SACM
Internet-Draft
Intended status: Standards Track
Expires: March 5, 2017

D. Waltermire, Ed.
NIST
K. Watson
DHS
C. Kahn
L. Lorenzin
Pulse Secure, LLC
M. Cokus
D. Haynes
The MITRE Corporation
H. Birkholz
Fraunhofer SIT

September 2016

SACM Information Model
draft-ietf-sacm-information-model-??

Abstract

This document defines the Information Elements that are necessary for the transportation of endpoint information between Secure Automation and Continuous Monitoring (SACM) components, as well as tasks carried out on the endpoint information by SACM components. The primary purpose of the SACM Information Model is to ensure the interoperable exchange of security posture information by defining the structure and metadata used to exchange the posture information represented by platform and/or vendor specific collection data models. The Information Elements and corresponding types are maintained as the IANA "SACM Information Elements" registry.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The SACM Information Model (IM) serves multiple purposes:

- to ensure interoperable exchange of security posture information which is represented by different data models,

- to provide a standardized set of Information Elements - the SACM Vocabulary - to enable the exchange of content vital to automated security posture assessment, and

- to enable secure information sharing in a scalable and extensible fashion in order to support the tasks conducted by SACM components.

A complete set of requirements imposed on the IM can be found in [I-D.ietf-sacm-requirements]. The SACM IM is intended to support a standardized data exchange mechanism between SACM components (data in motion).

The information model expresses, for example, target endpoint (TE) attributes, guidance, and evaluation results metadata describing the collected posture data. The corresponding Information Elements are consumed and produced by SACM components as they carry out tasks.

The primary tasks that this information model supports (on data, control, and management plane) are:

- TE Discovery

- TE Characterization

- TE Classification

- Collection

- Evaluation

- Information Sharing

- SACM Component Discovery

- SACM Component Authentication

- SACM Component Authorization

- SACM Component Registration

These tasks are defined in [I-D.ietf-sacm-terminology].

2. Conventions Used In This Document

2.1. Requirements Language

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 [RFC2119].

2.2. Information Element Examples

The notation used to define the SACM Information Elements (IEs) is based on a customized version of the IPFIX information model syntax. It should be noted that while examples may include actual names of information elements, they are not intended to influence how corresponding SACM IEs should be defined. The examples are provided for demonstration purposes only.

3. Information Elements Overview

The IEs defined in this document comprise the building blocks by which all SACM content is composed. They are consumed and provided by SACM components on the data plane. Every Information Element has a unique label: its name. Every IE defined by the SACM IM is registered as a type at the IANA registry. The Integer Index of the IANA SMI number tables can be used by SACM data models.

3.1. Information Element Naming Convention

SACM Information Elements must adhere to the following naming conventions.

- Names SHOULD be descriptive

- Names MUST be unique within the SACM registry. Enterprise-specific names SHOULD be prefixed with a Private Enterprise Number [PEN].

- Names MUST start with lowercase letters unless it begins with a Private Enterprise Number

- Composed names MUST use capital letters for the first letter of each part

3.2. Extensibility of Information Elements

A SACM data model based on this information model MAY include additional information elements that are not defined here. The labels of additional Information Elements included in different SACM data models MUST NOT conflict with the labels of the Information Elements defined by this information model, and the names of additional Information Elements MUST NOT conflict with each other or across multiple data models. In order to avoid naming conflicts, the labels of additional IEs SHOULD be prefixed to avoid collisions across extensions. The prefix MUST include an organizational identifier and therefore, for example, MAY be an IANA enterprise number, a (partial) name space URI, or an organization name abbreviation.

3.3. Structure of Information Elements

There are two basic types of IEs:

- Attributes: are "atomic" information elements with a unique name and a simple value. Attributes can be components of Subjects.

- Subjects: are composite information elements with a unique name and are composed of attributes and/or other subjects. Every IE that is part of a subject can have a quantity associated with it (0..1, 0..*, etc.). The contents of a subject can be ordered or unordered.

Subjects can be nested and the SACM IM allows for circular or recursive nesting. The association of IEs via nesting results in a tree-like structure wherein subjects compose the root and intermediary nodes and attributes the leaves of the tree. This semantic structure does not impose a specific structure on SACM data models regarding data in motion or data repository schemata for data at rest.

3.4. Notation of Information Elements

The notation of the SACM IM is defined below and is based on a modified version of the IP Information Flow Export (IPFIX) Information Model syntax described in Section 2.1 of [RFC7012]. The customized syntax used by the SACM IM is defined below in Figure 1.

```
     elementId (required):    The numeric identifier of the
                              Information Element. It is used
                              for the compact identification
                              of an Information Element. If
                              this identifier is used without
                              an enterpriseID, then the
                              elementId must be unique, and
                              the description of allowed values
                              is administrated by IANA. The
                              value "TBD" may be used during
```

development of the information model until an elementId is assigned by IANA and filled in at publication time.

enterpriseId (optional): Enterprises may wish to define Information Elements without registering them with IANA, for example, for enterprise-internal purposes. For such Information Elements, the elementId is not sufficient when used outside the enterprise. If specifications of enterprise-specific Information Elements are made public and/or if enterprise-specific identifiers are used by SACM components outside the enterprise, then the enterprise-specific identifier MUST be made globally unique by combining it with an enterprise identifier. Valid values for the enterpriseId are defined by IANA as Structure of Management Information (SMI) network management private enterprise numbers.

name (required): A unique and meaningful name for the Information Element.

dataType (required): There are two kinds of datatypes: simple and structured. Attributes are defined using simple datatypes and subjects are defined using structured datatypes. The contents of the datatype field will be either a reference to one of the simple datatypes listed in Section 5.1, or the specification of structured datatype as defined in Section 5.2.

status (required): The status of the specification of the Information Element. Allowed values are "current" and "deprecated". All newly defined Information Elements have "current"

```
                                status. The process for moving
                                Information Elements to the
                                "deprecated" status is TBD.

       description (required): Describes the meaning of the
                               Information Element, how it is
                               derived, conditions for its use,
                               etc.

       structure (optional):   A parsable property that provides
                               details about the definition of
                               structured Information Elements as
                               described in Section 5.2.

       references (optional):  Identifies other RFCs or documents
                               outside the IETF which provide
                               additional information or context
                               about the Information Element.
```

Figure 1: Information Element Specification Template

3.5. Categories

Categories are special IEs which represent a choice among multiple IEs via just one name. A prominent example of a category is network-address. Network- address is a category that every kind of network address is associated with, e.g. mac-address, ipv4-address, ipv6-address, or typed-network-address. If a subject includes network-address as one of its components, any of the category members are valid to be used in its place.

Another prominent example is EndpointIdentifier. Some IEs can be used to identify (and over time re-recognize) target endpoints - those are associated with the category endpoint-identifier.

