MAP-PDP-REQ-1-M]

[CTNC-MAP-PDP-REQ-3-M]

[CTNC-MAP-PDP-REQ-4-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements:

- Server must keep track of all published identifiers and metadata, and issue warnings after the end of a session.

Test Steps:

1. Begin tracking SOAP transactions contents from server side.
2. Configure PDP to publish metadata with lifetime of “session,” if configurable.
3. Client requests new session from server.
4. Have a device (endpoint) authenticate itself through the PDP.

5. Disconnect access-requesting device from network/PDP.
6. Disconnect client.

Expected Outcomes:

1. Client publishes access-request-device, access-request-mac, access-request-ip and authenticated-by metadata when successfully authenticating an endpoint.

Anticipated Failures:

1. If PDP client fails to publish access-request-device, access-request-ip and authenticated-by metadata upon successful authentication of an endpoint,

2.If a PDP client fails to publish the MAC address upon successful authentication, the test script should issue a Warning message.

3.If a PDP client fails to publish access-request-device metadata on the link between the access-request identifier and the endpoint's device identifier, the test fails.

4.If a PDP client fails to publish access-request-ip metadata on the link between the access-request identifier and the endpoint's ip-address identifier, the test script should issue a Warning message.

5.If a PDP client fails to publish authenticated-by metadata on the link between the access-request identifier and the PDP's device identifier, the test fails.

6.If a PDP client fails to publish the access-request-mac metadata on the link between the access-request identifier and the endpoint's mac-address identifier, the test script should issue a Warning message.

7.If a PDP fails to delete the metadata associated with an endpoint when it disconnects from the network, the test fails.

8.If PDP fails to publish with lifetime=session, the test case fails.

NOTE: Since the test script cannot "know" whether the PDP is aware of certain metadata, a Warning message must be issued describing the condition that caused the warning; e.g Client authenticated without publishing MAC address. The person running the test must determine whether the PDP device was aware of the data and should have published it.

5.3.1.2 PDP Clients Send Conditional Metadata

CTNC-MAP-PDP-TC-2

Purpose: To verify that PDP client publishes required metadata when related information is known by the PDP.

This test case is for the following requirements:

[CTNC-MAP-PDP-REQ-2-M]

Preconditions:

- Devices configured to "Client Test Setup"..

Server Requirements:

- Server must keep track of all published identifiers and metadata, and issue warnings after the end of a session.

Test Steps:

1. Begin tracking SOAP transactions contents from server side.

2. Client requests new session from server.
3. PDP Client authenticates an endpoint, and learns metadata about the endpoint.
4. Disconnect client.

Expected Outcomes:

1. PDP clients publish, when known, the metadata in this requirement.

Anticipated Failures:

1. If PDP client does not publish identity identifiers to the MAP server, the test script should issue a Warning message.

2. If PDP client does not publish identity capabilities to the MAP server, the test script should issue a Warning message.

3. If PDP client does not publish device attributes to the MAP server, the test script should issue a Warning message.

4. If PDP client does not publish device characteristics to the MAP server, the test script should issue a Warning message.

5. If PDP client does not publish role information to the MAP server, the test script should issue a Warning message.

6. If PDP client does not publish layer-2 information to the MAP server, the test script should issue a Warning message.

7. If PDP client does not publish wlan information to the MAP server, the test script should issue a Warning message.

NOTE: Since the test script cannot “know” whether the PDP is aware of certain metadata, a Warning message must be issued describing the condition that caused the warning; e.g Client authenticated without publishing MAC address. The t person running the test must determine whether the PDP device was aware of the data and should have published it..

5.3.1.3 PDP with subscription capabilities takes remedial action

CTNC-MAP-PDP-TC-3

Purpose: To verify that PDP client which implements subscription capabilities takes remediation action against endpoints based on policy definitions.

