TCG Trusted Network Connect

TNC IF-MAP Metadata for Network Security

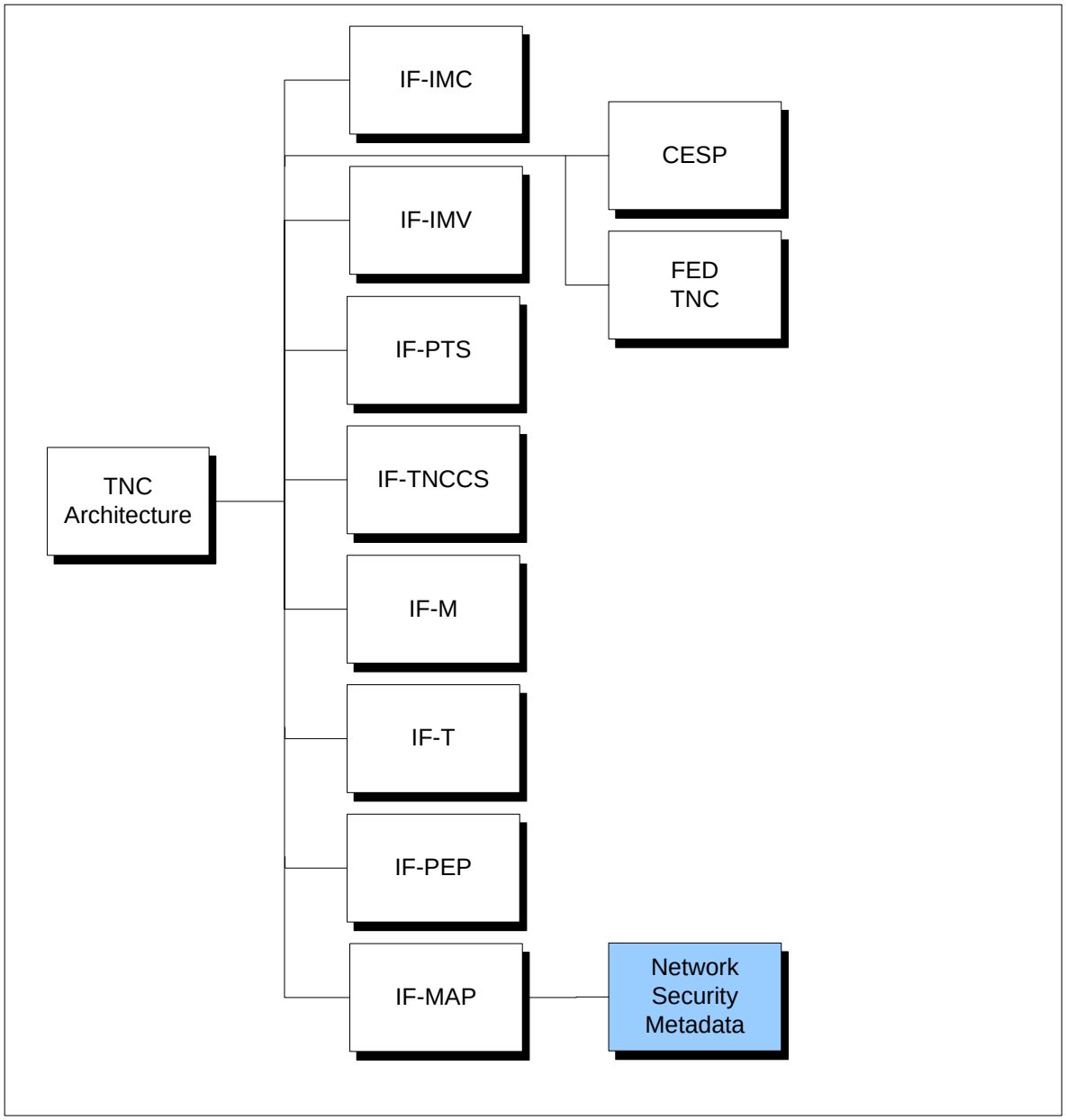
Specification Version 1.0
Revision 25
13 September 2010
Published

Contact:
admin@trustedcomputinggroup.org

TCG PUBLISHED

TCG

IWG TNC Document Roadmap



Acknowledgements

The TCG wishes to thank all those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Scott Kelly	**Aruba Networks**
Amit Agarwal	**Avaya**
Mahalingam Mani	**Avaya**
Craig Dupler	**Boeing Corporation**
David Mattes	**Boeing Corporation**
Steven Venema	**Boeing Corporation**
Eric Byres (Invited Expert)	**Byres Security**
Mark Townsend	**Enterasys**
Michael McDaniels	**Extreme Networks**
Ingo Bente	**Fachhochschule Hannover**
Arne Welzel	**Fachhochschule Hannover**
Hidenobu Ito	**Fujitsu Limited**
Seigo Kotani	**Fujitsu Limited**
Houcheng Lee	**Fujitsu Limited**
Sung Lee	**Fujitsu Limited**
Graeme Proudler	**Hewlett-Packard**
Mauricio Sanchez	**Hewlett-Packard**
Han Yin	**Huawei Technologies**
Diana Arroyo	**IBM**
Guha Prasad Venkataraman	**IBM**
Sean Convery	**Identity Engines**
Chris Hessing	**Identity Engines**
Morteza Ansari	**Infoblox**
Stuart Bailey (Co-Editor)	**Infoblox**
Andrew Benton	**Infoblox**
Tom Clark	**Infoblox**
Peter Lee	**Infoblox**
Rod Murchison	**Infoblox**
Ivan Pulleyn	**Infoblox**
David Vigier	**Infoblox**
Rena Yang	**Infoblox**
Ravi Sahita	**Intel Corporation**
Ned Smith	**Intel Corporation**
Josh Howlett	**JANET(UK)**
Yan Avlasov	**Juniper Networks**
Roger Chickering (Co-Editor)	**Juniper Networks**
Charles Goldberg	**Juniper Networks**
Steve Hanna (TNC co-chair)	**Juniper Networks**
Clifford Kahn	**Juniper Networks**
PJ Kirner	**Juniper Networks**
Lisa Lorenzin (Co-Editor)	**Juniper Networks**
John Jerrim	**Lancope**
Mark Labbancz	**Lumeta**
Matt Webster	**Lumeta**
Bill Nemec	**Lumeta**
Eric Fitzgerald	**Microsoft**
Ryan Hurst	**Microsoft**

Sandilya Garimella	**Motorola**
Meenakshi Kaushik	**Nortel**
Paul Sangster (TNC co-chair)	**Symantec Corporation**
Ted Fornoles	**Trapeze Networks**
Matthew Gast	**Trapeze Networks**
Tim McCarthy	**Trapeze Networks**
Jeffrey Peden	**Trapeze Networks**
Brian Wangerian	**Trapeze Networks**
Brad Upson	**UNH InterOperability Lab**
Mike Boyle	**US National Security Agency**
Lauren Giroux	**US National Security Agency**
Chris Salter	**US National Security Agency**
Thomas Hardjono	**Wave Systems**
Greg Kazmierczak	**Wave Systems**

Special thanks to the members of the TNC contributing to this document:

David Mattes	Boeing
Steve Venema	Boeing
Peter Lee	Infoblox
David Vigier	Infoblox
Steve Hanna	Juniper
Clifford Kahn	Juniper
Matt Webster	Lumeta
Matthew Gast	Trapeze
Jeffrey Peden	Trapeze
Mike Boyle	US National Security Agencey

Table of Contents

1 Introduction 8
1.1 Scope and Audience 8
1.2 Keywords 8
2 Background 9
2.1 Purpose of IF-MAP Metadata for Network Security 9
2.2 Supported Use Cases 9
2.3 Requirements 10
3 IF-MAP Metadata for Network Security 11
3.1 IF-MAP Metadata For Network Security Types 11
3.1.1 access-request-device *(Link)* 12
3.1.2 access-request-ip *(Link)* 12
3.1.3 access-request-mac *(Link)* 13
3.1.4 authenticated-as *(Link)* 13
3.1.5 authenticated-by *(Link)* 14
3.1.6 capability 14
3.1.7 device-attribute *(Link)* 15
3.1.8 device-characteristic *(Link)* 15
3.1.9 device-ip *(Link)* 16
3.1.10 discovered-by *(Link)* 17
3.1.11 enforcement-report *(Link)* 17
3.1.12 event 18
3.1.13 ip-mac *(Link)* 20
3.1.14 layer2-information *(Link)* 21
3.1.15 location 22
3.1.16 request-for-investigation *(Link)* 23
3.1.17 role *(Link)* 23
3.1.18 wlan-information *(Link)* 24
3.1.19 unexpected-behavior 25
4 IF-MAP Operations 27
4.1 MAP Client Operations 27
4.1.1 PDP Operations 27
4.1.2 Sensor Operations 28
4.1.3 Flow Controller Operations 29
5 Security Considerations 30
6 Privacy Considerations 31
7 Recommendations for Backward Compatibility 32
8 References 34
9 Examples 35
9.1 Network Diagram 35
9.2 Example 1 35
9.3 Example 2 43
9.4 Example 3 45
9.5 Example 4 48
9.6 Example 5 51
9.7 Example 6 53
9.8 Example 7 54
9.9 Example 8 59
9.10 Example 9 60
9.11 Example 10 61
9.12 Example 11 61
9.13 Example 12 62
9.14 Example 13 66
9.15 Example 14 68

10 IF-MAP Metadata for Network Security Schema 75
10.1 Standard Metadata Types 75
Appendix A: Device Types 83

1 Introduction

1.1 Scope and Audience

The Trusted Network Connect Working Group (TNC-WG) has defined an open solution architecture that enables network operators to enforce policies regarding the security state of endpoints in order to determine whether to grant access to a requested network infrastructure. Part of the TNC architecture is IF-MAP, a standard interface between the Metadata Access Point and other elements of the TNC architecture. This document defines and specifies IF-MAP Metadata for Network Security.

Architects, designers, developers and technologists who wish to implement, use, or understand IF-MAP in a network security context should read this document carefully. Before reading this document any further, the reader should review and understand the TNC architecture as described in [1] and the TNC IF-MAP Binding for SOAP[3].

1.2 Keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [2]. This specification does not distinguish blocks of informative comments and normative requirements. Therefore, for the sake of clarity, note that lower case instances of must, should, etc. do not indicate normative requirements.

2 Background

2.1 Purpose of IF-MAP Metadata for Network Security

IF-MAP Metadata for Network Security enables a set of network devices from disparate vendors to work together to enhance the security of a network. IF-MAP Metadata for Network Security includes use cases for endpoints, including Access Requestors (ARs) as well as Clientless Endpoints (CEs) that do not run access requestor software, that request access to the network.

2.2 Supported Use Cases

Use cases that this version of IF-MAP Metadata for Network Security supports:

- A Policy Decision Point (PDP) publishes endpoint authentication, VLAN, MAC address, and other status such as device health and/or wireless LAN information to a MAP.

- A DHCP server assigns an IP address to an endpoint with a certain lease duration. The DHCP server publishes ip-mac metadata associated with the endpoint to the MAP Server. The ip-mac metadata includes an attribute that specifies that the ip-mac metadata expires at the same time that the DHCP lease expires.

- A Flow Controller detects a previously undetected flow from an endpoint and queries a MAP Server to obtain authentication and compliance status associated with this endpoint in order to make enforcement decisions about the new flow.

- A Flow Controller subscribes to notifications from a MAP Server about changes in authentication, compliance, vulnerability, or other status for an endpoint so the Flow Controller can make appropriate enforcement adjustments to an existing flow.

- A Sensor publishes information related to an endpoint (such as authentication, VLAN, IP address, and/or device characteristics) or flows originated from an endpoint (vulnerability detection, flow classification, flow compliance, location, etc) to a MAP Server.

- A PDP queries a MAP for metadata that a MAP Client has associated with an endpoint (flow classification, vulnerability information, location, etc.). The PDP uses the metadata to make policy decisions. The PDP subscribes to notifications from the MAP Server about changes to the endpoint's metadata so the PDP can adjust the endpoint's access when the endpoint's metadata changes.

- A Sensor (such as an intrusion detection system or network behavioral anomaly detection system) sends information related to an attack or anomalous traffic originated from an endpoint to a MAP Server. The MAP Server forwards the information to other MAP Clients that have subscribed to attack or anomalous traffic notifications. The MAP Server does not store the information about the attack or anomalous traffic notification and thus the MAP Clients never receive delete notifications about the attack or anomalous traffic. This multicast interaction reflects the fact that the attack is an instantaneous event and not a persistent attribute of the network.

- A PDP or Flow Controller takes enforcement action against an endpoint, and publishes metadata indicating that the enforcement action is in effect. This information can be used by an administrator troubleshooting user access problems. When the enforcement action is no longer in effect, the PDP or Flow Controller deletes the metadata.

- A Sensor (such as an endpoint profiling or behavior monitoring system) sends information related to anomalous behavior by an endpoint to a MAP Server. The MAP Server forwards

the information to other MAP Clients that have subscribed to notifications for that endpoint. When the endpoint goes back to normal behavior, the Sensor deletes the metadata.

- A PDP requests investigation of a clientless endpoint via the MAP. The MAP notifies the available Sensor(s) of the request for investigation; the Sensor(s) conduct investigations and publish the results to the MAP. The PDP may then modify its initial access control decision based on the investigation metadata..

2.3 Requirements

The following are the requirements which IF-MAP Metadata for Network Security must meet in order to successfully play its role in the TNC architecture. These are stated as general requirements, with specific requirements called out as appropriate.

1. **Meets the needs of the TNC architecture**

   IF-MAP Metadata for Network Security must support all the functions and use cases described in the TNC architecture as they apply to characterizing endpoints in MAP.

   Specific requirements include:

   - IF-MAP Metadata for Network Security must enable a Policy Decision Point to publish information about an endpoint that other devices in the network can use to make security decisions.
   - IF-MAP Metadata for Network Security must enable a Sensor to publish information about an endpoint that other devices in the network can use to make security decisions.

2. **Easy to use and implement**

   IF-MAP Metadata for Network Security should be easy for vendors to use. It should allow them to enhance existing products to support the TNC architecture and integrate legacy code without requiring substantial changes.

3. **Unambiguous**

   There should be clarity and lack of ambiguity for identification of specific entities for which metadata exists and which are interacting with the MAP Server. For example users, endpoints and all other instances of TNC elements should be uniquely identifiable within an IF-MAP system containing a MAP Server and its associated MAP Clients..

3 IF-MAP Metadata for Network Security

IF-MAP Metadata for Network Security defines a standard set of metadata for use in determining information about an endpoint such as security status, location, behavior, etc. Some attributes of an endpoint, such as device-characteristics of manufacturer or model, may be nearly static over time; other attributes, such as location or wlan-information, may change frequently. The network security metadata can be used by network elements to determine whether to allow the endpoint to access a requested network and what level of access is appropriate.

The figure below represents valid identifier and metadata relationships that could arise on a MAP during a network security scenario. For descriptions of each standard metadata type, see Section 3.1. For more details on how this state could emerge on a MAP, see Section 9. In the figure, ovals represent identifiers, lines between ovals represent links, and rectangles on links represent metadata attached to the link. A line between an oval and a rectangle indicates metadata attached to the identifier.