4. Predefined SACM Subjects

The SACM IM provides two conceptual top-level subjects that are used to ensure a homogeneous structure for SACM content and its associated metadata: SACM statements and SACM content elements. Every set of IEs that is provided by a SACM component must provide the information contained in these two subjects although it is up to the implementer whether or not the subjects are explicitly defined in a data model.

In general, every piece of information that enables security posture assessment or further enriches the quality of the assessment process can be associated with metadata. In the SACM IM, metadata is represented by special, predefined subjects and is bundled with other attributes or subjects to provide additional information about them. The IM explicitly defines two kinds of metadata: sacmMetadata, focusing on the data origin (the SACM component that provides the information to the SACM domain), and contentMetadata, focusing on the data source (the target endpoint that is assessed).

Metadata can also include relationships that refer to other associated IEs (or SACM content in general) by using referencing labels that have to be included in the metadata of the associated IE.

Lastly, event subjects provide a structure to represent the change of IE values that was detected by a collection task at a specific point of time.

4.1. SACM Content Elements

Every piece of information that is provided by a SACM component is always associated with a set of metadata, for example, the timestamp at which this set of information was produced (e.g. by a collection task) or what target endpoint this set of information is about (e.g. the data-source or a target endpoint identifier, respectively). The subject that associates content IE with content-metadata IE is called a content-element. Content metadata can also include relationships that express associations with other content-elements.

```
contentElement = (
  contentMetadata = (
    collectionTimestamp = 146193322,
    dataSource = fb02e551-7101-4e68-8dec-1fde6bd10981
  ),
  collectedData
  )
)
```

Figure 2: Example of collected data associated with a timestamp and a target endpoint label.

4.2. SACM Statements

One or more SACM content elements are bundled in a SACM statement. In contrast to content-metadata, statement-metadata focuses on the providing SACM component instead of the target endpoint that the content is about. The only content-specific metadata included in the SACM statement is the content-type IE. Therefore, multiple content-elements that share the same statement metadata and are of the same content-type can be included in a single SACM statement. A SACM statement functions similar to an envelope or a header. Its purpose is to enable the tracking of the origin of data inside a SACM domain and more importantly to enable the mitigation of conflicting information that may originate from different SACM components. How a consuming SACM component actually deals with conflicting information is out-of-scope of the SACM IM. Semantically, the term statement implies that the SACM content provided by a SACM component might not be correct in every context, but rather is the result of a best-effort to produce correct information.

```
            sacmStatement = (
              statementMetadata = (
                publishTimestamp = 1461934031,
                dataOrigin = 24e67957-3d31-4878-8892-da2b35e121c2,
                content-type = observation
              ),
              contentElement = (
                contentMetadata = (
                  collectionTimestamp = 146193322,
                  dataSource = fb02e551-7101-4e68-8dec-1fde6bd10981
                ),
                collectedData
              )
              contentElement = (
                contentMetadata = (
                  collectionTimestamp = 146193324,
                  dataSource = fb02e001-7104-4e68-8dec-1fde6bd10981
                ),
                collectedData
              )
            )
```

Figure 3: Example of a simple SACM statement including content elements.

4.3. Relationships

An IE can be associated with another IE, e.g. a user-name attribute can be associated with a content-authorization subject. These references are expressed via the relationships subject, which can be included in a corresponding content-metadata subject. The relationships subject includes a list of one or more references. The SACM IM does not enforce a SACM domain to use unique identifiers as references. Therefore, there are at least two ways to reference another

- o The value of a reference represents a specific content-label that is unique in a SACM domain (and has to be included in the corresponding content-element metadata in order to be referenced), or

- The reference is a subject that includes an appropriate number of IEs in order to identify the referenced content-element by its actual content.

It is recommended to provide unique identifiers in a SACM domain and the SACM IM provides a corresponding naming-convention as a reference in Section 3.1. The alternative highlighted above summarizes a valid approach that does not require unique identifiers and is similar to the approach of referencing target endpoints via identifying attributes included in a characterization record.

```
contentElement = (
  contentMetadata = (
    collectionTimestamp = 1461934031,
    teLabel =
    fb02e551-7101-4e68-8dec-1fde6bd10981
    relationships = (
      associated-with-user-account =
      f3d70ef4-7e18-42af-a894-8955ba87c95d
    )
  ),
  collectedData
)

contentElement = (
  contentMetadata = (
    contentLabel = f3d70ef4-7e18-42af-a894-8955ba87c95d
  ),
  collectedData
  )
)
```

Figure 4: Example instance of a content-element subject associated with another subject via its content metadata.

4.4. Event

Event subjects provide a structure to represent the change of IE values that was detected by a collection task at a specific point of time. It is mandatory to include the new values and the collection timestamp in an event subject and it is recommended to include the past values and a collection timestamp that were replaced by the new IE values. Every event can also be associated with a subject-specific event-timestamp and a lastseen-timestamp that might differ from the corresponding collection-timestamps. If these are omitted the collection-timestamp that is included in the content-metadata subject is used instead.

```
           sacmStatement = (
             statementMetadata = (
               publishTimestamp = 1461934031,
               dataOrigin = 24e67957-3d31-4878-8892-da2b35e121c2,
               contentType = event
             ),
             event = (
               eventAttributes = (
                 eventName = "host-name change",
                 contentElement = (
                   contentMetadata = (
                   collectionTimestamp = 146193322,
                   dataSource =
                     fb02e551-7101-4e68-8dec-1fde6bd10981,
                     eventComponent = past-state
                  ),
                  collectedData
                 ),
                 contentElement = (
                   contentMetadata = (
                     collectionTimestamp = 146195723,
                     dataSource =
                     fb02e551-7101-4e68-8dec-1fde6bd10981,
                     eventComponent = current-state
                   ),
                   collectedData
                 )
               )
             )
```

Figure 5: Example of a SACM statement containing an event.

5. Abstract Data Types

This section describes the set of valid abstract data types that can be used for the specification of the SACM Information Elements in Section 7. SACM currently supports two classes of datatypes that can be used to define Information Elements.

- o Simple: Datatypes that are atomic and are used to define the type of data represented by an attribute Information Element.

- o Structured: Datatypes that can be used to define the type of data represented by a subject Information Element.

Note that further abstract data types may be specified by future extensions of the SACM information model.

5.1. Simple Datatypes

5.1.1. IPFIX Datatypes

To facilitate the use of existing work, SACM supports the following abstract data types defined in Section 3 of [RFC7012].