This test case is for the following requirements:

[CTNC-MAP-PDP-REQ-5-M]

[CTNC-MAP-PDP-REQ-6-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements: none

Test Steps:

1. Begin tracking SOAP transaction contents from server side.
2. PDP Client requests new session from server.

3. Successfully authenticate an endpoint via PDP.
4. Tester uses script to publish a policy violation.
5. PDP learns about policy violation via subscription and poll.
6. PDP takes action against endpoint.
7. Disconnect PDP client.

Expected Outcomes:

1. PDP takes action against endpoint violating policy. Note that PDP action might typically be isolating the endpoint to a controlled VLAN until remediation of the policy violation can be done.

Anticipated Failures:

1. If PDP fails to take action on an endpoint that violates policy, test fails.

5.4 IF-MAP Metadata for Network Security Compliance Test Cases, Sensor Clients

In the IF-MAP Metadata for Network Security Compliance Test Cases for Sensor Clients, the Device Under Test (DUT) is a Sensor operating as an IF-MAP Client. To verify that the IF-MAP Client correctly implements the specification, the test program will enter a test step state, and then require specific packets from the Client. Some cases will require loading the MAP server with data before running the test; these pre-conditions will be specified in the test case.

5.4.1.1 Sensor client that responds to request-for-investigation

CTNC-MAP-SEN-TC-1

Purpose: To verify that a sensor that responds to a request-for-investigation is configurable to respond only to a particular set of qualifier strings. Also verify that the client only responds if the request is in the configured set.

This test case is for the following requirements:

[CTNC-MAP-SEN-REQ-11-M]

[CTNC-MAP-SEN-REQ-12-M]

[CTNC-MAP-SEN-REQ-13-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements: none

Test Steps:

1. Begin tracking SOAP transactions contents from server side.
2. Configure sensor to publish metadata with lifetime of “session,” if configurable.
3. Sensor client requests new session from server.
4. Sensor client supporting request-for-investigation is configured to respond to a set of qualifiers.
5. Tester uses script to publish request-for-investigation (no qualifiers).

6. Tester uses script to publish request-for-investigation with qualifier matching client configuration.

7. Tester uses script to publish request-for-investigation with qualifier not matching client configuration.

8. Disconnect client.

Expected Outcomes:

1. Client supporting request-for-investigation is configurable to respond to a set of qualifiers (Step 4).
2. Client returns configured qualifier metadata in Step 5 and Step 6.
3. Client does not return non-configured qualifier metadata in Step 7.

Anticipated Failures:

1. If sensor client supporting request-for-investigation is not configurable for qualifiers, the test fails.

2.If sensor client fails to respond to request-for-investigation without qualifier, or with specific configured qualifier, the test fails.

3.If sensor client responds with metadata for qualifiers not configured, the test fails.

5.4.1.2 Sensor Event Tests

CTNC-MAP-SEN-TC-2

Purpose: To verify that Sensor clients publish event metadata using “notify,” which must be attached to either the IP address or MAC address of the endpoint for which the event occurred. Sensor must also delete published event metadata when it no longer applies.

This test case is for the following requirements:

[CTNC-MAP-SEN-REQ-2-M]

[CTNC-MAP-SEN-REQ-3-M]

[CTNC-MAP-SEN-REQ-7-M]

Preconditions:

- Devices configured to “Client Test Setup”..
- Preload endpoint metadata onto IF-MAP server using test suite publishing utility

Server Requirements: none

Test Steps:

Expected Outcomes:

1. Event metadata about an endpoint is published to the MAP server when it is known.
2. Sensor deletes event metadata when a change in state of endpoint related to that published event occurs.

Anticipated Failures:

1. If sensor client detects event metadata and fails to publish to the IF-MAP server, the test fails.

2.If the sensor client fails to use “notify” to publish event metadata, the test fails.

3.If the sensor client fails to delete metadata for an endpoint when it no longer describes the state of the endpoint, the test fails.