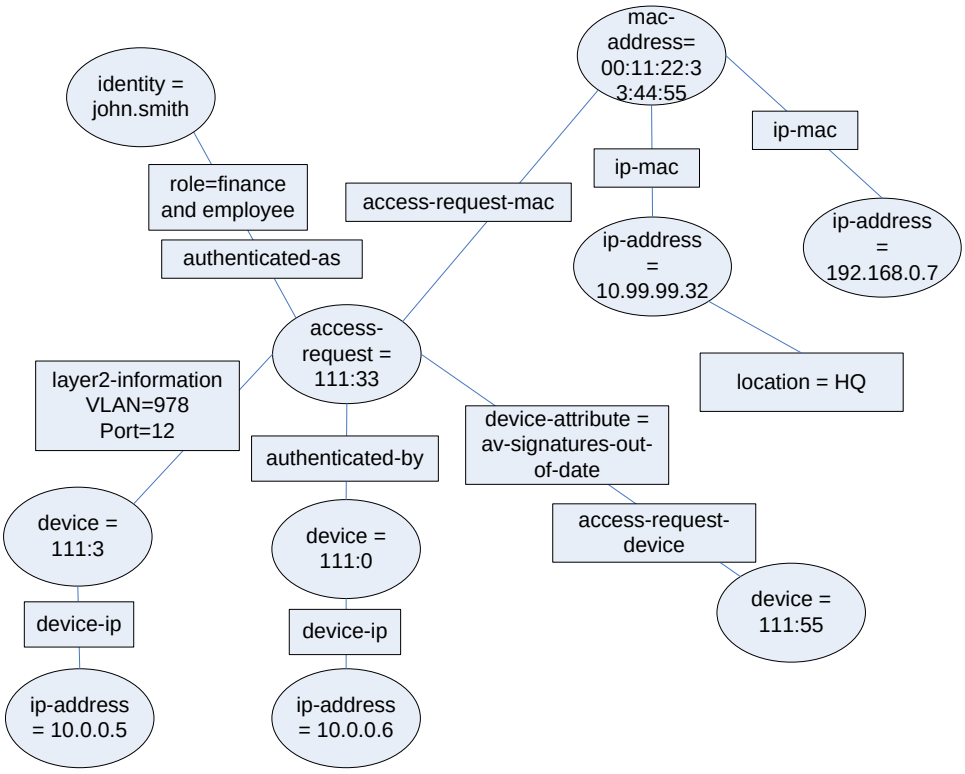


Figure 1

3.1 IF-MAP Metadata For Network Security Types

Note that the following metadata types are intended for use in match-links filters:

- access-request-device
- access-request-mac
- access-request-ip
- authenticated-as
- authenticated-by
- device-ip
- discovered-by
- enforcement-report

These metadata types 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. Other metadata types may be applied more broadly; recommendations are provided for common use cases.

Several elements may take vendor-defined or TCG-defined types. For these elements, the type field's value MUST take one of two forms:

1. "Vendor-ID:Type": A vendor-defined type. Vendor-ID is the 24-bit SMI Private Enterprise Number Vendor ID[5] of the party specifying the type, and Type is the type specified.
2. "Type": A TCG-defined type. A TCG-defined type may be specified in a future version of IF-MAP Metadata for Network Security or in a supplement to IF-MAP Metadata for Network Security, as specified in the type description.

A MAP Client that publishes multi-valued metadata has a responsibility to maintain the accuracy of this metadata; publishing a new value for a multi-valued attribute does not automatically delete the old values. Therefore, when publishing new multi-valued metadata, the MAP Client SHOULD, in the same publish request, remove any metadata with its ifmap-publisher-id that has gone stale or is no longer relevant. The MAP Client MUST NOT delete metadata with any other ifmap-publisher-id.

In some cases, a MAP Client such as a Security Information and Event Management (SIEM) system may serve as an aggregator, publishing metadata based on information from multiple network security devices. For metadata types such as device-characteristic, event, and location, the discoverer-id attribute is intended to accurately delineate the original source of the information. A MAP Client SHOULD use its ifmap-publisher-id as its discoverer-id if it is only publishing metadata that it generated itself. 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.

3.1.1 access-request-device *(Link)*

Clients MUST publish this only between: access-request and device

access-request-device metadata is attached to a link to associate an access-request identifier with a device identifier. It is a simple metadata type and does not have any content. Note that the device identifier in this case represents the endpoint, rather than the element creating the metadata.

A PDP may create the association after provisioning network access for an access-request. A Flow Controller may match links marked with access-request-device to search for device-attribute metadata. A MAP Client SHOULD only publish access-request-device metadata on a link adjacent to an access-request identifier or a device identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="access-request-device">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.2 access-request-ip *(Link)*

Clients MUST publish this only between: access-request and ip-address

access-request-ip metadata is attached to a link to associate an access-request identifier with an ip-address identifier. It is a simple metadata type and does not have any content.

A PDP or other authenticating element may create the association when an ip-address is provisioned for an access-request. A Flow Controller may attempt to match links marked with access-request-ip to search for capability, device-attribute, and role metadata associated with a specific ip-address when making access control decisions. A MAP Client SHOULD only publish access-request-ip metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="access-request-ip">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.3 access-request-mac *(Link)*

Clients MUST publish this only between: access-request and mac-address

access-request-mac metadata is attached to a link to associate a specific access-request identifier with a specific mac-address identifier. It is a simple metadata type and does not have any content.

A PDP or PEP may create the association when an endpoint is authenticated and granted access. A Flow Controller may attempt to match links marked with access-request-mac to search for capability, device-attribute, and role metadata associated with a specific mac-address when making access control decisions. A DHCP server may attempt to match links marked with access-request-mac to search for access-requests when making decisions regarding provisioning IP addresses to endpoints. A MAP Client SHOULD only publish access-request-mac metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="access-request-mac">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.4 authenticated-as *(Link)*

Clients MUST publish this only between: access-request and identity

authenticated-as metadata is attached to a link to associate a specific access-request identifier with a specific identity. It is a simple metadata type and does not have any content.

A PDP or other authenticating element may create the association when an endpoint is authenticated and granted access. A Flow Controller may attempt to match links marked with authenticated-as metadata to search for roles associated with an identity when making access control decisions. A MAP Client SHOULD only publish authenticated-as metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

An access-request SHOULD have only one authenticated-as link. If an access-request originates from a multi-user endpoint, a PDP may be able to determine that more than one user is using the endpoint. However, creating multiple authenticated-as links, one for each user, may have the effect of combining the roles of all users for the purpose of access control decisions regarding the endpoint. Properly handling multi-user endpoints is beyond the scope of this specification.

```
<xsd:element name="authenticated-as">
  <xsd:complexType>
    <xsd:attributeGroup
       ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.5 authenticated-by *(Link)*

Clients MUST publish this only between: access-request and device

authenticated-by metadata is attached to a link to associate a specific access-request identifier with a specific device identifier representing a PDP or other authenticating element. It is a simple metadata type and does not have any content.

A PDP or other authenticating element may create the association when an endpoint is authenticated and granted access. A MAP Client may attempt to match links marked with authenticated-by to subscribe for changes associated with endpoints authenticated by a specific PDP when making access control decisions. A MAP Client SHOULD only publish authenticated-by metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="authenticated-by">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.6 capability

Recommended for: access-request

capability metadata refers to a collection of privileges assigned to an endpoint as a result of an access request. Privileges are defined outside the scope of this specification; however, capability is the name used in IF-MAP to reference those privileges for use in controlling access.

For example, an organization might define Trusted and Untrusted capabilities which are attached to an access-request. A PDP or Flow Controller may use capabilities from an access-request identifier when making access control decisions.

Different parts of an organization may be empowered to designate capabilities. In order to distinguish the same capability name assigned by different parts of the organization, an optional administrative-domain may be specified to further qualify capability names.

A PDP that provisions access to an endpoint SHOULD publish capability metadata (see example section 9.15). A MAP Client other than a PDP or other authenticating element that provisions access to an endpoint SHOULD NOT publish capability metadata. A MAP Client SHOULD only publish capability metadata on an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="capability">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="1"
        maxOccurs="1"/>
      <xsd:element name="administrative-domain" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.7 device-attribute *(Link)*

Recommended for: access-request and device

A device-attribute is an attribute assigned to a specific endpoint. device-attribute metadata links a specific access-request identifier with the device identifier of the endpoint making the access request. Note that the device identifier in this case represents the endpoint, rather than the element creating the metadata.

The values of device-attribute metadata are defined outside the scope of this specification; however, device-attribute is used in IF-MAP Metadata for Network Security to reference those values for use in controlling access.

device-attribute metadata is meant to be a function, in part, of the organization's *policies*, such as policies about required software updates.For example, an organization might define AntiVirusRunning and PatchesUpToDate attributes which are attached to the link between an access-request identifier and a device identifier. A PDP or Flow Controller may use those attributes when making access control decisions.

device-attribute metadata SHOULD be published by a PDP that provisioned access to an endpoint (see example section 9.5). A MAP Client other than a PDP that provisions access to an endpoint SHOULD NOT publish device-attribute metadata. A MAP Client SHOULD only publish device-attribute metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="device-attribute">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string" minOccurs="1"
        maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.8 device-characteristic *(Link)*

Recommended for: access-request and device, ip-address and device, or mac-address and device

device-characteristic is metadata assigned to a specific endpoint by a MAP Client (usually a PDP or Sensor) to reflect an inherent characteristic of that endpoint, such as its manufacturer or what

operating system it is running, along with what element discovered the information and what method of discovery was used. device-characteristic metadata links an access-request identifier with a device identifier representing a PDP, Sensor, or other element that discovered the characteristic.

device-characteristic metadata differs from device-attribute metadata in that it describes the nature of an endpoint, rather than the endpoint's relationship to an organization's policies; it is an organization-independent statement about the endpoint itself.

For the 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. For the device-type element, TCG-defined types are listed in Appendix A: Device Types.

The discovered-time element refers to the time at which this device-characteristic was first detected. The discoverer-id element is a machine-generated string uniquely identifying the element that discovered the characteristic.

device-characteristic metadata SHOULD be published to the access-request and device link (see example section 9.15) by a MAP Client that discovered information about an endpoint.

```
<xsd:element name="device-characteristic">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="manufacturer" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="model" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="os" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="os-version" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="device-type" type="xsd:string"
        minOccurs="0"/>
      <xsd:element name="discovered-time" type="xsd:dateTime"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discoverer-id" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discovery-method" type="xsd:string"
        minOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.9 device-ip *(Link)*

Clients MUST publish this only between: device and ip-address

A device-ip link means that the specified device has the specified IP address. device-ip is a simple metadata type and does not have any content. A multi-homed device may have more than one IP address. A set of devices that are modeled as one device – a cluster – may likewise have more than one IP address.

An authenticating element may publish device-ip links to announce its IP addresses.

device-ip links are not intended for use with endpoints, only with authenticating elements. Access requests SHOULD have access-request-ip links, not device-ip links.

```
<xsd:element name="device-ip">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.10 discovered-by *(Link)*

Clients MUST publish this only between: ip-address and device or mac-address and device

discovered-by metadata is attached to a link to associate an ip-address identifier or mac-address identifier with a device identifier representing a Sensor that has noticed that endpoint on the network. It is a simple metadata type and does not have any content.

discovered-by metadata is intended to support the request-for-investigation use case. A PDP can subscribe to a known Sensor and listen for discovered-by notifications. As new endpoints are discovered, the PDP can publish a request for investigation, if desired.

A Sensor may create the association when it discovers an endpoint. A PDP or Sensor (e.g. analytics device) may attempt to match links marked with discovered-by to subscribe for changes associated with specific Sensors when making access control decisions. A MAP Client may publish discovered-by metadata on a link adjacent to an ip-address identifier or mac-address identifier generated by a different MAP Client.

```
<xsd:element name="discovered-by">
  <xsd:complexType>
    <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.11 enforcement-report *(Link)*

Clients MUST publish this only between: ip-address and device or mac-address and device

enforcement-report metadata is attached to a link to associate a specific mac-address identifier or ip-address identifier with a specific device identifier representing a PEP or Flow Controller. A Flow Controller may create the association when it takes enforcement action against an endpoint. A PDP may create the association when it instructs a PEP to take enforcement action against an endpoint.

The following enforcement-action types are supported:

- block: The Flow Controller denied access to the resource, or the PEP denied access to the network, in response to an access request by the endpoint.
- quarantine: The PEP restricted access to the network, e.g. by VLAN or filter assignment, in response to an access request by the endpoint.
- other: Extension point for other kinds of enforcement actions. If a MAP Client specifies enforcement-action type as other, the client MUST specify a non-empty string for the other-type-definition field. 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.

The enforcement-reason is an optional element to allow the PDP or Flow Controller to specify the cause of the action taken against the endpoint.

A PDP or Sensor may attempt to match links marked with enforcement-action to subscribe for changes associated with specific endpoints which have been affected by network enforcement policies; the PDP may additionally inform the endpoint of the results if desired.

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). A MAP Client may publish enforcement-report metadata on a link adjacent to an ip-address identifier or mac-address identifier generated by a different MAP Client.

```
<xsd:element name="enforcement-report">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="enforcement-action" type="xsd:string"
        minOccurs="1" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="block"/>
            <xsd:enumeration value="quarantine"/>
            <xsd:enumeration value="other"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="other-type-definition" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="enforcement-reason" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.12 event

Recommended for: access-request, identity, ip-address or mac-address

event metadata refers to activity of interest detected on the network. Examples include network traffic that matches the profile of a virus attack, excessive network traffic originating from a particular endpoint, and the use of a specific protocol such as an Instant Messaging protocol.

The discovered-time element refers to the time at which this event was first detected. The time of publication of the metadata to the MAP Server provides some temporal data, but does not accurately reflect the start time. For example, by the time an event is correlated and reported by the IDS/IPS to the MAP Server, several seconds, minutes, or even hours may have passed since the first action on the part of the endpoint that resulted in the event being created.

The discoverer-id element is a machine-generated string uniquely identifying the element that discovered the characteristic.

The magnitude element indicates how widespread the effects of the activity are. Magnitude ranges from 0 to 100, with higher magnitudes indicating more widespread effects.

The confidence element indicates how confident the MAP Client that published the event is that it accurately describes the activity of interest. Confidence ranges from 0 to 100, with higher values indicating higher confidence.

The significance element indicates how important the event is.

The following event types are supported:

- p2p: Peer-to-peer traffic was detected
- cve: Common Vulnerabilities and Exposures. Event types of “cve” MUST identify the vulnerability using the vulnerability-uri element.
- botnet infection: The endpoint is infected with a botnet and appears to be under remote control
- worm infection: The endpoint is infected with a worm which is trying to infect other endpoints in the network
- excessive flows: The endpoint is engaged in an unreasonable or unexpected amount of network traffic
- behavioral change: The endpoint’s profile has changed. For example, an endpoint which initially behaved like an IP phone is now behaving like a personal computer.
- policy violation: The endpoint has violated a policy. For example, the network policy may forbid the use of instant messaging, and a Sensor detected instant messaging traffic.
- other: Extension point for other kinds of events. If a MAP Client specifies event type as other, the client MUST specify a non-empty string for the other-type-definition field. 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.

The value of the information element is a human consumable informational string. Any machine consumable information should be put into a vendor-specific metadata schema rather than in the information element.

The vulnerability-uri element is used for events with type cve to indicate which vulnerability was detected.

Additional details about the event may be added by vendor and implementation specific metadata. To accomplish this, a MAP Client publishes auxiliary metadata from a vendor-specific schema, and correlates the event metadata with the vendor-specific metadata using the name element and ifmap-publisher-id attribute of the event metadata.

MAP Clients that publish events are responsible for aggregating events to limit the number of events published for a particular policy violation in order to avoid flooding the MAP Server with redundant events. 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 [3]. MAP Clients that use version 1.x of the TNC IF-MAP Binding for SOAP, and therefore publish events using the update operation, SHOULD delete the events when they no longer apply.

A particular ip-address or mac-address identifier may have event metadata attached to it from multiple MAP Clients.