- unsigned8, unsigned16, unsigned32, unsigned64
- signed8, signed16, signed32, signed64
- float32, float64
- boolean
- macAddress
- octetArray
- string
- dateTimeSeconds, dateTimeMilliseconds, dateTimeMicroseconds, dateTimeNanoSeconds
- ipv4Address, ipv6Address

5.2. Structured Datatypes

5.2.1. List Datatypes

SACM defines the following abstract list data types that are used to represent the structured data associated with subjects.

- list: indicates that the Information Element order is not significant but MAY be preserved.
- orderedList: indicates that Information Element order is significant and MUST be preserved.

The notation for defining a SACM structured datatype is based on regular expressions, which are composed of the keywords "list" or "orderedList" and an Information Element expression. IE expressions use some of the regular expression syntax and operators, but the terms in the expression are the names of defined Information Elements instead of character classes. The syntax for defining list and orderedList datatypes is described below, using BNF:

```
    <list-def> -> ("list"|"orderedList") "(" <ie-expression> ")"

    <ie-expression> -> <ie-name> <cardinality>?
                       ( ("," | "|") <ie-name> <cardinality>?)*

    <cardinality> -> "*" | "+" | "?" |
                     ( "(" <non-neg-int> ("," <non-neg-int>)? ")" )
```

Figure 6: Syntax for Defining List Datatypes

As seen above, multiple occurences of an Information Element may be present in a structured datatype. The cardinality of an Information Element within a structured Information Element definition is defined by the following operators:

```
    * - zero or more occurrences

    + - one or more occurrences

    ? - zero or one occurrence

   (m,n) - between m and n occurrences
```

Figure 7: Specifying Cardinality for Structured Datatypes

The absence of a cardinality operator implies one mandatory occurrence of the Information Element.

Below is an example of a structured Information Element definition.

```
personInfo = list(firstName, middleNames?, lastName)
firstName = string
middleNames = orderedList(middleName+)
middleName = string
lastName = string
```

As an example, consider the name "John Ronald Reuel Tolkien". Below are instances of this name, structured according to the personInfo definition.

```
personInfo = (firstName="John", middleNames(middleName="Ronald",
              middleName="Reuel"), lastName="Tolkien")

personInfo = (middleNames(middleName="Ronald", middleName=" Reuel"),
              lastName="Tolkien", firstName="John")
```

The instance below is not legal with respect to the definition of personInfo because the order in middleNames is not preserved.

```
personInfo = (firstName="John", middleNames(middleName=" Reuel",
              middleName="Ronald"), lastName="Tolkien")
```

Figure 8: Example of Defining a Structured List Datatype

5.2.2. Enumeration Datatype

SACM defines the following abstract enumeration datatype that is used to represent the restriction of an attribute value to a set of values.

```
name, hex-value, description
<enumeration-def> -> -> <name> ";" <hex-value> ";" <description>
<name> -> [0-9a-zA-Z]+
<hex-value> -> 0x[0-9a-fA-F]+
<description> -> [0-9a-zA-Z\.\,]+
```

Figure 9: Syntax for Defining an Enumeration Datatype

Below is an example of a structured Information Element definition for an enumeration.

```
                            Red    ; 0x1  ; The color is red.
                            Orange ; 0x2  ; The color is orange.
                            Yellow ; 0x3  ; The color is yellow.
                            Green  ; 0x4  ; The color is green.
                            ...
```

Figure 10: Example of Defining a Structured Enumeration Datatype

6. Information Model Assets

In order to represent the Information Elements related to the areas listed in , the information model defines the information needs (or metadata about those information needs) related to following types of assets which are defined in [I-D.ietf-sacm-terminology] (and included below for convenience) which are of interest to SACM. Specifically:

- Endpoint
- Software Component
- Hardware Component
- Identity
- Guidance
- Evaluation Results

The following figure shows the make up of an Endpoint asset which contains zero or more hardware components and zero or more software components each of which may have zero or more instances running an endpoint at any given time as well as zero or more identities that act on behalf of the endpoint when interfacing with other endpoints, tools, or services. An endpoint may also contain other endpoints in the case of a virtualized environment.

```
           +---------+*______in>_______*+-----+
           |Hardware |                  |!   !|
           |Component|   +---------+    |!   !|
           +---------+   |Software |in> |!   !|
                         |Component|____|!   !|
                         +---------+*  *|!   !|
                             1|         |!   !|
                             *|         |     |       +----------+
                         +---------+    |End- |*_____*| Identity |
                         |Software |in> |point| acts  +----------+
                         |Instance |____|     | for>
                         +---------+*  1|!   !|
                                        |!   !|
                                        |!   !|
                                        |!   !|
                                        |!   !|____
                                        |!   !|0..1|
                                        +-----+    |
                                           |*      |
                                           |_______|
                                              in>
```

Figure 11: Model of an Endpoint

6.1. Asset

As defined in [RFC4949], an asset is a system resource that is (a) required to be protected by an information system’s security policy, (b) intended to be protected by a countermeasure, or (c) required for a system’s mission.

In the scope of SACM, an asset can be composed of other assets. Examples of Assets include: Endpoints, Software, Guidance, or Identity. Furthermore, an asset is not necessarily owned by an organization.

6.2. Endpoint

From [RFC5209], an endpoint is any computing device that can be connected to a network. Such devices normally are associated with a particular link layer address before joining the network and potentially an IP address once on the network. This includes: laptops, desktops, servers, cell phones, or any device that may have an IP address.

To further clarify, an endpoint is any physical or virtual device that may have a network address. Note that, network infrastructure devices (e.g. switches, routers, firewalls), which fit the definition, are also considered to be endpoints within this document.

Physical endpoints are always composites that are composed of hardware components and software components. Virtual endpoints are composed entirely of software components and rely on software components that provide functions equivalent to hardware components.

The SACM architecture differentiates two essential categories of endpoints: Endpoints whose security posture is intended to be assessed (target endpoints) and endpoints that are specifically excluded from endpoint posture assessment (excluded endpoints).

6.3. Hardware Component

Hardware components are the distinguishable physical components that compose an endpoint. The composition of an endpoint can be changed over time by adding or removing hardware components. In essence, every physical endpoint is potentially a composite of multiple hardware components, typically resulting in a hierarchical composition of hardware components. The composition of hardware components is based on interconnects provided by specific hardware types (e.g. mainboard is a hardware type that provides local busses as an interconnect). In general, a hardware component can be distinguished by its serial number.

Examples of a hardware components include: motherboards, network interfaces, graphics cards, hard drives, etc.

6.4. Software Component

A software package installed on an endpoint (including the operating system) as well as a unique serial number if present (e.g. a text editor associated with a unique license key).

It should be noted that this includes both benign and harmful software packages. Examples of benign software components include: applications, patches, operating system kernel, boot loader, firmware, code embedded on a webpage, etc. Examples of malicious software components include: malware, trojans, viruses, etc.