5.4.1.3 Sensor Device-Characteristic Tests

CTNC-MAP-SEN-TC-3

Purpose: To verify that Sensor clients publish device characteristic metadata using “notify,” and that device characteristic metadata must be attached to either the IP address or MAC address of the endpoint, and be published on the link between the device identifier for the sensor and either an IP address or MAC address identifier. The sensor must generate a unique device identifier for itself. When published metadata is no longer valid, the sensor must delete it from the server.

This test case is for the following requirements:

[CTNC-MAP-SEN-REQ-2-M]

[CTNC-MAP-SEN-REQ-4-M]

[CTNC-MAP-SEN-REQ-5-M]

[CTNC-MAP-SEN-REQ-7-M]

Preconditions:

- Devices configured to “Client Test Setup”..
- Preload endpoint metadata onto IF-MAP server using test suite publishing utility

Server Requirements: none

Expected Outcomes:

1. Sensor published device characteristic metadata about an endpoint when it is known.
2. Sensor publishes changes to device characteristic data, when known.
3. Sensor deletes published metadata when it no longer applies.

Anticipated Failures:

1. If sensor client detects device characteristic metadata and fails to publish them to the IF-MAP server, the test fails.

2.If sensor client publishes device-characteristic metadata but fails to attach that metadata to either the ip-address or mac-address of the endpoint, the test fails.

3.If the sensor client fails to publish the device-characteristic metadata with a unique device identifier for itself, the test fails.

4.If the sensor fails to update device characteristic metadata when it changes, the test fails.

5.If the sensor client fails to delete metadata for an endpoint when it no longer describes the state of the endpoint, the test fails.

5.5 IF-MAP Metadata for Network Security Compliance Test Cases, Flow Controller Clients

In the IF-MAP Metadata for Network Security Compliance Test Cases for Flow Controller Clients, the Device Under Test (DUT) is a Flow Controller operating as an IF-MAP Client. To verify that the IF-MAP Client correctly implements the specification, the test program will enter a test step state, and then require specific packets from the Client. Some cases will require loading the MAP server with data before running the test; these pre-conditions will be specified in the test case.

5.5.1.1 Flow Controller Tests

CTNC-MAP-FC-TC-1

Purpose: To verify that Flow Controller clients implement the ability to subscribe to notifications, and apply policy based on poll results. Flow controllers must also subscribe to endpoint metadata when the endpoint attempts to send traffic, and provide access to endpoints based on policy definitions. Optionally, the Flow Controller can publish enforcement-report metadata. This test case is for the following requirements:

[CTNC-MAP-FC-REQ-1-M]

[CTNC-MAP-FC-REQ-2-M]

[CTNC-MAP-FC-REQ-3-M]

[CTNC-MAP-FC-REQ-4-M]

[CTNC-MAP-FC-REQ-5-M]

Preconditions:

- Devices configured to “Client Test Setup”..

Server Requirements: none

Test Steps:

1. Begin tracking SOAP transactions contents from server side.
2. Flow Controller client requests new session from server.
3. Configure Flow Controller to publish metadata with lifetime of “session,” if configurable.
4. Initiate a traffic flow through the Flow Controller from a previously-unused endpoint.
5. Tester will publish a policy violation event on the endpoint (e.g. no virus protection) using a test suite utility.
6. Disconnect client.

Expected Outcomes:

1. Flow controller subscribes to notifications, and applies policy based on poll results.
2. Flow controller subscribes to notifications for specified ip-addresses when they attempt to send traffic through the flow controller.
3. A Flow controller that publishes enforcement-report metadata publishes it on the link between the Flow controller’s device identifier and the ip-address or mac-address of the endpoint.

Anticipated Failures:

1. If the flow controller is not configurable to lifetime=session OR lifetime=session is not the default, the test fails.

2.If flow controller client fails to subscribe to notifications, the test fails.

3. If flow controller fails to poll MAP server for updates and notifications, the test fails.

4. If flow controller fails to take remediation action based on policy violation as detected in metadata received via a poll, the test fails.