```
<xsd:element name="event">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="name" type="xsd:string"
       minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discovered-time" type="xsd:dateTime"
       minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discoverer-id" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="magnitude" minOccurs="1" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100"/>
          </xsd:restriction>
```

```
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="confidence" minOccurs="1" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="significance" minOccurs="1"
        maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="critical"/>
            <xsd:enumeration value="important"/>
            <xsd:enumeration value="informational"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="type" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="p2p"/>
            <xsd:enumeration value="cve"/>
            <xsd:enumeration value="botnet infection"/>
            <xsd:enumeration value="worm infection"/>
            <xsd:enumeration value="excessive flows"/>
            <xsd:enumeration value="behavioral change"/>
            <xsd:enumeration value="policy violation"/>
            <xsd:enumeration value="other"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="other-type-definition" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="information" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="vulnerability-uri" type="xsd:anyURI"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.13 ip-mac *(Link)*

Recommended for: ip-address and mac-address

ip-mac is a binding between an ip-address and a mac-address which is valid for a specific time duration and optionally sanctioned by a specific DHCP server.

A DHCP server or other Sensor can attach ip-mac to a link between an ip-address and a mac-address when leasing that IP. Sensors other than DHCP servers MUST NOT publish end-time metadata on ip-mac links.

If an endpoint has one network adapter with more than one IP address, a MAP Client may publish an ip-mac link between each ip-address and the adapter's MAC address.

A particular ip-address identifier may have links to multiple mac-address identifiers with ip-mac metadata. Similarly, a particular mac-address identifier may have links to multiple ip-address identifiers with ip-mac metadata. The links may be published by different MAP Clients. Multiple MAP Clients may publish ip-mac metadata to the same link.

```
<xsd:element name="ip-mac">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="start-time" type="xsd:dateTime"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="end-time" type="xsd:dateTime"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="dhcp-server" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.14 layer2-information *(Link)*

Recommended for: access-request and device

layer2-information is attached to a link between an access-request and the device identifier of the PEP through which access is occurring. layer2-information includes vlan, which specifies the VLAN assigned to the access request; port, which specifies the port on the layer 2 PEP that the access-request originates from; and an optional administrative-domain, which may be used to distinguish between two instances of the same VLAN number in different parts of a network.

layer2-information metadata is typically published by a PDP when assigning the endpoint to a specific VLAN. A MAP Client SHOULD only publish layer2-information metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="layer2-information">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="vlan" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="vlan-name" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="port" type="xsd:integer"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="administrative-domain" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.15 location

Recommended for: identity, ip-address or mac-address

The location metadata element represents a named region of space – usually a region with security import. The region may be contiguous or discontiguous and may have any shape and boundaries as defined by an organization.

location metadata may be published by a Sensor acting as a MAP Client to provide information on the current location and changes in location for a user or device requesting access to the network. Examples of Sensors providing location may include location tracking systems (e.g. WiFi RTLS) and physical security access systems (e.g. proximity badge reader).

A PDP or other network device may leverage that information to apply location-based security policies and provision network access only for users physically present and authorized to be in a location, and may help to determine access control policy (such as VLAN assignment) based on a combination of user authentication and user location.

A Sensor publishing location metadata MUST include at least one location-information subelement that provides contextual data about the location of the user or device. A Sensor MAY provide additional detail known about the location of a user or device by including more than one location-information subelement in the location element. A PDP MAY use these location-information elements to further refine local policy decisions. The types and values of attributes are defined outside the scope of this specification: types may be defined by a Sensor vendor or by the deploying organization; values SHOULD be left to be specified by the deploying organization.

A Sensor SHOULD delete the location metadata for an identifier when it can no longer be said that the device or user is in that named location.

A Sensor SHOULD publish both a delete and an update of the location metadata for an identifier when a device or user has been detected to have moved from one named location to another.

The location type is defined as having a multiValue ifmap-cardinality, and therefore multiple location metadata elements may be published for a single device or user. These may be published by a single or multiple MAP Clients, and may be overlapping or non-overlapping in the physically or organizationally defined sense. 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.

```
<xsd:element name="location">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="location-information"
        minOccurs="1" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:attribute name="type" type="xsd:string"/>
            <xsd:attribute name="value" type="xsd:string"/>
          </xsd:complexType>
        </xsd:element>
      <xsd:element name="discovered-time" type="xsd:dateTime"
       minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discoverer-id" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
```

```
</xsd:element>
```

3.1.16 request-for-investigation *(Link)*

Recommended for: device and mac-address or device and ip-address

request-for-investigation metadata indicates that specified device, which may be a PDP or other MAP Client, wants Sensors to publish device-characteristic metadata about the specified MAC or IP address. The link may include a qualifier to indicate what investigation should be done. The values of the qualifier are defined outside the scope of this specification.

The requesting device SHOULD be an authenticating element or another Sensor. A device SHOULD attach request-for-investigation metadata only to a link adjacent to a device identifier that the device generated itself.

A Sensor may subscribe for request-for-investigation starting at an IP address of a requesting device. When a Sensor notices a request-for-investigation, it may attempt to learn about the ip-address or mac-address adjacent to the request-for-investigation link and may publish device-characteristic metadata associated with that address identifier (see example section 9.15).

Any IF-MAP Client that responds to request-for-investigation metadata MUST be configurable to respond only to a particular set of qualifier strings. 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. If the request-for-investigation does not contain a qualifier, any MAP Client may respond.

request-for-investigation metadata SHOULD be published using notify rather than using update. In case a Sensor is momentarily down at the time the request is published, the publisher may republish the request-for-investigation after a timeout period (such as one minute) if it did not get a response to the original request-for-investigation. A Sensor SHOULD republish only once or twice.

It can be useful to have more than one Sensor subscribe to a given device's request-for-investigation metadata. First, this can improve availability. Second, different kinds of Sensors may report different characteristics. Third, if Sensors detect the same characteristics using different methods, they may usefully confirm or cast doubt on each others' results.

```
<xsd:element name="request-for-investigation">
    <xsd:complexType>
      <xsd:attribute name="qualifier"
        type="xsd:string" use="optional"/>
      <xsd:attributeGroup
        ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>
```

3.1.17 role *(Link)*

Recommended for: access-request and identity

A role is the name of a collection of privileges assigned to a specific access-request and identity. Privileges are defined outside the scope of this specification; however, role is a name used in IF-MAP Metadata for Network Security to reference those privileges for use in controlling access.

For example, an organization might designate Employee and Contractor roles which are attached to links between access-request and identity identifiers. A PDP or Flow Controller may use roles from these links when making access control decisions.

Different parts of an organization may be empowered to designate roles. In order to distinguish the same role name assigned by different parts of the organization, an optional administrative-domain may be specified to further qualify role names.

role metadata SHOULD be published by a PDP that provisioned access to an endpoint. A MAP Client other than a PDP that provisions access to an endpoint SHOULD NOT publish role metadata. A MAP Client SHOULD only publish role metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:element name="role">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="administrative-domain"
        type="xsd:string" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="name" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

3.1.18 wlan-information *(Link)*

Recommended for: access-request and device

wlan-information is attached to a link to associate a specific access-request identifier with a specific device identifier representing the PEP through which access is occurring. wlan-information includes information on the SSID to which an endpoint has connected, and security information for the SSID that provides information on the encryption used for various types of frames by the endpoint. Due to limitations in 802.11 security handshakes, security information can be published only after a session has been successfully established.

wlan-information metadata is typically published by a PDP when an endpoint has completed the 4-Way Handshake and negotiated cryptographic parameters for an 802.11 link. A MAP Client SHOULD only publish wlan-information metadata on a link adjacent to an access-request identifier generated by a different MAP Client in cases with explicit coordination between clients about how sharing is handled. The nature of that coordination is beyond the scope of this specification.

```
<xsd:simpleType name="wlan-security-enum">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="open"/>
    <xsd:enumeration value="wep"/>
    <xsd:enumeration value="tkip"/>
    <xsd:enumeration value="ccmp"/>
    <xsd:enumeration value="bip"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="wlan-security-type">
  <xsd:simpleContent>
    <xsd:extension base="wlan-security-enum">
      <xsd:attribute name="other-type-definition"
        type="xsd:string" use="optional"/>
```

```
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:element name="wlan-information">
<xsd:complexType>
   <xsd:sequence>
     <xsd:element name="ssid" type="xsd:string"
       minOccurs="0" maxOccurs="1"/>
     <xsd:element name="ssid-unicast-security" minOccurs="1"
         maxOccurs="unbounded" type="wlan-security-type">
     </xsd:element>
     <xsd:element name="ssid-group-security" minOccurs="1"
         maxOccurs="1" type="wlan-security-type"/>
     <xsd:element name="ssid-management-security"
         minOccurs="1" maxOccurs="unbounded"
         type="wlan-security-type"/>
   </xsd:sequence>
   <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
 </xsd:complexType>
</xsd:element>
```

3.1.19 unexpected-behavior

Recommended for: access-request, identity, ip-address or mac-address

Unexpected-behavior metadata indicates that an endpoint is behaving in an unauthorized or unexpected manner (e.g. an endpoint previously profiled as a printer that starts sending non-print-related traffic). Where event metadata indicates a transient event, and is published using the notify operation, unexpected-behavior metadata indicates that a behavior monitoring system has determined that the endpoint is in a state of behaving unexpectedly, and should be published with an update operation (and deleted once the endpoint returns to normal behavior).

The discovered-time element refers to the time at which the unexpected behavior was first identified. The time of publication of the metadata to the MAP Server provides some temporal data, but does not accurately reflect the start time. For example, by the time unexpected behavior is identified and reported by the behavior monitoring system to the MAP Server, several seconds, minutes, or even hours may have passed since the first action on the part of the endpoint that resulted in the unexpected behavior being identified.

The magnitude element indicates how severe the effects of the activity are. Magnitude ranges from 0 to 100, with higher magnitudes indicating more severe effects.

The confidence element indicates how confident the MAP Client that published the metadata is that it accurately describes the activity of interest. Confidence ranges from 0 to 100, with higher values indicating higher confidence.

The significance element indicates how important the unexpected behavior is.

The value of the type element is a machine consumable string indicating the nature of the unexpected behavior. This is an organization-specific string that allows a PDP or Flow Controller to take action based on the type of behavior exhibited. A TCG-defined type may be specified in a future version of this specification.

The value of the information element is a human consumable informational string that may contain details on the type of behavior detected, the policy violation involved, etc. Any machine consumable information should be put into a vendor-specific metadata schema rather than in the type element.

Additional details about the unexpected behavior may be added by vendor and implementation specific metadata. To accomplish this, a MAP Client publishes auxiliary metadata from a vendor-specific schema, and correlates the event metadata with the vendor-specific metadata using the name element and ifmap-publisher-id attribute of the event metadata.

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). A MAP Client may publish enforcement-report metadata on a link adjacent to an ip-address identifier or mac-address identifier generated by a different MAP Client.