6.4.1. Software Instance

A running instance of the software component (e.g. on a multi-user system, one logged-in user has one instance of a text editor running and another logged-in user has another instance of the same text editor running, or on a single-user system, a user could have multiple independent instances of the same text editor running).

6.5. Identity

Any mechanism that can be used to identify an asset during an authentication process. Examples include usernames, user and device certificates, etc. Note, that this is different than the identity of assets in the context of designation as described in Section 11.1.

6.6. Guidance

Guidance is input instructions to processes and tasks, such as collection or evaluation. Guidance influences the behavior of a SACM component and is considered content of the management plane. Guidance can be manually or automatically generated or provided. Typically, the tasks that provide guidance to SACM components have a low-frequency and tend to be sporadic. A prominent example of guidance are target endpoint profiles,but guidance can have many forms, including:

Configuration, e.g. a SACM component’s name, or a CMDB’s IPv6 address.

Profiles, e.g. a set of expected states for network behavior associated with target endpoints employed by specific users.

Policies, e.g. an interval to refresh the registration of a SACM component, or a list of required capabilities for SACM components in a specific location.

6.6.1. Collection Guidance

A collector may need guidance to govern what it collects and when. Collection Guidance provides instructions for a Collector that specifies which endpoint attributes to collect, when to collect them, and how to collect them. Collection Guidance is composed of Target Endpoint Attribute Guidance, Frequency Guidance, and Method Guidance.

- Target Endpoint Attribute Guidance: Set of endpoint attributes that are supposed to be collected from a target endpoint. The definition of the set of endpoint attributes is typically based on an endpoint characterization record.

- Frequency Guidance: Specifies when endpoint attributes are to be collected.

- Method Guidance: Indicates how endpoint attributes are to be collected.

6.6.2. Evaluation Guidance

An evaluator typically needs guidance to govern what it considers to be a good or bad security posture. Evaluation Guidance provides instructions for an Evaluator that specifies which endpoint attributes to evaluate, the desired state of those endpoint attributes, and any special requirements that enable an Evaluator to determine if the endpoint attributes can be used in the evaluation (e.g. freshness of data, how it was collected, etc.). Evaluation Guidance is composed of Target Endpoint Attribute Guidance, Expected Endpoint Attribute Value Guidance, and Frequency Guidance.

- Target Endpoint Attribute Guidance: Set of target endpoint attributes that are supposed to be used in an evaluation as well as any requirements on the endpoint attributes. The definition of the set of endpoint attributes is typically based on an endpoint characterization record.

- Expected Endpoint Attribute Value Guidance: The expected values of the endpoint attributes described in the Target Endpoint Attribute Guidance.

- Frequency Guidance: Specifies when endpoint attributes are to be evaluated.

- Method Guidance: Indicates how endpoint attributes are to be collected.

6.6.3. Classification Guidance

A SACM Component carrying out the Target Endpoint Classification Task may need guidance on how to classify an endpoint. Specifically, how to associate endpoint classes with a specific target endpoint characterization record. Target Endpoint Classes function as guidance for collection, evaluation, remediation and security posture assessment in general. Classification Guidance is composed of Target Endpoint Attribute Guidance and Class Guidance.

- Target Endpoint Attribute Guidance: Set of target endpoint attributes that are supposed to be used to identify the endpoint characterization record.

- Class Guidance: A list of target endpoint classes that are to be associated with the identified target endpoint characterization record.

6.6.4. Storage Guidance

An SACM Component typically needs guidance to govern what information it should store and where. Storage Guidance provides instructions for a SACM Component that specifies which security automation information should be stored, for how long, and on which endpoint. Storage Guidance is composed of Target Endpoint Attribute Guidance, Expected Security Automation Information Guidance, and Retention Guidance.

- Target Endpoint Attribute Guidance: Set of target endpoint attributes that are supposed to be used to identify the endpoint where the security automation information is to be stored.

- Expected Security Automation Information Guidance: The security automation information that is expected to be stored (guidance, collected posture attributes, results, etc.).

- Retention Guidance: Specifies how long the security automation information should be stored.

6.6.5. Evaluation Results

Evaluation Results are the output of comparing the actual state of an endpoint against the expected state of an endpoint. In addition to the actual results of the comparison, Evaluation Results should include the Evaluation Guidance and actual target endpoint attributes values used to perform the evaluation.

7. Information Model Elements

This section defines the specific Information Elements and relationships that will be implemented by data models and transported between SACM Components.

7.1. accessPrivilegeType

```
            elementId: TBD
            name: accessPrivilegeType
            dataType: string
            status: current
            description: A set of types that represent access
            privileges (read, write, none, etc.).
```

7.2. accountName

elementId: TBD
name: accountName
dataType: string
status: current
description: A label that uniquely identifies an account that can require some form of (user) authentication to access.

7.3. administrativeDomainType

elementId: TBD
name: accessPrivilegeType
dataType: string
status: current
description: A label the is supposed to uniquely identify an administrative domain.

7.4. addressAssociationType

elementId: TBD
name: accessPrivilegeType
dataType: string
status: current
description: A label the is supposed to uniquely identify an administrative domain.

7.5. addressMaskValue

elementId: TBD
name: addressMaskValue
dataType: string
status: current
description: A value that expresses a generic address subnetting bitmask.

7.6. addressType

elementId: TBD
name: addressType
dataType: string
status: current
description: A set of types that specifies the type of address that is expressed in an address subject (e.g. ethernet, modbus, zigbee).

7.7. addressValue

```
               elementId: TBD
               name: addressValue
               dataType: string
               status: current
               description: A value that expresses a generic network
                            address.
```

7.8. authenticator

```
               elementId: TBD
               name: authenticator
               dataType: string
               status: current
               description: A label that references a SACM component
               that can authenticate target endpoints (can be used in
               a target-endpoint subject to express that the target
               endpoint was authenticated by that SACM component.
```

7.9. authenticationType

```
               elementId: TBD
               name: authenticationType
               dataType: string
               status: current
               description: A set of types that expresses which type
               of authentication was used to enable a network
               interaction/connection.
```

7.10. certificate

```
               elementId: TBD
               name: certificate
               dataType: string
               status: current
               description: A value that expresses a certificate that
               can be collected from a target endpoint.
```

7.11. collectionTaskType

```
               elementId: TBD
               name: collectionTaskType
               dataType: string
               status: current
               description: A set of types that defines how collected
               SACM content was acquired (e.g. network-observation,
               remote-acquisition, self-reported).
```

7.12. confidence

elementId: TBD
name: confidence
dataType: string
status: current
description: A representation of the subjective probability that the assessed value is correct. If no confidence value is given, it is assumed that the confidence is 1. Acceptable values are between 0 and 1.

7.13. contentAction

elementId: TBD
name: contentAction
dataType: string
status: current
description: A set of types that express a type of action (e.g. add, delete, update). It can be associated, for instance, with an event subject or with a network observation.