5.If flow controller with enforcement-report capability does not publish an enforcement-report, the test fails.

6.If flow controller with enforcement-report capability fails to publish enforcement-report link between Flow Controller’s identifier and ip-address or mac-address of endpoint, the test fails.

5.6 IF-MAP Metadata for Network Security Compliance Test Cases, NS Servers

In the IF-MAP Metadata for Network Security Compliance Test Cases for NS Servers the Device Under Test (DUT) is the IF-MAP Server. To verify that the IF-MAP Server correctly implements the specification for supporting NS Clients running IF-MAP 1.0 and IF-MAP 1.1, the test program will enter a test step state, and then issue specific packets from the Client to the Server.

5.6.1.1 IF-MAP 2.0 Servers choose non-conflicting device identifiers for 1.1 clients

CTNC-MAP-NS-SERVER-TC-1

Purpose: To verify that IF-MAP servers supporting both IF-MAP 1.1 and IF-MAP 2.0 correctly translate

This test case is for the following requirements:

[CTNC-MAP-NS-SERVER-REQ-1-M]

[CTNC-MAP-NS-SERVER-REQ-2-M]

Preconditions:

- Devices configured to “Server Test Setup”.

Test Steps:

1. Begin tracking SOAP transactions contents from client side.
2. IF-MAP 1.1 client requests new session from server. and IF-MAP 2.0 client requests a new session from server using a different login
3. IF-MAP 2.0 client publishes authenticated by link between access-request(ar1) and device(dev1), and a device-ip link between device(dev1) and ip(1.2.3.4)
4. IF-MAP 1.1 client searches for access-request(ar1) with max-depth of 2
5. IF-MAP 1.1 client publishes authenticated-by link between access-request(ar2) and ip-address(5.6.7.8).
6. IF-MAP 2.0 client searches for access-request(ar2) with a max-depth of 2
7. IF-MAP 1.1 client publishes a layer2-information link between access-request(ar3) and ip-address(10.0.0.1).
8. IF-MAP 2.0 client searches for access-request(ar3) with a max-depth of 2
9. Disconnect IF-MAP 1.1 and 2.0 clients.

Expected Outcomes:

1. IF-MAP Server accepts published links in step 3, sends publish-received to client.
2. Search in step 4 results in the 1.1 client receiving an authenticated-by link between access-request(ar1) and ip(1.2.3.4)
3. IF-MAP Server accepts published links in step 5, sends publish-received to client.
4. Search in step 6 results in the 2.0 client receiving an authenticated-by link between access-request(ar2) and a generated device.
5. Search in step 6 results in the 2.0 client receiving a device-ip link between the generated device and ip(5.6.7.8)

6. Search in step 8 results in the 2.0 client receiving a layer2-informatin link between access-request(ar3) and a generated device
7. Search in step 8 results in the 2.0 client receiving a device-ip link between the generated device and ip(10.10.0.1)

Anticipated Failures:

1. If IF-MAP Server fails to create unique device-identifier name for IF-MAP 1.1 clients, the test case fails.

References

This section lists specifications and other documents that are referred to in the document. Since this document is informative (not normative), all of these references are informative with respect to this document.

Informative References

[1] Trusted Computing Group, TNC_IFMAP_Metadata_For_Network_Security, Specification Version 1.0 Revision 25, 13 September, 2010

[2] Trusted Computing Group, TNC IF-MAP Binding for SOAP, Specification Version 2.0, Revision 36, 30 July 2010.

[3] Trusted Computing Group, Trusted Network Connect TNC Certification Strategy, Specification Version 1.0, Revision 0.05, 5 June, 2007 (Draft)

[4] Trusted Computing Group, TNC Compliance and Interoperability Principles, Specification Version 1.0, Revision 0.06, 2 November, 2006 (Draft)

[5] Trusted Computing Group, TNC Architecture for Interoperability with IF-MAP, Specification Version 1.1, Feb 2010.