```
<xsd:element name="unexpected-behavior">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="discovered-time" type="xsd:dateTime"
       minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discoverer-id" type="xsd:string"
        minOccurs="1" maxOccurs="1"/>
      <xsd:element name="information" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
      <xsd:element name="magnitude" minOccurs="1" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="confidence" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer">
            <xsd:minInclusive value="0"/>
            <xsd:maxInclusive value="100"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="significance" minOccurs="1"
        maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="critical"/>
            <xsd:enumeration value="important"/>
            <xsd:enumeration value="informational"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="type" type="xsd:string"
        minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attributeGroup
      ref="ifmap:multiValueMetadataAttributes"/>
  </xsd:complexType>
</xsd:element>
```

4 IF-MAP Operations

PDP, Sensors, and Flow Controllers acting as MAP Clients MUST comply with this section.

4.1 MAP Client Operations

4.1.1 PDP Operations

Authenticating an Endpoint. 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:

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

In addition, if the PDP is aware of any of the following metadata, it MUST publish:

- authenticated-as metadata on the link between the access-request identifier and any identity identifiers associated with the user's authenticated identity
- capability metadata on the access-request identifier
- device-attribute metadata on the link between the access-request identifier and the endpoint's device identifier
- device-characteristic metadata on the link between the access-request identifier and its own device identifier
- role metadata on the link between the access-request identifier and an identity identifier
- 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)
- wlan-information metadata on the link between the access-request identifier and the device identifier of the PEP

The PDP MUST be configurable to create all of this metadata with a lifetime attribute of "session".

Figure 2 illustrates this.

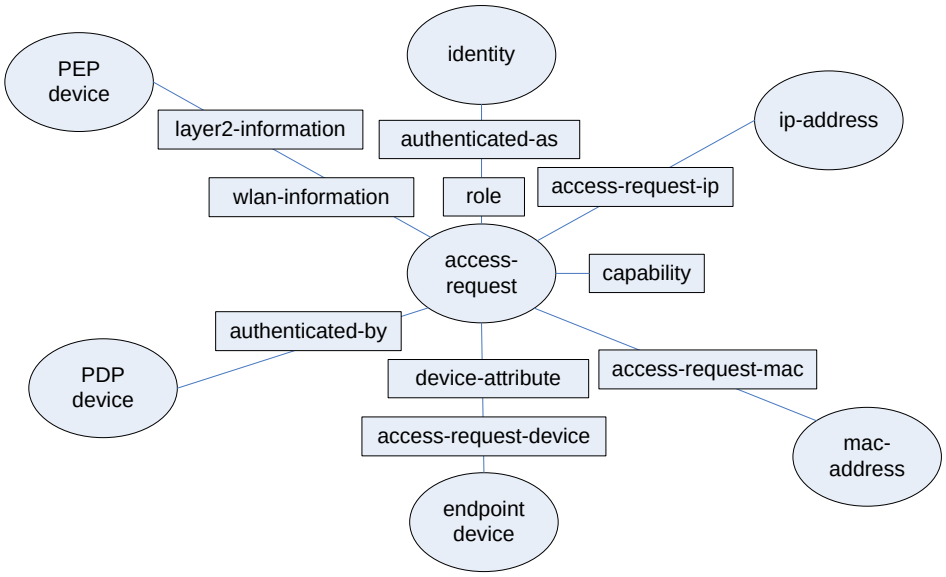


Figure 2

Endpoint Disconnection. 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".

Search/subscription. A MAP Client in a PDP SHOULD subscribe to event metadata and receive notifications, and the PDP SHOULD apply policy based on poll results.

A PDP which implements subscription capabilities MUST also take remediation action against endpoints based on policy definitions. To take remediation action against an endpoint, the PDP MUST implement subscription capabilities. Remediation actions are outside the scope of this specification, but may include disconnecting an endpoint from the network, changing the VLAN the endpoint is attached to, or changing role, capability, or device-attribute metadata associated with the endpoint.

4.1.2 Sensor Operations

Publishing a Detected Event. A Sensor acting as a MAP Client MUST publish event metadata, or device characteristic metadata, or both. This MUST be attached to either the IP address or MAC address of the endpoint for which the event or device characteristic was detected.

The Sensor MUST use notify to publish event metadata.

Figure 3 illustrates this.

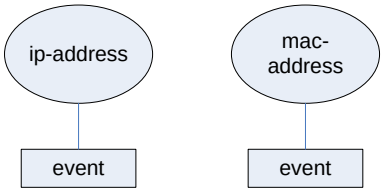


Figure 3

Publishing device-characteristic metadata. A Sensor that publishes device-characteristic metadata MUST generate a unique device identifier for itself. device-characteristic metadata MUST be published on a link between the device identifier for the Sensor and either an ip-address or mac-address identifier.

A Sensor that publishes *device-characteristic* metadata may publish with a lifetime attribute of “session” or “forever”. If the Sensor is publishing a large amount of device-characteristic metadata and loses connectivity for a few minutes, replacing all that metadata could be very expensive. In such cases, Sensors SHOULD publish with a lifetime attribute of “forever”. In either case, the Sensor SHOULD continue to monitor the network and update published metadata as changes in device characteristics are observed.

Publishing other types of metadata. When publishing 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”. When published metadata no longer describes the state of the network, the Sensor MUST delete the metadata.

4.1.3 Flow Controller Operations

Subscription. A MAP Client in a Flow Controller MUST implement the ability to subscribe to notifications, and the Flow Controller MUST apply policy based on poll results.

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. A Flow Controller’s subscriptions start at either an ip-address identifier or a mac-address identifier, depending on the network layer at which the Flow Controller operates.

A Flow Controller MUST provide access to endpoints based on policy definitions relative to metadata received in poll results. Access provisioning is outside the scope of this specification, but may include permitting or denying traffic, or changing the allowed rate at which traffic may be processed by the network.

A Flow Controller MAY peform an initial search operation for initial provisioning of access to an endpoint as an alternative to waiting for the first pollResult corresponding to a new subscription to the ip-address or mac-address of the endpoint.

When a Flow Controller blocks or quarantines an endpoint, the Flow Controller MAY publish enforcement-report metadata. 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.

5 Security Considerations

Most of the Security Considerations in the IF-MAP specifications also apply to IF-MAP Metadata for Network Security. The requirements in IF-MAP intended to address those considerations also apply to IF-MAP Metadata for Network Security. However, some issues are specific to IF-MAP Metadata for Network Security. Those issues are described here.

Clients trust the data in the MAP to be true and current. They grant and deny access based on it. False or stale data can lead to improper grants and improper denial of service.

Metadata can become stale if the publisher is shut down or network problems prevent it from connecting to the MAP server. Use of the *lifetime* attribute can mitigate this problem of stale data. A client SHOULD apply lifetime="session" when publishing if all of these conditions are true:

(a) access is intended to be granted on the basis of the metadata,

(b) when the publishing client no longer has a MAP session, the metadata may come to be false,

(c) this false information would create a security threat or other problems, and

(d) these problems are deemed more important than the consequences of deleting the metadata.

Specifically, suppose an endpoint's network session ends and the IP address is assigned to another computer, but a stale access-request-ip link is still in the MAP. Or suppose a user logs out and another user logs into the same computer, but a stale access-request-identity link is still in the MAP. Either way, someone can get piggyback access to protected resources.

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.

On the other hand, for a business-critical endpoint with a static IP address and that has a perpetual access-request based on machine credentials, the metadata will not come to be false. There is no security threat. Deleting the metadata can deny the endpoint access and cause harm. In a case like that, the links should probably be given lifetime="forever". So a client MAY offer an option to give such links a lifetime of "forever" in particular cases or in all cases.

6 Privacy Considerations

Network security metadata carries information about the endpoint or end user that can be misused by inappropriate access to the metadata and/or or unauthorized application of that knowledge. Examples include, but are not limited to:

- In some organizations it may be possible to identify a particular end user based on role and capability metadata published by MAP Clients. In this case, a MAP Client could determine how a particular end user is accessing the network by observing role and capability metadata associated with links and identifiers.
- location metadata could allow a MAP Client to determine exactly where an endpoint or end user is at a given point in time.
- device-characteristic metadata could allow an organization which makes, or is standardized upon, one type of equipment to identify that an end user is using another type of equipment. While desirable for the organization, this may be undesirable for the individual user.

To address these concerns, MAP server administrators may wish to have explicit security policies specifying who should have access to this data.

7 Recommendations for Backward Compatibility

The IF-MAP Metadata for Network Security 1.0 specification depends on the IF-MAP Binding for SOAP 2.0[3]. A MAP Client SHOULD NOT simultaneously publish or consume metadata from multiple versions of the IF-MAP protocol specification. If a network uses both the older IF-MAP Binding for SOAP 1.1[4] schema and the newer Metadata for Network Security 1.0 schema, MAP Clients SHOULD rely on the MAP Server to translate between the two schemas.

In addition to the Recommendations for Backward Compatibility contained in TNC IF-MAP Binding for SOAP 2.0, a MAP Server that supports both the IF-MAP 1.1 protocol and the IF-MAP 2.0 protocol SHOULD make the following adjustments to metadata returned in search and poll results. In the descriptions below, "old namespace" refers to the IF-MAP 1.1 metadata namespace, and "new namespace" refers to the Metadata for Network Security 1.0 namespace.

- For IF-MAP 1.1 clients, the authenticated-by (new namespace) link between an access-request and a device and the device-ip (new namespace) links between a device and ip-addresses SHOULD be replaced with an authenticated-by link (old namespace) between the access-request and one of the ip-addresses. If no ip-addresses are available, then no authenticated-by link appears in the old namespace. That is, as the authenticated-by link from the new namespace is copied to the old namespace, the device identifier and device-ip links SHOULD NOT be included in the search results and instead the access-request SHOULD be linked to an ip-address as specified in IF-MAP 1.1. In the face of multiple device-ip links from the same device identifier, the MAP Server SHOULD use the same ip-address identifier each time the metadata translation is done as long as the chosen ip-address identifier remains linked to the device identifier.
- 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. This may be accomplished by using as a prefix a string that will never be used as an ifmap-publisher-id. The ifmap-publisher-id of the device SHOULD be the same as the ifmap-publisher-id of the authenticated-by link.
- For IF-MAP 1.1 clients, the layer2-information (new namespace) link between an access-request and a device and the device-ip (new namespace) links between a device and ip-addresses SHOULD be replaced with a layer2-information (old namespace) link betwen the access-request and one of the ip-addresses. If no ip-addresses are available, then no authenticated-by link appears in the old namespace. The strategy for using a consistent ip-address identifier for authenticated-by links should also be used for layer2-information links.
- 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. The ifmap-publisher-id of the device SHOULD be the same as the ifmap-publisher-id of the layer2-information link.
- When an IF-MAP 2.0 client uses notify to publish an event, if an IF-MAP 1.1 client has a subscription resulting in the IF-MAP 1.1 client receiving information about that event, the MAP Server should send the event to the IF-MAP 1.1 client using update and delete it after 30 seconds.
- Metadata from the old namespace containing a vendor-defined type using a DNS name as the Vendor-ID SHOULD be passed as a string, despite the change to SMI as Vendor-ID in the new namespace.

- The ip-mac link between an ip-address and a mac-address is a multi-valued metadata in the Metadata for Network Security 1.0 namespace whereas it is single-valued in the IF-MAP 1.1 metadata namespace. For backward compatibility, only the last updated ip-mac link SHOULD be returned to IF-MAP 1.1 clients.

8 References

[1] Trusted Computing Group, *TNC Architecture for Interoperability*, Revision 1.4, May 2009

[2] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119, Best Practices, March 1997, IETF

[3] Trusted Computing Group, *TNC IF-MAP Binding for SOAP*, Revision 2.0, August 2010

[4] Trusted Computing Group, *TNC IF-MAP Binding for SOAP*, Revision 1.1, May 2009

[5] K. McCloghrie, D. Perkins, J. Schoenwaelder, *Structure of Management Information Version 2 (SMIv2)*, April 1999, IETF

9 Examples

This section describes in detail example flows of information between Policy Decision Points (PDPs), Policy Enforcement Points (PEPs), Flow Controllers, Sensors, and the MAP Server.

9.1 Network Diagram

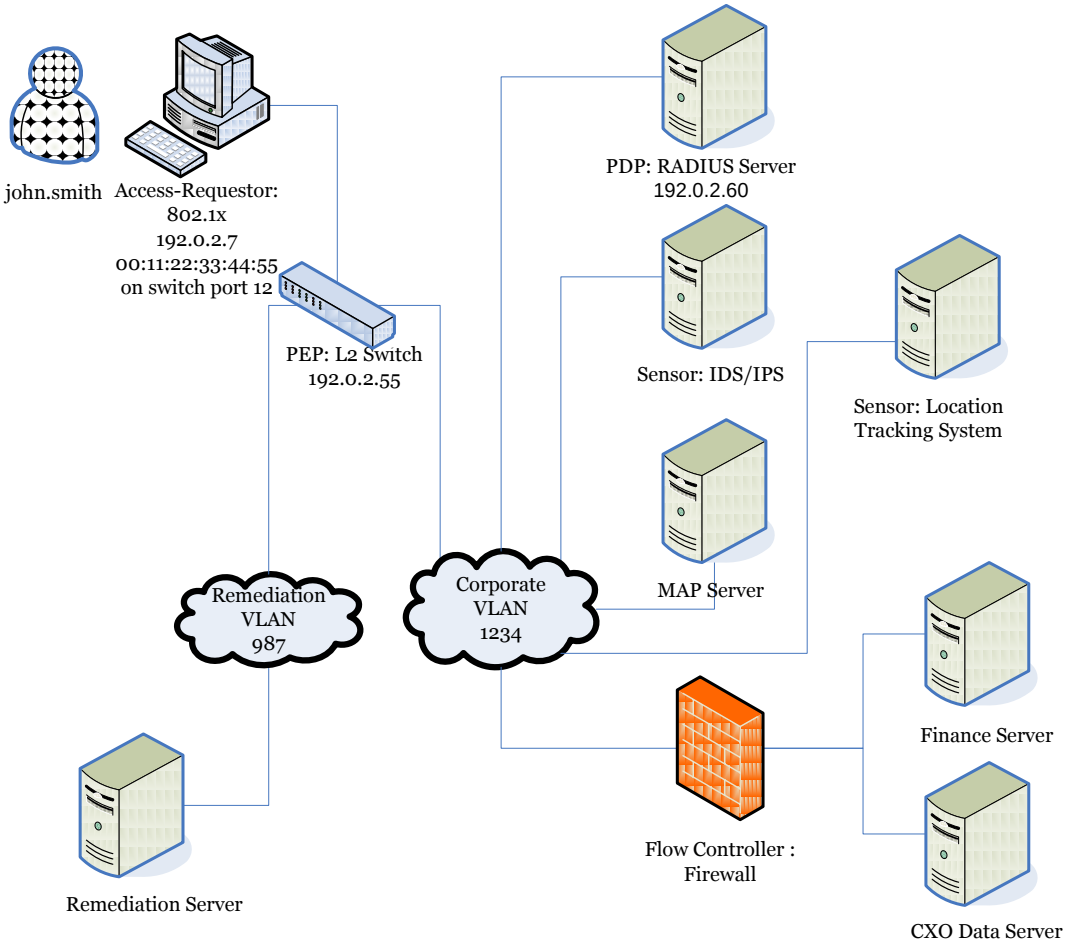


Figure 4

9.2 Example 1

Description: An 802.1X access requestor (AR) gains access through an L2 switch PEP to the corporate VLAN.

1. The PDP acting as a MAP Client connects to the MAP Server for the first time and asks the MAP Server to create a new session for the client:

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
```

```
  xmlns:meta=" http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:newSession/>
  </env:Body>
</env:Envelope>
```

2. The MAP Server response informs the PDP of the PDP's ifmap-publisher-id and session-id:

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"

xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2">
 <env:Body>
   <ifmap:response>
      <newSessionResult ifmap-publisher-id="111"
       session-id="222"/>
   </ifmap:response>
 </env:Body>
</env:Envelope>
```

3. User John Smith initiates an 802.1X connection to an L2 switch PEP. The PEP is configured to communicate with the PDP using IF-PEP (RADIUS). The PEP communicates the identifying information for this specific access request to the PDP as RADIUS attributes:
    - NAS-IP-Address: The L2 Switch's IP address, in this example is 192.0.2.55
    - NAS-Port: The physical port number to which the AR is attached, in this example port 12
    - Calling-Station-Id: The MAC address of the access-request as seen by the switch, in this example 00:11:22:33:44:55
4. The PDP uses EAP to authenticate the user and perform a TNC Handshake on the AR. In this example the IP address of the PDP is 192.0.2.60
5. Based on the authentication identity, credentials, and endpoint integrity data, the PDP applies local policy to define the roles, capabilities, VLAN, and device-attributes.
6. The PDP generates a unique access-request ID and assigns the "access-finance-service-allowed" capability. The access-request ID is of the form "ifmap-publisher-id:UID" - in this case, 111:33. The prefixing of the ifmap-publisher-id guarantees that a PDP's access-request identifiers will not collide with access-request identifiers generated by other MAP Clients. The UID can be a simple ordinal value, even monotonically increasing starting at one. The PDP's mechanism for assigning this UID should be safe and consistent across crashes and reboots.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2" >
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
         <access-request name="111:33"/>
         <metadata>
           <meta:capability ifmap-cardinality="multiValue">
```

```
                <name>access-finance-service-allowed</name>
          </meta:capability>
         </metadata>
      </update>
   </ifmap:publish>
  </env:Body>
</env:Envelope>
```

7. The MAP Server responds with a success result. For brevity this is the only IF-MAP response to a publish request described in this example chapter.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
<env:Body>
  <ifmap:response>
    <publishReceived/>
 </ifmap:response>
</env:Body>
</env:Envelope>
```

8. The PDP links the identity identifier and the access-request identifier using the roles “finance” and “employee” to the user “john.smith”

```
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <identity name="john.smith" type="username"/>
        <metadata>
          <meta:role ifmap-cardinality="multiValue">
            <name>finance</name>
          </meta:role>
          <meta:role ifmap-cardinality="multiValue">
            <name>employee</name>
          </meta:role>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

9. The PDP links the identity identifier and the access-request identifier using the authenticated-as link type.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
```

```
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <identity name="john.smith" type="username"/>
        <metadata>
          <meta:authenticated-as ifmap-
cardinality="singleValue"/>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata updates in the previous steps result in a metadata graph that looks like:

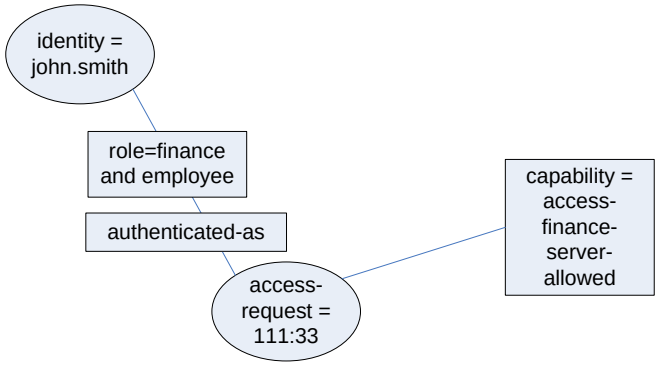


Figure 5

10. The PDP assigns the access requestor to VLAN 1234 (the corporate VLAN). The PDP combines the information it received in the RADIUS request from the PEP with the VLAN assignment to create the link to the layer 2 PEP.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:4</name>
        </device>
        <metadata>
```

```
          <meta:layer2-information ifmap-
cardinality="multiValue">
            <vlan>1234</vlan>
            <port>12</port>
          </meta:layer2-information>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata graph now looks like:

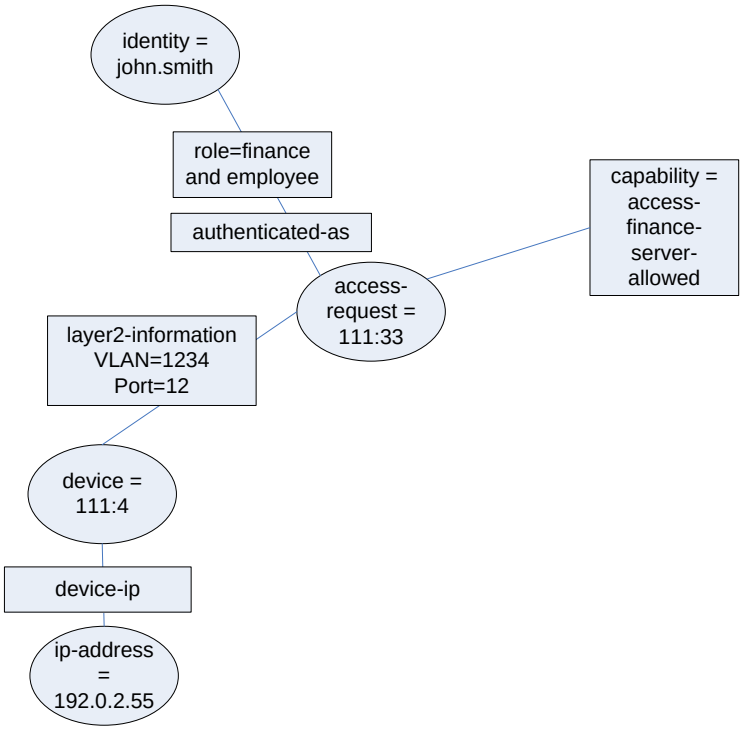


Figure 6

11. The PDP also received information from the layer 2 switch about the MAC address of the access requestor. The PDP uses this information to create a link between the access-request and the mac-address identifiers.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <mac-address value="00:11:22:33:44:55"/>
```