7.14. dataOrigin

elementId: TBD
name: dataOrigin
dataType: string
status: current
description: A label that uniquely identifies a SACM component in and across SACM domains.

7.15. dataSource

elementId: TBD
name: dataSource
dataType: string
status: current
description: A label that is supposed to uniquely identify the data source (e.g. a target endpoint or sensor) that provided an initial endpoint attribute record.

7.16. discoverer

elementId: TBD
name: contentAction
dataType: string
status: current
description: A label that refers to the SACM component that discovered a target endpoint (can be used in a target-endpoint subject to express, for example, that the target endpoint was authenticated by that SACM component).

7.17. eventType

elementId: TBD
name: eventType
dataType: string
status: current
description: a set of types that define the categories of an event (e.g. access-level-change, change-of-priviledge, change-of-authorization, environmental-event, or provisioning-event).

7.18. eventThreshold

elementId: TBD
name: eventThreshold
dataType: string
status: current
description: if applicable, a value that can be included in an event subject to indicate what numeric threshold value was crossed to trigger that event.

7.19. eventThresholdName

elementId: TBD
name: eventThresholdName
dataType: string
status: current
description: If an event is created due to a crossed threshold, the threshold might have a name associated with it that can be expressed via this value.

7.20. eventTrigger

elementId: TBD
name: eventTrigger
dataType: string
status: current
description: This value is used to express more complex trigger conditions that may cause the creation of an event.

7.21. eventTrigger

elementId: TBD
name: eventTrigger
dataType: string
status: current
description: This value is used to express more complex trigger conditions that may cause the creation of an event.

7.22. firmwareId

elementId: TBD
name: firmwareId
dataType: string
status: current
description: A label that represents the BIOS or firmware ID of a specific target endpoint.

7.23. hostName

elementId: TBD
name: hostName
dataType: string
status: current
description: A label typically associated with an endpoint, but, not always intended to be unique given scope.

7.24. interfaceLabel

elementId: TBD
name: interfaceLabel
dataType: string
status: current
description: A unique label that can be used to reference a network interface.

7.25. ipv6AddressSubnetMask

```
elementId: TBD
name: ipv6AddressSubnetMask
dataType: string
status: current
description: An IPv6 subnet bitmask.
```

7.26. ipv6AddressSubnetMaskCidrNotation

```
elementId: TBD
name: ipv6AddressSubnetMaskCidrNotation
dataType: string
status: current
description: An IPv6 subnet bitmask in CIDR notation.
```

7.27. ipv6AddressValue

```
elementId: TBD
name: ipv6AddressValue
dataType: ipv6Address
status: current
description: An IPv6 subnet bitmask in CIDR notation.
a network interface.
```

7.28. ipv4AddressSubnetMask

```
elementId: TBD
name: ipv4AddressSubnetMask
dataType: string
status: current
description: An IPv4 subnet bitmask.
```

7.29. ipv4AddressSubnetMaskCidrNotation

```
elementId: TBD
name: ipv4AddressSubnetMaskCidrNotation
dataType: string
status: current
description: An IPv4 subnet bitmask in CIDR notation.
```

7.30. ipv4AddressValue

```
elementId: TBD
name: ipv4AddressValue
dataType: ipv4Address
status: current
description: An IPv4 address value.
```

7.31. layer2InterfaceType

elementId: TBD
name: layer2InterfaceType
dataType: string
status: current
description: A set of types referenced by IANA ifType.

7.32. layer4PortAddress

elementId: TBD
name: layer4PortAddress
dataType: unsigned32
status: current
description: A layer 4 port address
typically associated with TCP and UDP
protocols.

7.33. layer4Protocol

elementId: TBD
name: layer4Protocol
dataType: string
status: current
description: A set of types that express a layer 4
protocol (e.g. UDP or TCP).

7.34. locationName

elementId: TBD
name: locationName
dataType: string
status: current
description: A value that represents a named region of
physical space.

7.35. macAddressValue

elementId: TBD
name: macAddressValue
dataType: string
status: current
description: A value that expresses an Ethernet address.

7.36. methodLabel

```
elementId: TBD
name: methodLabel
dataType: string
status: current
description: A label that references a specific method
registered and used in a SACM domain (e.g. method to
match and re-identify target endpoints via identifying
attributes).
```

7.37. methodRepository

```
elementId: TBD
name: methodRepository
dataType: string
status: current
description: A label that references a SACM component
methods can be registered at and that can provide
guidance in the form of registered methods to other
SACM components.
```

7.38. networkAccessLevelType

```
elementId: TBD
name: networkAccessLevelType
dataType: string
status: current
description: A set of types that expresses categories
of network access-levels (e.g. block, quarantine, etc.).
```

7.39. networkId

```
elementId: TBD
name: networkId
dataType: string
status: current
description: Most networks such as AS, OSBF domains,
or VLANs can have an ID.
```

7.40. networkInterfaceName

```
elementId: TBD
name: networkInterfaceName
dataType: string
status: current
description: A label that uniquely identifies an interface
associated with a distinguishable endpoint.
```

7.41. networkLayer

```
elementId: TBD
name: networkLayer
dataType: string
status: current
description: A set of layers that expresses the specific
network layer an interface operates on.
```

7.42. networkName

```
elementId: TBD
name: networkName
dataType: string
status: current
description: A label that is associated with a network.
Some networks, for example, effetive layer2-broadcast-domains
are difficult to "grasp" and therefore quite difficult to name.
```

7.43. organizationId

```
elementId: TBD
name: organizationId
dataType: string
status: current
description: A label that uniquely identifies an
             organization via a PEN.
```

7.44. osComponent

```
elementId: TBD
name: osComponent
dataType: string
status: current
description: A label that references a "sub-component"
that is part of the operating system (e.g. a kernel
module, microcode, or ACPI table).
```

7.45. osLabel

```
elementId: TBD
name: osLabel
dataType: string
status: current
description: A label that references a specific version
of an operating system, including patches and hotfixes.
```

7.46. osName

elementId: TBD
name: osName
dataType: string
status: current
description: The name of an operating system.

7.47. osType

elementId: TBD
name: osType
dataType: string
status: current
description: A set of types that identifies the type of an operating system (e.g. real-time, security-enhanced, consumer, server).

7.48. osVersion

elementId: TBD
name: osVersion
dataType: string
status: current
description: A value that represents the version of an operating-system.

7.49. privilegeName

elementId: TBD
name: privilegeName
dataType: string
status: current
description: The attribute name of the privilege represented as an AVP.

7.50. privilegeValue

elementId: TBD
name: privilegeValue
dataType: string
status: current
description: The value content of the privilege represented as an AVP.

7.51. protocol

```
elementId: TBD
name: protocol
dataType: string
status: current
description: A set of types that defines specific
protocols above layer 4 (e.g. http, https, dns, ipp,
or unknown).
```

7.52. publicKey

```
elementId: TBD
name: publicKey
dataType: string
status: current
description: The value of a public key (regardless of its
method of creation, crypto-system, or signature scheme)
that can be collected from a target endpoint.
```

7.53. relationshipContentElementGuid

```
elementId: TBD
name: relationshipContentElementGuid
dataType: string
status: current
description: A reference to a specific content element
used in a relationship subject.
```

7.54. relationshipStatementElementGuid

```
elementId: TBD
name: relationshipStatementElementGuid
dataType: string
status: current
description: A reference to a specific SACM statement
used in a relationship subject.
```

7.55. relationshipObjectLabel

elementId: TBD
name: relationshipObjectLabel
dataType: string
status: current
description: A reference to a specific label used in content (e.g. a te-label or a user-id). This reference is typically used if matching content attribute can be done efficiantly and can also be included in addition to a relationship-content-element-guid reference.

7.56. relationshipType

elementId: TBD
name: relationshipType
dataType: string
status: current
description: A set of types that is in every instance of a relationship subject to highlight what kind of relationship exists between the subject the relationship is included in (e.g. associated_with_user, applies_to_session, seen_on_interface, associated_with_flow, contains_virtual_device).

7.57. roleName

elementId: TBD
name: roleName
dataType: string
status: current
description: A label that references a collection of privileges assigned to a specific entity (identity? FIXME).

7.58. sessionStateType

elementId: TBD
name: sessionStateType
dataType: string
status: current
description: A set of types a discernible session (an ongoing network interaction) can be in (e.g. Authenticating, Authenticated, Postured, Started, Disconnected).

7.59. statementGuid

elementId: TBD
name: statementGuid
dataType: string
status: current
description: A label that expresses a global unique ID referencing a specific SACM statement that was produced by a SACM component.

7.60. statementType

elementId: TBD
name: statementType
dataType: string
status: current
description: A set of types that define the type of content that is included in a SACM statement (e.g. Observation, DirectoryContent, Correlation, Assessment, Guidance).

7.61. status

elementId: TBD
name: status
dataType: string
status: current
description: A set of types that defines possible result values for a finding in general (e.g. true, false, error, unknown, not applicable, not evaluated).

7.62. subAdministrativeDomain

elementId: TBD
name: subAdministrativeDomain
dataType: string
status: current
description: A label for related child domains an administrative domain can be composed of (used in the subject administrative-domain)

7.63. subInterfaceLabel

```
               elementId: TBD
               name: subInterfaceLabel
               dataType: string
               status: current
               description: A unique label a sub network interface
               (e.g. a tagged vlan on a trunk) can be referenced
               with.
```

7.64. superAdministrativeDomain

```
               elementId: TBD
               name: superAdministrativeDomain
               dataType: string
               status: current
               description: a label for related parent domains an
                            administrative domain is part of (used
                            in the subject s.administrative-domain).
```

7.65. superInterfaceLabel

```
               elementId: TBD
               name: superInterfaceLabel
               dataType: string
               status: current
               description: a unique label a super network interface
                            (e.g. a physical interface a tunnel
                            interface terminates on) can be referenced
                            with.
```

7.66. teAssessmentState

```
               elementId: TBD
               name: teAssessmentState
               dataType: string
               status: current
               description: a set of types that defines the state of
                            assessment of a target-endpoint (e.g.
                            in-discovery, discovered, in-classification,
                            classified, in-assessment, assessed).
```

7.67. teLabel

```
            elementId: TBD
            name: teLabel
            dataType: string
            status: current
            description: an identifying label created from a set
                         of identifying attributes used to reference
                         a specific target endpoint.
```

7.68. teId

```
            elementId: TBD
            name: teId
            dataType: string
            status: current
            description: an identifying label that is created
                         randomly, is supposed to be unique, and
                         used to reference a specific target
                         endpoint.
```

7.69. timestampType

```
            elementId: TBD
            name: timestampType
            dataType: string
            status: current
            description: a set of types that express what type of
                         action or event happened at that point
                         of time (e.g. discovered, classified,
                         collected, published). Can be included in
                         a generic s.timestamp subject.
```

7.70. WGS84Longitude

```
            elementId: TBD
            name: WGS84Longitude
            dataType: float
            status: current
            description: a label that represents WGS 84 rev 2004
            longitude.
```

7.71. WGS84Latitude

```
            elementId: TBD
            name: WGS84Latitude
            dataType: float
            status: current
            description: a label that represents WGS 84 rev 2004
            latitude.
```

7.72. WGS84Altitude

```
               elementId: TBD
               name: WGS84Altitude
               dataType: float
               status: current
               description: a label that represents WGS 84 rev 2004
               altitude.
```

7.73. hardwareSerialNumber

```
   elementId: TBD
   name: hardwareSerialNumber
   dataType: string
   status: current
   description: A globally unique identifier for a particular
                piece of hardware assigned by the vendor.
```

7.74. interfaceName

```
   elementId: TBD
   name: interfaceName
   dataType: string
   status: current
   description: A short name uniquely describing an interface,
                eg "Eth1/0". See [RFC2863] for the definition
                of the ifName object.
```

7.75. interfaceIndex

```
   elementId: TBD
   name: interfaceIndex
   dataType: unsigned32
   status: current
   description: The index of an interface installed on an endpoint.
                The value matches the value of managed object
                ’ifIndex’ as defined in [RFC2863]. Note that ifIndex
                values are not assigned statically to an interface
                and that the interfaces may be renumbered every time
                the device’s management system is re-initialized,
                as specified in [RFC2863].
```

7.76. interfaceMacAddress

```
   elementId: TBD
   name: interfaceMacAddress
   dataType: macAddress
   status: current
   description: The IEEE 802 MAC address associated with a network
                interface on an endpoint.
```

7.77. interfaceType

```
   elementId: TBD
   name: interfaceType
   dataType: unsigned32
   status: current
   description: The type of a network interface. The value matches
                the value of managed object 'ifType' as defined in
                [IANA registry ianaiftype-mib].
```

7.78. interfaceFlags

```
elementId: TBD
name: interfaceFlags
dataType: unsigned16
status: current
description: This information element specifies the flags
             associated with a network interface. Possible
             values include:
structure: Up                  ; 0x1   ; Interface is up.
           Broadcast           ; 0x2   ; Broadcast address valid.
           Debug               ; 0x4   ; Turn on debugging.
           Loopback            ; 0x8   ; Is a loopback net.
           Point-to-point      ; 0x10  ; Interface is point-to-point link.
           No trailers         ; 0x20  ; Avoid use of trailers.
           Resources allocated ; 0x40  ; Resources allocated.
           No ARP              ; 0x80  ; No address resolution protocol.
           Receive all         ; 0x100 ; Receive all packets.
```