```
        <metadata>
          <meta:access-request-mac ifmap-
cardinality="singleValue"/>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata graph now looks like:

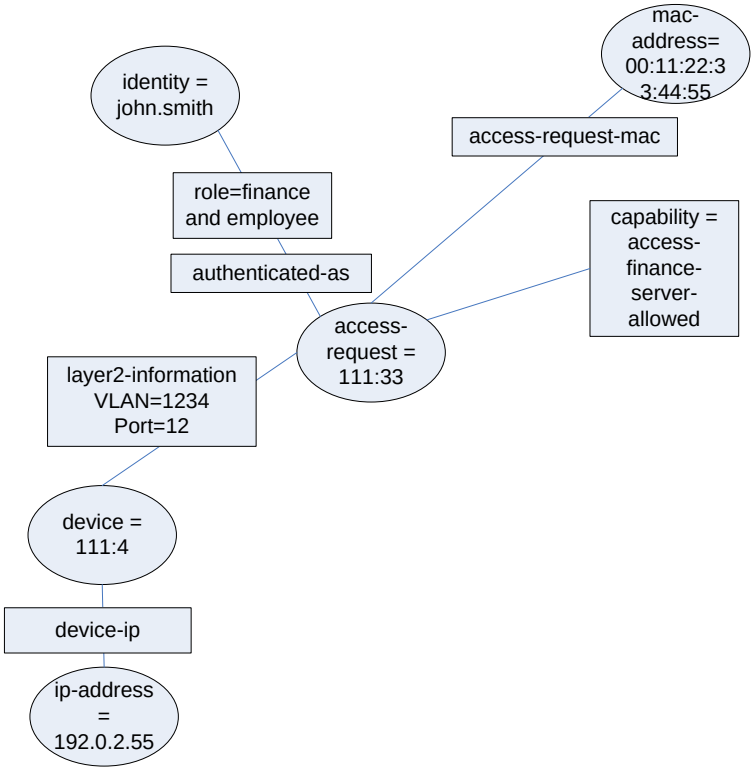


Figure 7

12. The PDP links information about itself to the access-request.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:0</name>
        </device>
        <metadata>
          <meta:authenticated-by
```

```
            ifmap-cardinality="singleValue"/>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata graph now looks like:

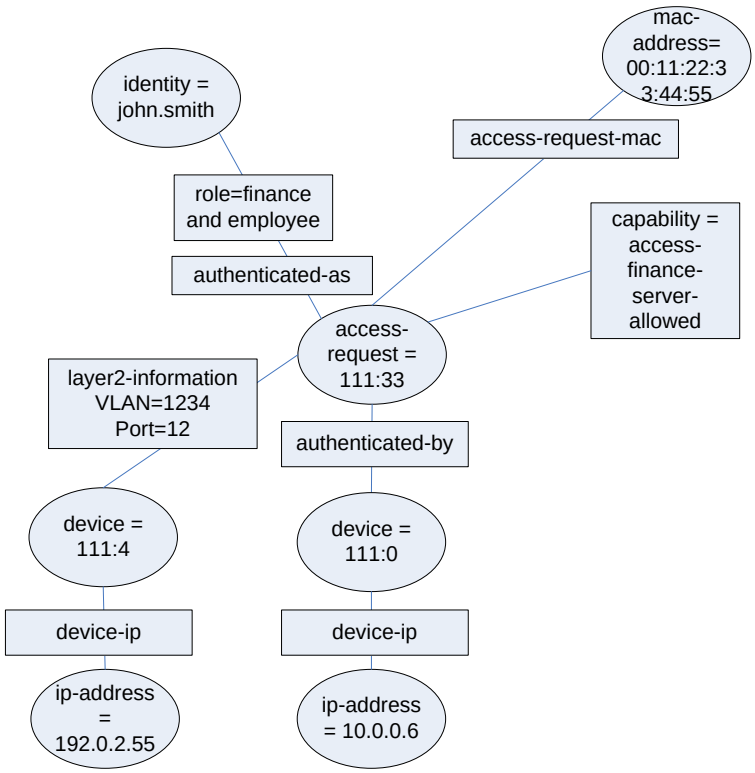


Figure 8

Note that the device-ip link between device 111:0 and ip-address 10.0.0.6 was not published in any of the requests in this example. The assumption is that the PDP published this link previously.

13. The PDP determines the device identifier name. Normally a PDP cannot determine that two access-requests are from the same device so in general the PDP creates a logical device identifier name of the form “ifmap-publisher-id:UID” (similar in form to an access-request name). However, when the PDP is provided the Attestation Identity Key (AIK) name as part of a TNC handshake the PDP should create a device identifier using the AIK. In this way two access-requests from the same device will automatically be linked in the MAP Server with no additional operations necessary. In addition to defining the device name, the PDP also attaches the access-request-device metadata. The PDP may also attach optional device-attributes, and in this example the device-attribute ‘anti-virus-running’ is attached.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
```

```
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:55</name>
        </device>
        <metadata>
          <meta:access-request-device ifmap-
cardinality=”singleValue”/>
          <meta:device-attribute ifmap-cardinality="multiValue">
            <name>anti-virus-running</name>
          </meta:device-attribute>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata graph now looks like:

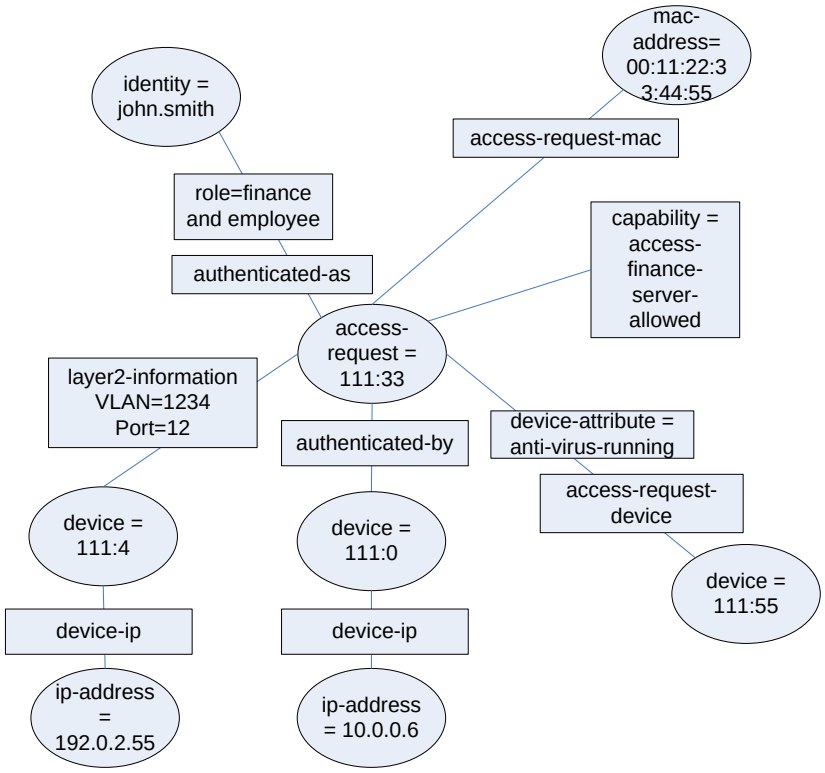


Figure 9

14. After the initial setup of the Synchronous Send-Receive Channel (SSRC), the PDP sets up an Asynchronous Receive Channel (ARC) for receiving poll results (see Section 4.1.1 of [3]. A poll request is issued on the ARC. The poll result doesn’t return until there is a change to a link or identifier in the IF MAP database that matches a subscription for the session.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:poll session-id="222"/>
  </env:Body>
</env:Envelope>
```

15. The PDP wishes to learn about any events that a Sensor might attach to the IP address identifier. The PDP issues a subscription request of the following form to be notified of those events.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="222">
      <update name="1"
        match-links="meta:access-request-mac or meta:ip-mac"
        max-depth="2" result-filter="meta:event">
        <access-request name="111:33"/>
      </update>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

16. The MAP Server acknowledges it received the subscription request. Additional subscription requests will generate responses from the MAP Server, but for brevity those are left out in this example.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <subscribeReceived/>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

9.3 Example 2

Description: The IP address of an L2 access requestor is discovered.

1. The next step in the process is for the association between the access-request and the IP address of the access requestor to be discovered. There are three possibilities
    a. The AR leases an IP address from a DHCP server
    b. An agent on the client (e.g. an IMC) indicates to the PDP that an IP address has been assigned
    c. A device in the network observes IP packets originating from the access requestor

    In case (a) the following publish request comes from an DHCP server. In case (b) the publish request comes from the PDP, and in case (c) the publish request comes from an Sensor. In all cases the publish request is the same. In this example the access requestor IP address is 192.0.2.7.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <mac-address value="00:11:22:33:44:55"/>
        <ip-address value="192.0.2.7" type="IPv4"/>
        <metadata>
          <meta:ip-mac ifmap-cardinality="singleValue"/>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

At this point the metadata graph looks like this:

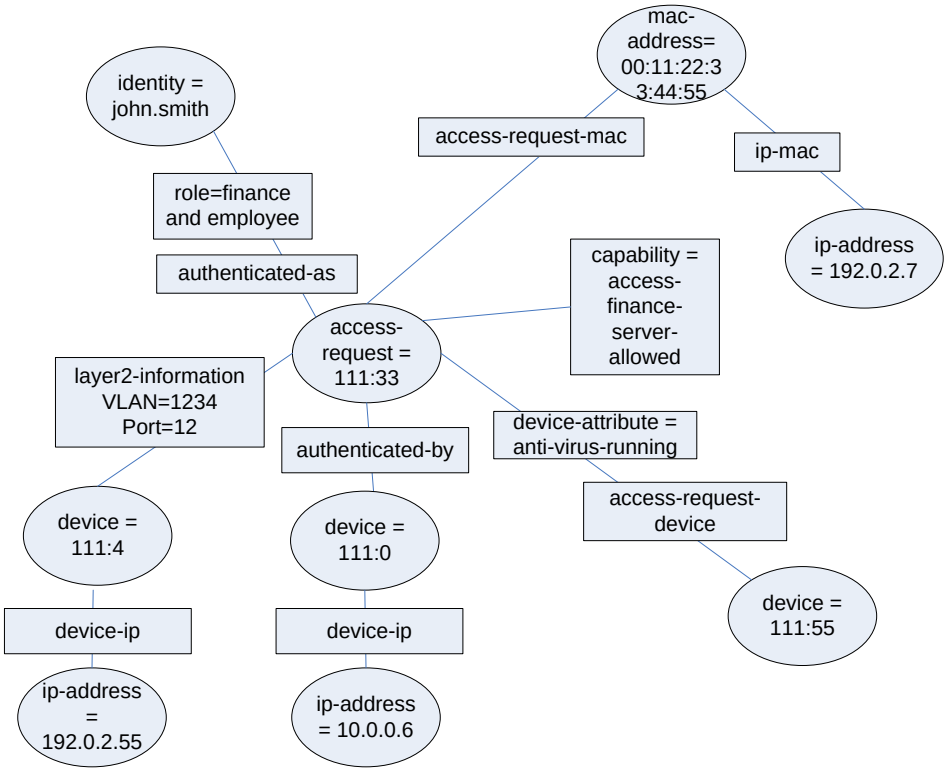


Figure 10

9.4 Example 3

Description: A Flow Controller (firewall) sees a network session from an endpoint that it has not seen before and needs to determine whether to allow or deny access.

1. The Flow Controller establishes SSRC and ARC connections to the MAP Server.
2. A network connection is attempted through a Flow Controller (such as a firewall). The Flow Controller uses the source IP address in the connection initiation packet (in this case 192.0.2.7) to subscribe to an ip-address identifier. The Flow Controller fetches the capabilities, events, device-attributes, and roles using a subscription.

   Event metadata is typically attached to ip-address identifiers, and capability metadata is attached to access-request identifiers. Getting from the ip-address to the access-request can take two possible paths through the metadata graph: a two step path through ip-mac and access-request-mac for layer 2 (802.1x) based authentications, or a single step path through access-request-ip for layer 3 (VPN) based authentications.

   Role metadata is attached to links between access-request identifiers and identity identifiers. In order to receive notification about changes to roles for the identity associated with an ip-address, the search depth is 3: ip-address → mac-address → access-request → identifier.

   Since device-attribute metadata appears on links from access-request identifiers to device identifiers, in order to receive notification about changes to device-attributes, the search depth is 3: ip-address → mac-address → access-request → device.

A search of depth 3 could potentially pull in metadata from other access-requests. If a single identity has more than one layer 3 (VPN) authentications, a search of depth 4 could follow these links: ip-address → access-request → identity → other access-requests. Any role metadata from links between the identity and other access-requests would be returned by the subscription. To prevent this from happening, identity is specified as a terminal identity type in the search. Device is also included to prevent device-attributes from other access-requests from being included in the subscription.

For a more detailed description of the metadata associated with a multi-homed device see section 9.12.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="222">
      <update name="1"
        match-links="meta:ip-mac or meta:access-request-mac or
meta:access-request-ip or meta:authenticated-as or access-
request-device"
        max-depth="3"
   terminal-identifier-type="identity,device"
        result-filter="meta:capabilities or meta:event or
meta:role or meta:device-attribute">
        <ip-address value="192.0.2.7" type="IPv4"/>
      </update>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

3. Since the MAP Server already contains metadata for the IP address 192.0.2.7, the subscription generates an immediate search result containing the metadata requested in the subscription. The reason the Flow Controller issues a subscription request instead of a search in this example is that the subscription will continue to deliver updated responses as the data in the MAP database changes without any further requests from the MAP Client.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <searchResult name="1">
          <resultItem>
            <ip-address value="192.0.2.7" type="IPv4"/>
          </resultItem>
          <resultItem>
            <mac-address value="00:11:22:33:44:55"/>
```

```
          </resultItem>
          <resultItem>
            <ip-address value="192.0.2.7" type="IPv4"/>
            <mac-address value="00:11:22:33:44:55"/>
          </resultItem>
          <resultItem>
            <mac-address value="00:11:22:33:44:55"/>
            <access-request name="111:33"/>
          </resultItem>
          <resultItem>
            <access-request name="111:33"/>
            <metadata>
              <meta:capability ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>access-finance-service-allowed</name>
              </meta:capability>
            </metadata>
          </resultItem>
          <resultItem>
            <access-request name="111:33"/>
            <identity name="john.smith" type="username"/>
            <metadata>
              <meta:role ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>finance</name>
              </meta:role>
              <meta:role ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>employee</name>
              </meta:role>
            </metadata>
          </resultItem>
          <resultItem>
            <identity name="john.smith" type="username"/>
          </resultItem>
          <resultItem>
            <access-request name="111:33"/>
            <device>
              <name>111:55</name>
            </device>
            <metadata>
              <meta:device-attribute
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>anti-virus-running</name>
              </meta:device-attribute>
            </metadata>
          </resultItem>
          <resultItem>
            <device>
              <name>111:55</name>
            </device>
          </resultItem>
```

```
        </searchResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

4. The Flow Controller uses the capabilities, roles, and device-attributes as well as local policy to determine the access privileges allowed for this endpoint. For example, john.smith might be allowed to access the finance server but not allowed access to the CXO data server.

Change of capabilities, roles or device-attribute is described in Example 4. A PDP de-authorizing a user is described in Example 7.

It is possible for the lease (ip-mac link metadata) to be deleted, or a new lease (ip-mac link) to be created; in this case the Flow Controller would get the new information for the IP address and modify access accordingly.

When all connections from this IP address have closed and some period of time has gone by, the Flow Controller may unsubscribe for this IP address. This implementation optimization allows the Flow Controller to receive fewer poll results from the MAP Server.

9.5 Example 4

Description: A TNC handshake occurs that causes device-attributes and capabilities to change.

1. A PDP completes a TNC handshake with the access requestor. The output from the IMV indicates that the antivirus signatures are out of date. The PDP's local policy requires that access to the finance server is not allowed in this case.
2. The PDP removes the "access-finance-server-allowed" capability.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta:capability[name='access-finance-
service-allowed' and @ifmap-publisher-id='111']">
        <access-request name="111:33"/>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

3. The PDP also changes the device-attribute from "anti-virus-running" to "av-signatures-out-of-date". Because device-attribute is a multi-valued metadata type, a single update operation would add to the list of device-attribute metadata elements. This is not what is desired. To accomplish a modify operation on a multi-valued metadata type, a single IF-MAP publish operation combines a delete followed by an update. Because these are combined into a single IF-MAP publish operation they are treated atomically by the MAP Server.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta:device-attribute[name='anti-virus-
running' and @ifmap-publisher-id='111']">
        <access-request name="111:33"/>
        <device>
          <name>111:55</name>
        </device>
      </delete>
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:55</name>
        </device>
        <metadata>
          <meta:device-attribute ifmap-cardinality="multiValue">
            <name>av-signatures-out-of-date </name>
          </meta:device-attribute>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The metadata graph now looks like:

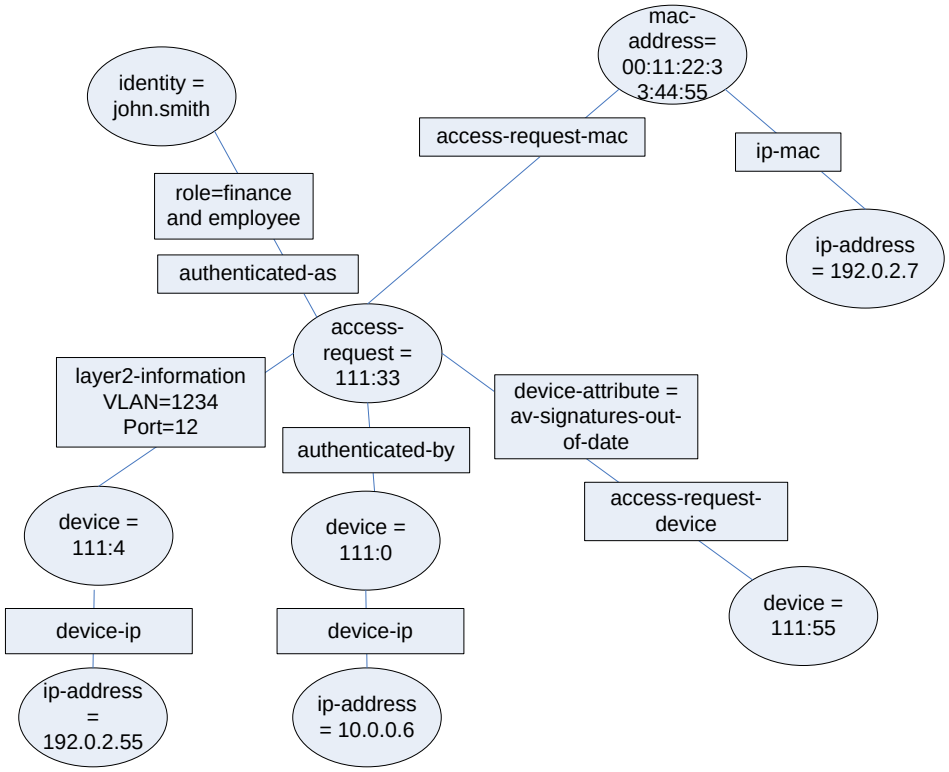


Figure 11

4. A Flow Controller subscribed to the ip-address identifier of the access requestor is notified that the capabilities and device attributes have changed and re-evaluates access as appropriate. The lack of the “access-finance-service-allowed” capability causes the Flow Controller to immediately shut down access to the finance server for this endpoint.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <deleteResult name="1">
          <resultItem>
            <access-request name="111:33"/>
            <metadata>
              <meta:capability ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>access-finance-service-allowed</name>
              </meta:capability>
            </metadata>
          </resultItem>
          <resultItem>
```

```
            <access-request name="111:33"/>
            <device>
              <name>111:55</name>
            </device>
            <metadata>
              <meta:device-attribute
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>anti-virus-running</name>
              </meta:device-attribute>
            </metadata>
          </resultItem>
        </deleteResult>
        <updateResult name="1">
          <resultItem>
            <access-request name="111:33"/>
            <device>
              <name>111:55</name>
            </device>
            <metadata>
              <meta:device-attribute
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:15:22Z">
                <name>av-signatures-out-of-date</name>
              </meta:device-attribute>
            </metadata>
          </resultItem>
        </updateResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

Since roles did not change, no searchResults are returned for subscription 2.

9.6 Example 5

Description: A TNC handshake occurs that causes the PDP to change the VLAN of an access requestor.

1. In addition to the action taken in Example 4, the PDP, based on local policy, also chooses to assign the access-request to the remediation VLAN. The PDP uses the following publish request to update the MAP database with the new VLAN. Because layer2-information is a single-valued metadata type, a single update operation updates the value.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
```

```
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device>
          <name>111:4</name>
        </device>
        <metadata>
          <meta:layer2-information ifmap-
cardinality="multiValue">
            <vlan>978</vlan>
            <port>12</port>
          </meta:layer2-information>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

2. The PDP communicates the change of VLAN via RADIUS, and the L2 switch places the access-request on the remediation VLAN.
3. The DHCP server updates the IF-MAP database with an additional ip-mac link representing the DHCP lease on the remediation VLAN. The metadata graph with all of the updates looks like:

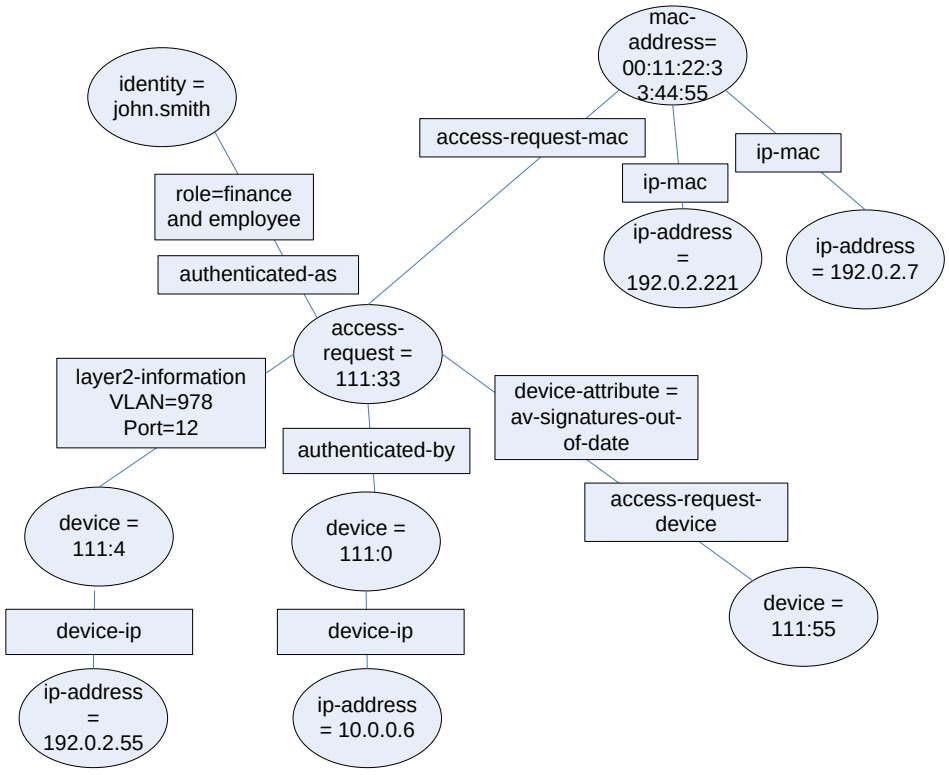


Figure 12

9.7 Example 6

Description: A Sensor detects a vulnerability on the access requestor device. Both the Flow Controller and the PDP discover this and cut off or limit access.

1. The Sensor establishes an SSRC connection to the MAP Server. The Sensor does not issue any subscription requests and therefore does not need to establish an ARC connection to the MAP Server.
2. The Sensor detects a vulnerability associated with the access requestor's IP Address and publishes an event to the MAP server using notify.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="224">
      <notify>
        <ip-address value=" 192.0.2.7" type="IPv4"/>
        <metadata>
          <meta:event ifmap-cardinality="multiValue">
            <name>Kazaa In use</name>
            <discovered-time>2010-04-20T12:21:39Z </discovered-
time>
            <magnitude>45</magnitude>
            <confidence>100</confidence>
            <significance>important</significance>
            <type>policy violation</type>
            <information>Possible MP3 file sharing
violation</information>
          </meta:event>
        </metadata>
      </notify>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

3. The Flow Controller subscribed directly on the ip-address identifier gets notification of this event and re-evaluates access privileges accordingly.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <notifyResult name="1">
          <resultItem>
            <ip-address value="192.0.2.7" type="IPv4"/>
            <metadata>
              <meta:event ifmap-cardinality="multiValue"
```

```
                ifmap-publisher-id="112"
                ifmap-timestamp="2010-04-20T12:21:39Z">
                <name>Kazaa In use</name>
                <discovered-time>2010-04-20T12:21:34Z
</discovered-time>
                <magnitude>45</magnitude>
                <confidence>100</confidence>
                <significance>important</significance>
                <type>policy violation</type>
                <information>Possible MP3 file sharing
violation</information>
              </meta:event>
            </metadata>
          </resultItem>
        </notifyResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

4. The PDP subscribed on the access-request that is linked to this ip-address identifier (directly or through an intermediate node) also gets notification. The PDP may choose to take action by re-evaluating access privileges in a similar way to what occurred during a TNC handshake in Example 5.

It would be reasonable for the Flow Controller to take no action in response to events. A Flow Controller could choose to not subscribe to events at all, instead relying on the PDP to change capabilities and device-attributes in response to events. The Flow Controller would then take action based on capabilities and device-attributes changes.

Since an event is published using notify, the event is not saved in the IF-MAP database and does not need to be cleaned up by the Sensor.

9.8 Example 7

Description: An access requestor disconnects from the network.

1. When the PDP detects that an access requestor has disconnected from the network, it first removes the subscription it requested for this access-request. There is no need for the PDP to get change notifications for metadata it will be deleting in subsequent steps.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="222">
      <delete name="1"/>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

2. The PDP deletes the capability it added.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete
        filter="meta:capability[@ifmap-publisher-id='111']">
        <access-request name="111:33"/>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

3. The PDP deletes the roles it added.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete
        filter="meta:role[@ifmap-publisher-id='111']">
        <access-request name="111:33"/>
        <identity name="john.smith" type="username"/>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

4. The PDP removes the authenticated-as link between the access-request and the identity.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta:authenticated-as[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <identity name="john.smith" type="username"/>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

5. The PDP removes the layer2-information link between the access-request and the IP address of the PEP.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta:layer2-information[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <device>
          <name>111:4</name>
        </device>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

6. The PDP removes the access-request-mac link between the access-request and the mac-address.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta: access-request-mac[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <mac-address value="00:11:22:33:44:55"/>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

7. The PDP removes the authenticated-by link between the access-request and the IP address of the PDP.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete filter="meta:authenticated-by[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
```

```
        <device>
           <name>111:0</name>
        </device>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

8. The PDP removes the access-request-device and device-attributes metadata from the link between the access-request and the device.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete
        filter="meta:device-attribute or meta:access-request-
device">
        <access-request name="111:33"/>
        <device>
          <name>111:55</name>
        </device>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

The PDP should combine all of these delete operations into a single publish request. This way the MAP Server will treat this as a single operation. Only a single poll response will go out to each client that has subscriptions to identifiers and links that have changed. This reduces the bandwith requirements between clients and servers as well as reduces the work the clients need to do in response to incremental changes. Batching requests of this type is preferred for any operation where it is logically possible.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete
        filter="meta:capability[@ifmap-publisher-id='111'111]">
        <access-request name="111:33"/>
      </delete>
      <delete filter="meta:role[@ifmap-publisher-id='111'] or
meta:authenticated-as[@ifmap-publisher-id='111']">
        <access-request name="111:33"/>
        <identity name="john.smith" type="username"/>
      </delete>
```

```
      <delete filter="meta:layer2-information[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <device>
           <name>111:4</name>
        </device>
      </delete>
      <delete filter="meta:access-request-mac[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <mac-address value="00:11:22:33:44:55"/>
      </delete>
      <delete filter="meta:authenticated-by[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <ip-address value="192.0.2.60" type="IPv4"/>
      </delete>
      <delete filter="meta:device-attribute[@ifmap-publisher-
id='111'] or meta:access-request-device[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <device>
          <name>111:55</name>
        </device>
      </delete>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

9. At this point a Flow Controller that has subscribed on the ip-address identifier will get notification of the change.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <deleteResult name="1">
          <resultItem>
            <access-request name="111:33"/>
            <identity name="john.smith" type="username"/>
            <metadata>
              <meta:role ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>finance</name>
              </meta:role>
              <meta:role ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:00:05Z">
                <name>employee</name>
              </meta:role>
```

```
            </metadata>
          </resultItem>
          <resultItem>
            <access-request name="111:33"/>
            <device>
              <name>111:55</name>
            </device>
            <metadata>
              <meta:device-attribute
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-20T12:15:22Z">
                <name>av-signatures-out-of-date</name>
              </meta:device-attribute>
            </metadata>
          </resultItem>
        </deleteResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

The Flow Controller gave access to this endpoint based on capabilities, roles, and device-attributes. On receiving the poll result containing none of those metadata elements, the Flow Controller shuts off access.

9.9 Example 8

Description: An access requestor requests access through a PDP using a layer 3 access protocol (e.g. VPN)

1. Provisioning access at layer 3 is different from provisioning access at layer 2:
    a. The PDP does not attach any type of layer2-information metadata
    b. The PDP does not link the access-request to the mac-address. Instead, the PDP links the access-request directly to the ip-address of the access requestor through an access-request-ip link

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <ip-address value="192.0.2.7" type="IPv4"/>
        <metadata>
          <meta:access-request-ip
            ifmap-cardinality="singleValue"/>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
```

```
</env:Envelope>
```

Assuming the PDP creates the authenticated-as, authenticated-by, access-request-device, and device-attribute links as in Example 1, the metadata graph looks like:

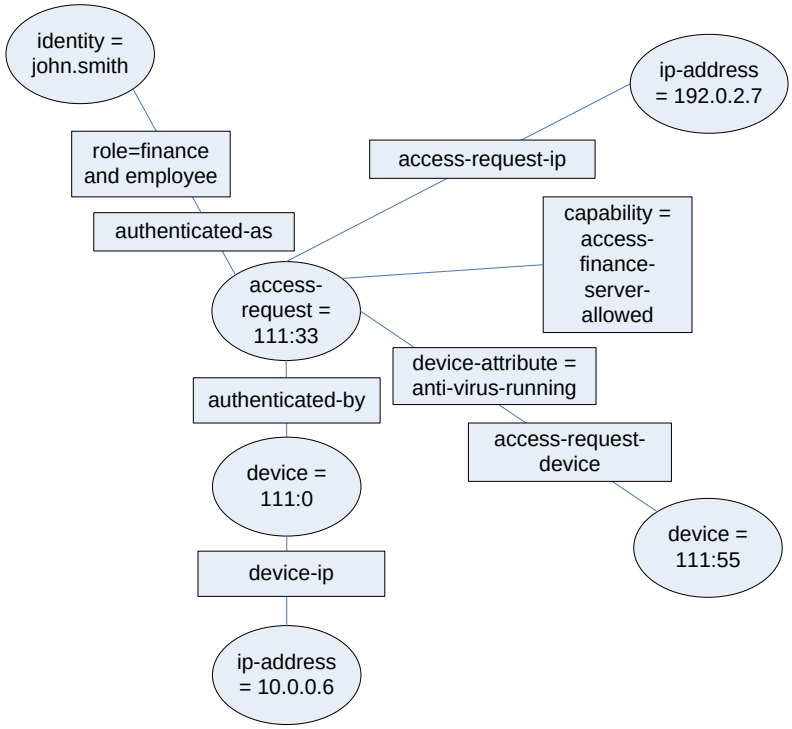


Figure 13

2. An DHCP server could apply the ip-mac link as additional information. But in the layer 3 case that link would be purely informational, and does not play a role in any of the processing of these examples.