7.79. networkInterface

```
   elementId: TBD
   name: networkInterface
   dataType: orderedList
   status: current
   description: Information about a network interface
                installed on an endpoint. The
                following high-level digram
                describes the structure of
                networkInterface information
                element.
   structure: orderedList(interfaceName, interfaceIndex, macAddress,
                          ifType, flags)
```

7.80. globallyUniqueIdentifier

```
   elementId: TBD
   name: globallyUniqueIdentifier
   dataType: unsigned8
   status: current
   metadata: true
   description: TODO.
```

7.81. dataOrigin

```
   elementId: TBD
   name: dataOrigin
   dataType: string
   status: current
   metadata: true
   description: The origin of the data.
```

7.82. dataSource

```
   elementId: TBD
   name: dataSource
   dataType: string
   status: current
   metadata: true
   description: The source of the data.
```

7.83. creationTimestamp

```
   elementId: TBD
   name: creationTimestamp
   dataType: dateTimeSeconds
   status: current
   metadata: true
   description: The date and time when the posture
                information was created by a SACM Component.
```

7.84. collectionTimestamp

```
   elementId: TBD
   name: collectionTimestamp
   dataType: dateTimeSeconds
   status: current
   metadata: true
   description: The date and time when the posture
                information was collected or observed by a SACM
                Component.
```

7.85. publicationTimestamp

```
   elementId: TBD
   name: publicationTimestamp
   dataType: dateTimeSeconds
   status: current
   metadata: true
   description: The date and time when the posture
                information was published.
```

7.86. relayTimestamp

```
   elementId: TBD
   name: relayTimestamp
   dataType: dateTimeSeconds
   status: current
   metadata: true
   description: The date and time when the posture
                information was relayed to another SACM Component.
```

7.87. storageTimestamp

```
   elementId: TBD
   name: storageTimestamp
   dataType: dateTimeSeconds
   status: current
   metadata: true
   description: The date and time when the posture
                information was stored in a Repository.
```

7.88. type

```
   elementId: TBD
   name: type
   dataType: enumeration
   status: current
   metadata: true
   description: The type of data model use to represent
                some set of endpoint information. The following
                table lists the set of data models supported by SACM.
   structure: TBD
```

7.89. protocolIdentifier

```
   elementId: TBD
   name: protocolIdentifier
   dataType: unsigned8
   status: current
   description: The value of the protocol number in the IP packet
                header. The protocol number identifies the IP packet
                payload type. Protocol numbers are defined in the
                IANA Protocol Numbers registry.

                In Internet Protocol version 4 (IPv4), this is
                carried in the Protocol field.  In Internet Protocol
                version 6 (IPv6), this is carried in the Next Header
                field in the last extension header of the packet.
```

7.90. sourceTransportPort

```
   elementId: TBD
   name: sourceTransportPort
   dataType: unsigned16
   status: current
   description: The source port identifier in the transport header.
                For the transport protocols UDP, TCP, and SCTP, this
                is the source port number given in the respective
                header.  This field MAY also be used for future
                transport protocols that have 16-bit source port
                identifiers.
```

7.91. sourceIPv4PrefixLength

elementId: TBD
name: sourceIPv4PrefixLength
dataType: unsigned8
status: current
description: The number of contiguous bits that are relevant in the sourceIPv4Prefix Information Element.

7.92. ingressInterface

elementId: TBD
name: ingressInterface
dataType: unsigned32
status: current
description: The index of the IP interface where packets of this Flow are being received. The value matches the value of managed object 'ifIndex' as defined in [RFC2863]. Note that ifIndex values are not assigned statically to an interface and that the interfaces may be renumbered every time the device's management system is re-initialized, as specified in [RFC2863].

7.93. destinationTransportPort

elementId: TBD
name: destinationTransportPort
dataType: unsigned16
status: current
description: The destination port identifier in the transport header. For the transport protocols UDP, TCP, and SCTP, this is the destination port number given in the respective header. This field MAY also be used for future transport protocols that have 16-bit destination port identifiers.

7.94. sourceIPv6PrefixLength

elementId: TBD
name: sourceIPv6PrefixLength
dataType: unsigned8
status: current
description: The number of contiguous bits that are relevant in the sourceIPv6Prefix Information Element.

7.95. sourceIPv4Prefix

```
   elementId: TBD
   name: sourceIPv4Prefix
   dataType: ipv4Address
   status: current
   description: IPv4 source address prefix.
```

7.96. destinationIPv4Prefix

```
   elementId: TBD
   name: destinationIPv4Prefix
   dataType: ipv4Address
   status: current
   description: IPv4 destination address prefix.
```

7.97. sourceMacAddress

```
   elementId: TBD
   name: sourceMacAddress
   dataType: macAddress
   status: current
   description: The IEEE 802 source MAC address field.
```

7.98. ipVersion

```
   elementId: TBD
   name: ipVersion
   dataType: unsigned8
   status: current
   description: The IP version field in the IP packet header.
```

7.99. interfaceDescription

```
   elementId: TBD
   name: interfaceDescription
   dataType: string
   status: current
   description: The description of an interface, eg "FastEthernet
                1/0" or "ISP
   connection".
```

7.100. exporterIPv4Address

```
   elementId: TBD
   name: exporterIPv4Address
   dataType: ipv4Address
   status: current
   description: The IPv4 address used by the Exporting Process.
                This is used by the Collector to identify the
                Exporter in cases where the identity of the Exporter
                may have been obscured by the use of a proxy.
```

7.101. exporterIPv6Address

```
   elementId: TBD
   name: exporterIPv6Address
   dataType: ipv6Address
   status: current
   description: The IPv6 address used by the Exporting Process.
                This is used by the Collector to identify the
                Exporter in cases where the identity of the
                Exporter may have been obscured by the use of a
                proxy.
```

7.102. portId

```
   elementId: TBD
   name: portId
   dataType: unsigned32
   status: current
   description: An identifier of a line port that is unique per
                IPFIX Device hosting an Observation Point.
                Typically, this Information Element is used for
                limiting the scope of other Information Elements.
```

7.103. templateId

```
   elementId: TBD
   name: templateId
   dataType: unsigned16
   status: current
   description: An identifier of a Template that is locally unique
                within a combination of a Transport session and an
                Observation Domain.

                Template IDs 0-255 are reserved for Template Sets,
                Options Template Sets, and other reserved Sets yet
                to be created. Template IDs of Data Sets are
                numbered from 256 to 65535.

                Typically, this Information Element is used for
                limiting the scope of other Information Elements.
                Note that after a re-start of the Exporting Process
                Template identifiers may be re-assigned.
```