3. The Flow Controller subscription in Example 3 already contains support for handling the layer 2 or layer 3 cases by allowing the search to follow the access-request-ip link or the ip-mac and access-request-mac links. The Flow Controller needs to be able to parse either type of poll result set.

9.10 Example 9

Description: A network element (eg. PDP or Sensor) crashes and is rebooted.

Any MAP Client SHOULD maintain a persistent local store of information about the metadata it published to the IF-MAP database. For a Sensor publishing events attached to IP addresses, the Sensor could keep a list of IP address for which it has published event metadata. When the Sensor is rebooted, it checks to see if the metadata is still valid, and cleans it up where appropriate. In a similar way a PDP keeps track of the access-request elements that it has added, and cleans up the associated links after a reboot.

9.11 Example 10

Description: A network element (e.g. PDP or Sensor) crashes in a way where it cannot be rebooted (disk crash, burst into flames etc).

After being repaired, the network element connects to the MAP Server and issues a purgePublisher request specifying its own ifmap-publisher-id.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:purgePublisher session-id="222" ifmap-publisher-
id="345"/>
  </env:Body>
</env:Envelope>
```

This removes all of the network element's data from the MAP Server.

9.12 Example 11

Description: A user with a single identity using a single multi-homed device authenticates to PDP1 using a layer 2 access protocol (e.g. 802.1X) and authenticates to PDP2 using a layer 3 access protocol (e.g. VPN).

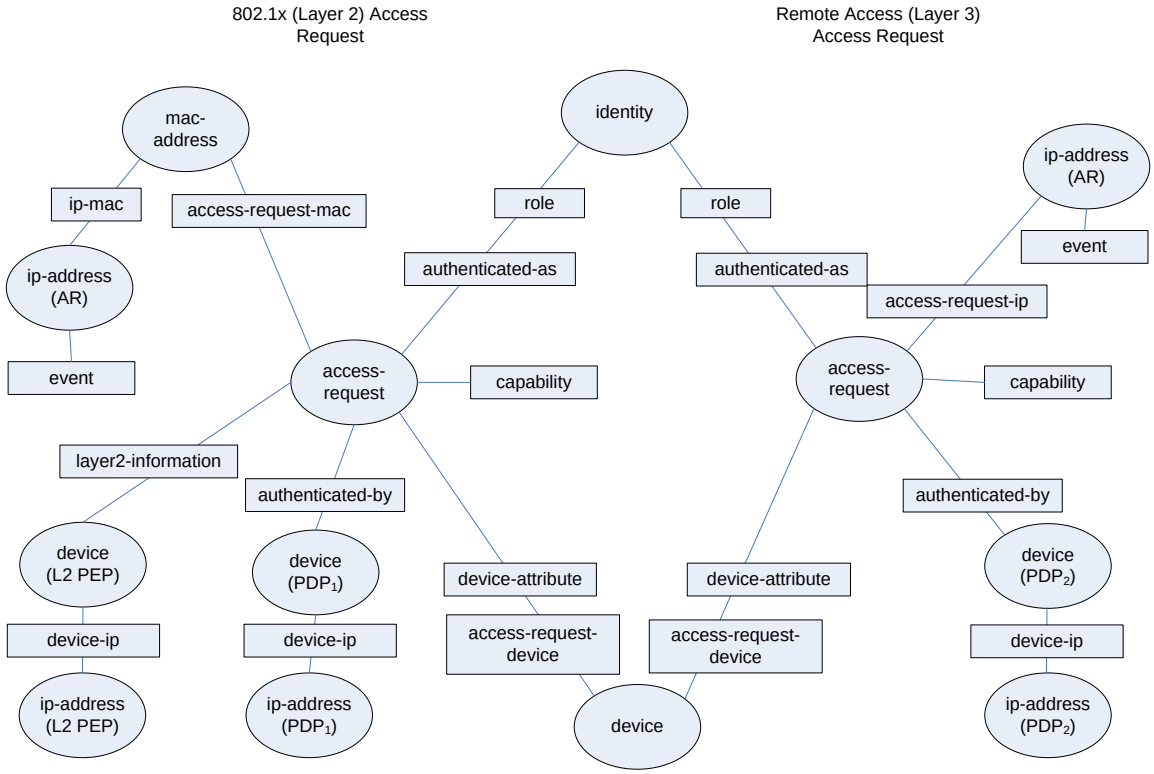


Figure 14

Some additional considerations must be handled to correctly address this case.

1. PDPs or Flow Controllers that are looking at results posted by other PDPs must consider that the same device-attribute string can be applied multiple times on links to the same device and the same role string can be applied multiple times on links to the same identity. In the above example both PDP_1 and PDP_2 can assign the “employee” role, and multiple instances of the same role name must be ignored when doing evaluation.
2. The subscriptions in the above examples are constructed with the appropriate depth parameters that will return poll results that contain all of the device-attributes and roles. A Flow Controller or PDP should use all of the device-attributes and roles on any of the links when doing access calculations. How they are combined would depend on local policy.
3. In order to get roles from all links, device-attributes from all links, but capabilities from only the access-request of interest, it is necessary to use multiple subscriptions as shown in the preceding examples.

9.13 Example 12

Description: A location tracking system (such as a WiFi RTLS) acts as a MAP Sensor, publishing wireless endpoint location information; other networking devices can leverage location information to help determine access control policy (such as VLAN assignment) based on a combination of user authentication and user location.

The PDP provides user authentication and endpoint health checking and provisions policy to the network devices acting as enforcement points. The PDP also acts as a MAP Client, searching the MAP for location metadata on endpoints requesting access and publishing metadata to the MAP when it authenticates an endpoint.

1. The location tracking system acting as a Sensor publishes location metadata about a device when the location of that device is known

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="223">
      <update>
        <mac-address value="00:11:22:33:44:55"/>
        <metadata>
          <meta:location ifmap-cardinality="multiValue">
            <location-information type="room" value="Conference
Room 277"/>
            <location-information type="building" value="HQ"/>
            <discovered-time>2010-04-20T14:18:41Z </discovered-
time>
            <discoverer-id>987</discoverer-id>
          </meta:location>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

2. The PDP acting as a MAP Client connects to the MAP Server for the first time and asks the MAP Server to create a new session for the client.

3. The MAP Server response informs the PDP of the PDP's ifmap-publisher-id and session-id
4. User John Smith initiates an 802.1X connection to an L2 switch PEP. The PEP is configured to communicate with the PDP using IF-PEP (RADIUS). The PEP communicates the identifying information for this specific access request to the PDP as RADIUS attributes:
    - NAS-IP-Address: The L2 Switch's IP address, in this example is 192.0.2.55
    - NAS-Port: The physical port number to which the AR is attached, in this example port 12
    - Calling-Station-Id: The MAC address of the access-request as seen by the switch, in this example 00:11:22:33:44:55
5. The PDP uses EAP to authenticate the user and perform a TNC Handshake on the AR.
6. Based on the authentication identity, credentials, and endpoint integrity data, the PDP applies local policy to define the roles, capabilities, VLAN, and device-attributes.
    a. The PDP wishes to learn about location metadata a location tracking system might have previously attached to the MAC address identifier by issuing a search request

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:search session-id="223"
     match-links="meta:access-request-ip or meta:ip-mac or
meta:access-request-mac"
     max-depth="3" result-filter="meta:location"
     terminal-identifier-type="identity,device">
      <mac-address value="00:11:22:33:44:55"/>
    </ifmap:search>
  </env:Body>
</env:Envelope>
```

The MAP server will return any results that were previously published to the MAP server by the Sensor

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <searchResult>
        <resultItem>
          <mac-address value="00:11:22:33:44:55"/>
          <metadata>
            <meta:location ifmap-cardinality="multiValue"
             ifmap-publisher-id="114"
             ifmap-timestamp="2010-04-20T14:18:42Z">
              <location-information type="room" value="Conference
Room 277"/>
              <location-information type="building" value="HQ"/>
```

```
              <discovered-time>2010-04-20T14:18:41Z</discovered-
time>
              <discoverer-id>987</discoverer-id>
            </meta:location>
          </metadata>
        </resultItem>
      </searchResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

The PDP may incorporate this location metadata into the decision making process when applying local policy to define the roles, capabilities, VLAN, and device-attributes based upon location of the device making the access request.

b. The PDP subscribes to changes in location-information metadata attached to the MAC address identifier

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="222">
      <update name="1"
        match-links="meta:access-request-mac"
        max-depth="2" result-filter="meta:location">
        <mac-address value="00:11:22:33:44:55"/>
      </update>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

7. The device which originally initiated the access request is detected to have changed location.
   a. The location tracking system acting as a Sensor publishes an update to the location metadata about a device when the location of that device changes

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="223">
      <delete filter="meta:location[@ifmap-publisher-id='111'">
        <mac-address value="00:11:22:33:44:55"/>
      </delete>
      <update>
        <mac-address value="00:11:22:33:44:55"/>
        <metadata>
          <meta:location ifmap-cardinality="multiValue">
            <location-information type="room" value="Lobby"/>
```

```
            <location-information type="building" value="HQ"/>
            <discovered-time>2010-04-20T14:26:15Z</discovered-
time>
            <discoverer-id>987</discoverer-id>
          </meta:location>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

b. The PDP receives the updated information via the subscription mechanism

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <deleteResult name="2">
          <resultItem>
            <mac-address value="00:11:22:33:44:55"/>
            <metadata>
              <meta:location ifmap-cardinality="multiValue"
                ifmap-publisher-id="114"
                ifmap-timestamp="2010-04-20T14:18:42Z">
                <location-information type="room"
value="Conference Room 277"/>
                <location-information type="building"
value="HQ"/>
                <discovered-time>2010-04-
20T14:18:41Z</discovered-time>
                <discoverer-id>987</discoverer-id>
                </meta:location>
            </metadata>
          </resultItem>
        </deleteResult>
        <updateResult name="2">
          <resultItem>
            <mac-address value="00:11:22:33:44:55"/>
            <metadata>
              <meta:location ifmap-cardinality="multiValue"
                ifmap-publisher-id="114"
                ifmap-timestamp="2010-04-20T14:26:13Z">
                <location-information type="room" value="Lobby"/>
                <location-information type="building"
value="HQ"/>
                <discovered-time>2010-04-20T14:26:15Z
</discovered-time>
                <discoverer-id>987</discoverer-id>
              </meta:location>
            </metadata>
          </resultItem>
```

```
        </updateResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

The PDP may incorporate this location metadata into the decision making process when re-evaluating the local policy as it applies to the access request.

9.14 Example 13

Description: A physical security access system (such as a proximity badge system) acts as a MAP Sensor, publishing user location information; other network devices can leverage that information to apply location-based security policies and provision network access only for users physically present and authorized to be in a location.

The PDP provides user authentication and endpoint health checking and provisions policy to the network devices acting as enforcement points. The PDP also acts as a MAP Client, searching the MAP for location metadata on endpoints requesting access and publishing metadata to the MAP when it authenticates an endpoint.

1. The physical security access system acting as a Sensor publishes location metadata about a user when the location of that user is known

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="223">
      <update>
        <identity name="john.smith" type="username"/>
        <metadata>
          <meta:location ifmap-cardinality="multiValue">
            <location-information type="name" value="Secured
Area"/>
            <location-information type="zone" value="top-secret-
clearance-required"/>
            <discovered-time>2009-09-21T10:32:52</discovered-
time>
            <discoverer-id>654</discoverer-id>
          </meta:location>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

2. The PDP acting as a MAP Client connects to the MAP Server for the first time and asks the MAP Server to create a new session for the client.
3. The MAP Server response informs the PDP of the PDP's ifmap-publisher-id and session-id
4. User John Smith initiates an 802.1X connection to an L2 switch PEP. The PEP is configured to communicate with the PDP using IF-PEP (RADIUS). The PEP

communicates the identifying information for this specific access request to the PDP as RADIUS attributes:

- NAS-IP-Address: The L2 Switch's IP address, in this example is 192.0.2.55
- NAS-Port: The physical port number to which the AR is attached, in this example port 12
- Calling-Station-Id: The MAC address of the access-request as seen by the switch, in this example 00:11:22:33:44:55

5. The PDP uses EAP to authenticate the user and perform a TNC Handshake on the AR.
6. Based on the authentication identity, credentials, and endpoint integrity data, the PDP applies local policy to define the roles, capabilities, VLAN, and device-attributes.
    a. The PDP wishes to learn about location metadata a physical security access system might have previously attached to the user identifier by issuing a search request

       The MAP server will return any results that were previously published to the MAP server by the Sensor

       The PDP may incorporate this location metadata into the decision making process when applying local policy to define the roles, capabilities, VLAN, and device-attributes based upon location of the user making the access request.

    b. The PDP subscribes to changes in location metadata attached to the MAC address identifier
7. The device which originally initiated the access request is detected to have changed location.
    c. The physical security access system acting as a Sensor publishes an update to the location metadata about a device when the location of that device changes

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="223">
      <delete filter="meta:location[@ifmap-publisher-id='111'">
        <identity name="john.smith" type="username"/>
      </delete>
      <update>
        <identity name="john.smith" type="username"/>
        <metadata>
          <meta:location ifmap-cardinality="multiValue">
            <location-information type="name" value="Guest
Area"/>
            <location-information type="zone" value="insecure-
public-access-zone"/>
            <discovered-time>2009-09-21T11:15:20</discovered-
time>
            <discoverer-id>654</discoverer-id>
          </meta:location>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

d. The PDP receives the updated information via the subscription mechanism

   The PDP may incorporate this location metadata into the decision making process when re-evaluating the local policy as it applies to the access request.

9.15 Example 14

Description: A PDP requests an investigation of the MAC address of an AR that is attempting to access the network. A Sensor responds with device characteristics. The PDP grants network access according to its policies as applied to the device characteristics.

1. The Sensor subscribes to the PDP's IP address identifier so that it can be notified when the PDP publishes request-for-investigation metadata.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="789">
      <update name="192.0.2.60"
          match-links="meta:device-ip or meta:request-for-
investigation"
          max-depth="2"
          result-filter="meta:request-for-investigation">
        <ip-address value="192.0.2.60"/>
      </update>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

2. The PDP receives a MAC auth RADIUS request from a switch. The PDP does not recognize the MAC address. The PDP responds to the switch, putting the endpoint onto a quarantine network. The PDP subscribes to the mac-address identifier of the endpoint so that it can receive pollResults when device-characteristic metadata is published.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:subscribe session-id="222">
      <update name="mac:11:22:33:44:55:66"
          match-links="meta:device-characteristic"
          max-depth="2"
          result-filter="meta:device-characteristic">
        <mac-address value="11:22:33:44:55:66"/>
      </update>
    </ifmap:subscribe>
  </env:Body>
</env:Envelope>
```

3. The PDP publishes a request-for-investigation link between its device identifier and the MAC address of the endpoint. The request-for-investigation link includes a qualifier attribute of “b3-switch” so that only Sensors that have visibility onto the networks served by the building 3 switch will respond.

```
<?xml version="1.0"?>
<env:Envelope
 xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <notify>
        <device><name>123:PDP</name></device>
        <mac-address value="11:22:33:44:55:66"/>
        <metadata>
          <meta:request-for-investigation qualifier="b3-switch"
ifmap-cardinality="multiValue"/>
        </metadata>
      </notify>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

4. The MAP Server notifies the Sensor of the request-for-investigation

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <notifyResult>
          <resultItem>
            <ip-address value="192.0.2.60"/>
            <device><name>123:PDP</name></device>
          </resultItem>
          <resultItem>
            <device><name>123:PDP</name></device>
            <mac-address value="11:22:33:44:55:66"/>
            <metadata>
              <meta:request-for-investigation
                qualifier="b3-switch"
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="111"
                ifmap-timestamp="2010-04-21T16:11:09Z"/>
            </metadata>
          </resultItem>
        </notifyResult>
      </pollResult>
    </ifmap:response>
```

```
  </env:Body>
</env:Envelope>
```

5. The Sensor scans the endpoint with MAC address 11:22:33:44:55:66 and determines that the device is a network printer. The Sensor publishes device-characteristic metadata using vendor-defined types with the TCG SMI of 21911:

```
<?xml version="1.0"?>
<env:Envelope
 xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
 xmlns:env="http://www.w3.org/2003/05/soap-envelope"
 xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="789">
      <update>
        <mac-address value="11:22:33:44:55:66"/>
        <device><name>115:Sensor</name></device>
        <metadata>
          <meta:device-characteristic
            ifmap-cardinality="multiValue">
            <manufacturer>Example Printer Corp</manufacturer>
            <model>21911:42100</model>
            <os>21911:ExampleOS</os>
            <os-version>21911:9.1</os-version>
            <device-type>printer</device-type>
            <discovered-time>2010-04-21T16:11:20Z</discovered-
time>
            <discoverer-id>115</discoverer-id>
            <discovery-method>scan</discovery-method>
          </meta:device-characteristic>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

6. MAP Server notifies the PDP of the device-characteristic metadata:

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <updateResult name="mac:11:22:33:44:55:66">
          <resultItem>
            <mac-address value="11:22:33:44:55:66"/>
            <device><name>115:Sensor</name></device>
            <metadata>
              <meta:device-characteristic
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="117"
```

```
                ifmap-timestamp="2010-04-21T16:11:21Z">
                <manufacturer>Example Printer Corp</manufacturer>
                <model>21911:42100</model>
                <os>21911:ExampleOS</os>
                <os-version>21911:9.1</os-version>
                <device-type>printer</device-type>
                <discovered-time>2010-04-
21T16:11:20Z</discovered-time>
                <discoverer-id>115</discoverer-id>
                <discovery-method>scan</discovery-method>
              </meta:device-characteristic>
            </metadata>
          </resultItem>
        </updateResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

7. The PDP applies its policies to the device-characteristic metadata and determines that the device belongs on the printer VLAN (VLAN 42). The PDP sends a RADIUS CoA message to the switch to change the VLAN of the port, and publishes information to the MAP server regarding the endpoint.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <update>
        <access-request name="111:33"/>
        <device><name>111:PEP</name></device>
        <metadata>
          <meta:layer2-information
            ifmap-cardinality="multiValue">
            <vlan>42</vlan>
            <port>16</port>
          </meta:layer2-information>
        </metadata>
      </update>
      <update>
        <access-request name="111:33"/>
        <mac-address value="11:22:33:44:55::66"/>
        <metadata>
          <meta:access-request-mac
            ifmap-cardinality="singleValue"/>
        </metadata>
      </update>
      <update>
        <access-request name="111:33"/>
        <metadata>
          <meta:capability ifmap-cardinality="multiValue">
            <name>print-destination</name>
```

```
          </meta:capability>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

8. Sometime later, the Sensor notices that the endpoint is no longer behaving like a printer, but instead it looks like an IP phone. The Sensor replaces the device-characteristic metadata that it published for the endpoint.

```
<?xml version="1.0"?>
<env:Envelope
 xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
 xmlns:env="http://www.w3.org/2003/05/soap-envelope"
 xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="789">
      <delete
        filter="meta:device-characteristic[@ifmap-publisher-
id='115]">
        <mac-address value="11:22:33:44:55::66"/>
        <device><name>115:Sensor</name></device>
      </delete>
      <update>
        <mac-address value="11:22:33:44:55:66"/>
        <device><name>115:Sensor</name></device>
        <metadata>
          <meta:device-characteristic
            ifmap-cardinality="multiValue">
            <device-type>phone</device-type>
            <discovered-time>2010-04-21T16:42:19Z</discovered-
time>
            <discoverer-id>115</discoverer-id>
            <discovery-method>scan</discovery-method>
          </meta:device-characteristic>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

9. The PDP is notified of the changed metadata by the MAP Server

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:response>
      <pollResult>
        <deleteResult name="mac:11:22:33:44:55::66">
          <resultItem>
```

```
            <mac-address value="11:22:33:44:55::66"/>
            <device><name>111:Sensor</name></device>
            <metadata>
              <meta:device-characteristic
                ifmap-cardinality="multiValue"
                  ifmap-publisher-id="115"
                  ifmap-timestamp="2010-04-21T16:11:21Z">
                <manufacturer>Example Printer Corp</manufacturer>
                <model>21911:42100</model>
                <os>21911:ExampleOS</os>
                <os-version>21911:9.1</os-version>
                <device-type>printer</device-type>
                <discovered-time>2010-04-
21T16:11:20Z</discovered-time>
                <discoverer-id>115</discoverer-id>
                <discovery-method>scan</discovery-method>
              </meta:device-characteristic>
            </metadata>
          </resultItem>
        </deleteResult>
        <updateResult name="mac:11:22:33:44:55::66">
          <resultItem>
            <mac-address value="11:22:33:44:55::66"/>
            <device><name>115:Sensor</name></device>
            <metadata>
              <meta:device-characteristic
                ifmap-cardinality="multiValue"
                ifmap-publisher-id="115"
                ifmap-timestamp="2010-04-21T16:11:21Z">
                <device-type>phone</device-type>
                <discovered-time>2010-04-
21T16:42:19Z</discovered-time>
                <discoverer-id>115</discoverer-id>
                <discovery-method>scan</discovery-method>
              </meta:device-characteristic>
            </metadata>
          </resultItem>
        </updateResult>
      </pollResult>
    </ifmap:response>
  </env:Body>
</env:Envelope>
```

10. The PDP responds by reconfiguring the switch device's switch port to be on the phone VLAN. The PDP publishes new layer2-information and new capability metadata. Since layer2-information and capability are both multi-valued metadata, the PDP's publish request includes delete operations to delete the old metadata along with update operations to publish the new metadata. Because the MAP Server treats all operations within a publish request as a single atomic change, MAP Clients whose subscriptions match the changes see both the delete and the update in a single pollResult.

```
<?xml version="1.0"?>
<env:Envelope
  xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
```

```
  xmlns:meta="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">
  <env:Body>
    <ifmap:publish session-id="222">
      <delete
        filter="meta:layer2-information[@ifmap-publisher-
id='111']">
        <access-request name="111:33"/>
        <device><name>111:PEP</name></device>
      </delete>
      <delete filter="meta:capability[@publisherid='111']">
        <access-request name="111:33"/>
      </delete>
      <update>
        <access-request name="111:33"/>
        <device><name>111:PEP</name></device>
        <metadata>
          <meta:layer2-information
            ifmap-cardinality="multiValue">
            <vlan>44</vlan>
            <port>16</port>
          </meta:layer2-information>
        </metadata>
      </update>
      <update>
        <access-request name="111:33"/>
        <metadata>
          <meta:capability ifmap-cardinality="multiValue">
            <name>phone-call</name>
          </meta:capability>
        </metadata>
      </update>
    </ifmap:publish>
  </env:Body>
</env:Envelope>
```

10 IF-MAP Metadata for Network Security Schema

10.1 Standard Metadata Types

```
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ifmap="http://www.trustedcomputinggroup.org/2010/IFMAP/2"
  xmlns="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2"

targetNamespace="http://www.trustedcomputinggroup.org/2010/IFMAP-
METADATA/2">

  <!-- Schema for IF-MAP Metadata for Network Security -->

  <!-- access-request-device is link metadata that
       associates an access-request identifier with
       a device identifier -->
  <xsd:element name="access-request-device">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- access-request-ip is link metadata that
       associates an access-request identifier with
       an ip-address identifier -->
  <xsd:element name="access-request-ip">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- access-request-mac is link metadata that
       associates an access-request identifier with
       a mac-address identifier -->
  <xsd:element name="access-request-mac">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- authenticated-as is link metadata that
       associates an access-request identifier with
       an identity identifier -->
  <xsd:element name="authenticated-as">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- authenticated-by is link metadata that
```

```
       associates an access-request identifier with
       the device identifier of the PDP that
       authenticated the access-request -->
  <xsd:element name="authenticated-by">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- capability is access-request metadata that names
       a collection of privileges assigned to an endpoint -->
  <xsd:element name="capability">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="administrative-domain"
type="xsd:string" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- device-attribute is link metadata that associates
       an access-request identifier with a device identifier
       and which includes information about the device such
       as its health -->
  <xsd:element name="device-attribute">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- device-characteristic is link metadata that associates
       an access-request, ip-address, or mac-address identifier
       of an endpoint with the device identifier of the MAP
       Client publishing the metadata, which includes information
       about what kind of device is represented by the
       mac-address, ip-address, or access-request -->
  <xsd:element name="device-characteristic">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="manufacturer" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="model" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="os" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
```

```
        <xsd:element name="os-version" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="device-type" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="discovered-time" type="xsd:dateTime"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="discoverer-id" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="discovery-method" type="xsd:string"
minOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- device-ip is link metadata that associates a device
       identifier of a PDP with an IP address which it has
       authenticated -->
  <xsd:element name="device-ip">
    <xsd:complexType>
      <xsd:attributeGroup
        ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- discovered-by is link metadata that associates
       an ip-address or mac-address identifier of an endpoint
       with the device identifier of a MAP Client that has
       noticed the endpoint on the network -->
  <xsd:element name="discovered-by">
    <xsd:complexType>
      <xsd:attributeGroup
ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- enforcement-report is link metadata between a device
       identifier of a PEP or flow controller and an ip-address
       or mac-address identifier of an endpoint, indicating an
       enforcement action in progress -->
  <xsd:element name="enforcement-report">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="enforcement-action" type="xsd:string"
minOccurs="1" maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="block"/>
              <xsd:enumeration value="quarantine"/>
              <xsd:enumeration value="other"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="other-type-definition"
type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

```
        <xsd:element name="enforcement-reason" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
        ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element

  <!-- event is access-request, identity, ip-address, or
       mac-address metadata that describes activity of
       interest detected on the network -->
  <xsd:element name="event">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="discovered-time" type="xsd:dateTime"
minOccurs="1" maxOccurs="1"/>
      <xsd:element name="discoverer-id" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="magnitude" minOccurs="1"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:minInclusive value="0"/>
              <xsd:maxInclusive value="100"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="confidence" minOccurs="1"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:minInclusive value="0"/>
              <xsd:maxInclusive value="100"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="significance" minOccurs="1"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="critical"/>
              <xsd:enumeration value="important"/>
              <xsd:enumeration value="informational"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="type" minOccurs="0" maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="p2p"/>
              <xsd:enumeration value="cve"/>
              <xsd:enumeration value="botnet infection"/>
              <xsd:enumeration value="worm infection"/>
              <xsd:enumeration value="excessive flows"/>
```

```
              <xsd:enumeration value="behavioral change"/>
              <xsd:enumeration value="policy violation"/>
              <xsd:enumeration value="other"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="other-type-definition"
type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="information" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="vulnerability-uri" type="xsd:anyURI"
minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

    <!-- ip-mac is link metadata that associates an
       ip-address identifier with a mac-address identifier
       and which includes optional DHCP lease information -->
  <xsd:element name="ip-mac">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="start-time" type="xsd:dateTime"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="end-time" type="xsd:dateTime"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="dhcp-server" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

<!-- layer2-information is link metadata that
       associates an access-request identifier with
       the device identifier of the PEP through
       which the endpoint is accessing the network -->
  <xsd:element name="layer2-information">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="vlan" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="vlan-name" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="port" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="administrative-domain"
type="xsd:string" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>
```

```
  <!-- location indicates information about the location of an
       identity, ip-address, or mac-address -->
  <xsd:element name="location">
    <xsd:complexType>
      <xsd:sequence>
      <xsd:element name="location-information" minOccurs="1"
maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:attribute name="type" type="xsd:string"/>
              <xsd:attribute name="value" type="xsd:string"/>
            </xsd:complexType>
          </xsd:element>
        <xsd:element name="discovered-time" type="xsd:dateTime"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="discoverer-id" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- request-for-investigation is link metadata that associates
       an ip-address or mac-address identifier with a device
       identifier representing a PDP that would like a Sensor to
       investigate the ip-address or mac-address -->
  <xsd:element name="request-for-investigation">
    <xsd:complexType>
      <xsd:attribute name="qualifier" type="xsd:string"
use="optional"/>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- role is link metadata that associates an
       access-request identifier with an identity
       identifier and which names collections of
       privileges associated with the end-user -->
  <xsd:element name="role">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="administrative-domain"
type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="name" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

    <xsd:simpleType name="wlan-security-enum">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="open"/>
```

```
        <xsd:enumeration value="wep"/>
        <xsd:enumeration value="tkip"/>
        <xsd:enumeration value="ccmp"/>
        <xsd:enumeration value="bip"/>
        <xsd:enumeration value="other"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="wlan-security-type">
      <xsd:simpleContent>
        <xsd:extension base="wlan-security-enum">
          <xsd:attribute name="other-type-definition"
type="xsd:string" use="optional"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

  <!-- wlan-information is link metadata between an access-
       request and device, and indicates properties of the
       wireless LAN connection of the access-request -->
  <xsd:element name="wlan-information">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ssid" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="ssid-unicast-security" minOccurs="1"
maxOccurs="unbounded" type="wlan-security-type">
        </xsd:element>
        <xsd:element name="ssid-group-security" minOccurs="1"
maxOccurs="1" type="wlan-security-type"/>
        <xsd:element name="ssid-management-security"
minOccurs="1" maxOccurs="unbounded" type="wlan-security-type"/>
      </xsd:sequence>
      <xsd:attributeGroup
      ref="ifmap:singleValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>

  <!-- unexpected-behavior is access-request, identity, ip-
       address, or mac-address metadata that describes activity
       of interest detected on the network -->
  <xsd:element name="unexpected-behavior">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="discovered-time" type="xsd:dateTime"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="discoverer-id" type="xsd:string"
minOccurs="1" maxOccurs="1"/>
        <xsd:element name="information" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="magnitude" minOccurs="1"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:minInclusive value="0"/>
              <xsd:maxInclusive value="100"/>
            </xsd:restriction>
          </xsd:simpleType>
```

```
        </xsd:element>
        <xsd:element name="confidence" minOccurs="0"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:integer">
              <xsd:minInclusive value="0"/>
              <xsd:maxInclusive value="100"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="significance" minOccurs="1"
maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="critical"/>
              <xsd:enumeration value="important"/>
              <xsd:enumeration value="informational"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="type" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attributeGroup
ref="ifmap:multiValueMetadataAttributes"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Appendix A: Device Types

This section of the document defines types of devices for use with the device-type element of device-characteristic.

- alarm-system
- analyzer
- badge-reader
- blade
- bridge
- camera
- cash-register
- communications-gateway
- communications-server
- fax-device
- firewall
- gateway
- host
- hub
- hvac-system
- imaging-device
- kvm
- l2-switch
- l3-switch
- p/slc-device
- print-server
- printer
- probe
- remote-access-device
- rmon-probe
- router
- security-device
- server
- stealth-router
- storage
- switch
- terminal-server
- turnstile

- ups
- vending-machine
- video-device
- voip-phone
- wap-device
- wlan-ap
- wlan-controller