7.104. collectorIPv4Address

```
   elementId: TBD
   name: collectorIPv4Address
   dataType: ipv4Address
   status: current
   description: An IPv4 address to which the Exporting Process sends
                Flow information.
```

7.105. collectorIPv6Address

```
   elementId: TBD
   name: collectorIPv6Address
   dataType: ipv6Address
   status: current
   description: An IPv6 address to which the Exporting Process sends
                Flow information.
```

7.106. interface

```
   elementId: TBD
   name: interface
   dataType: list
   structure: list (InterfaceName, hwAddress, inetAddr, netmask)
   status: current
   description: Represents an interface and its configuration
   options.
```

7.107. interfaceName

```
elementId: TBD
name: interfaceName
dataType: string
status: current
description: The interface
      name.
```

7.108. physicalProtocol

```
elementId: TBD
name: physicalProtocol
dataType: enumeration
structure:
ETH_P_LOOP ; 0x1 ; Ethernet loopback packet.
ETH_P_PUP ; 0x2 ; Xerox PUP packet.
ETH_P_PUPAT ; 0x3 ; Xerox PUP Address Transport packet.
ETH_P_IP ; 0x4 ; Internet protocol packet.
ETH_P_X25 ; 0x5 ; CCITT X.25 packet.
ETH_P_ARP ; 0x6 ; Address resolution packet.
ETH_P_BPQ ; 0x7 ; G8BPQ AX.25 ethernet packet.
ETH_P_IEEEPUP ; 0x8 ; Xerox IEEE802.3 PUP packet.
ETH_P_IEEEPUPAT ; 0x9 ; Xerox IEEE802.3 PUP address transport
                        packet.
ETH_P_DEC ; 0xA ; DEC assigned protocol.
ETH_P_DNA_DL ; 0xB ; DEC DNA Dump/Load.
ETH_P_DNA_RC ; 0xC ; DEC DNA Remote Console.
ETH_P_DNA_RT ; 0xD ; DEC DNA Routing.
ETH_P_LAT ; 0xE ; DEC LAT.
ETH_P_DIAG ; 0xF ; DEC Diagnostics.
ETH_P_CUST ; 0x10 ; DEC Customer use.
ETH_P_SCA ; 0x11 ; DEC Systems Comms Arch.
ETH_P_RARP ; 0x12 ; Reverse address resolution packet.
ETH_P_ATALK ; 0x13 ; Appletalk DDP.
ETH_P_AARP ; 0x14 ; Appletalk AARP.
ETH_P_8021Q ; 0x15 ; 802.1Q VLAN Extended Header.
ETH_P_IPX ; 0x16 ; IPX over DIX.
ETH_P_IPV6 ; 0x17 ; IPv6 over bluebook.
ETH_P_SLOW ; 0x18 ; Slow Protocol. See 802.3ad 43B.
ETH_P_WCCP ; 0x19 ; Web-cache coordination protocol.
ETH_P_PPP_DISC ; 0x1A ; PPPoE discovery messages.
ETH_P_PPP_SES ; 0x1B ; PPPoE session messages.
ETH_P_MPLS_UC ; 0x1C ; MPLS Unicast traffic.
ETH_P_MPLS_MC ; 0x1D ; MPLS Multicast traffic.
ETH_P_ATMMPOA ; 0x1E ; MultiProtocol Over ATM.
ETH_P_ATMFATE ; 0x1F ; Frame-based ATM Transport over Ethernet.
ETH_P_AOE ; 0x20 ; ATA over Ethernet.
```

ETH_P_TIPC ; 0x21 ; TIPC.
ETH_P_802_3 ; 0x22 ; Dummy type for 802.3 frames.
ETH_P_AX25 ; 0x23 ; Dummy protocol id for AX.25.
ETH_P_ALL ; 0x24 ; Every packet.
ETH_P_802_2 ; 0x25 ; 802.2 frames.
ETH_P_SNAP ; 0x26 ; Internal only.
ETH_P_DDCMP ; 0x27 ; DEC DDCMP: Internal only
ETH_P_WAN_PPP ; 0x28 ; Dummy type for WAN PPP frames.
ETH_P_PPP_MP ; 0x29 ; Dummy type for PPP MP frames.
ETH_P_PPPTALK ; 0x2A ; Dummy type for Atalk over PPP.
ETH_P_LOCALTALK ; 0x2B ; Localtalk pseudo type.
ETH_P_TR_802_2 ; 0x2C ; 802.2 frames.
ETH_P_MOBITEX ; 0x2D ; Mobitex.
ETH_P_CONTROL ; 0x2E ; Card specific control frames.
ETH_P_IRDA ; 0x2F ; Linux-IrDA.
ETH_P_ECONET ; 0x30 ; Acorn Econet.
ETH_P_HDLC ; 0x31 ; HDLC frames.
ETH_P_ARCNET ; 0x32 ; 1A for ArcNet.
; 0x33 ; The empty string value is permitted here to allow for detailed error reporting.
status: current
description: The physical layer protocol used by the AF_PACKET socket.

7.109. hwAddress

elementId: TBD
name: hwAddress
dataType: string
status: current
description: The hardware address associated with the interface.

7.110. transportProtocol

elementId: TBD
name: transportProtocol
dataType: string
status: current
description: The transport-layer protocol (tcp or udp).

7.111. localAddress

elementId: TBD
name: localAddress
dataType: ipAddress
status: current
description: This is the IP address being listened to. Note that the IP address can be IPv4 or IPv6.

7.112. localPort

elementId: TBD
name: localPort
dataType: integer
status: current
description: This is the TCP or UDP port being listened to.

7.113. localFullAddress

elementId: TBD
name: localFullAddress
dataType: string
status: current
description: The IP address and network port on which the program listens, including the local address and the local port. Note that the IP address can be IPv4 or IPv6.

7.114. foreignAddress

elementId: TBD
name: foreignAddress
dataType: ipAddress
status: current
description: The IP address with which the program is communicating, or with which it will communicate. Note that the IP address can be IPv4 or IPv6.

7.115. foreignFullAddress

elementId: TBD
name: foreignFullAddress
dataType: ipAddress
status: current
description: The IP address and network port to which the program is communicating or will accept communications from, including the foreign address and foreign port. Note that the IP address can be IPv4 or IPv6.

8. Acknowledgements

Many of the specifications in this document have been developed in a public-private partnership with vendors and end-users. The hard work of the SCAP community is appreciated in advancing these efforts to their current level of adoption.

Over the course of developing the initial draft, Brant Cheikes, Matt Hansbury, Daniel Haynes, Scott Pope, Charles Schmidt, and Steve Venema have contributed text to many sections of this document.

9. IANA Considerations

This document specifies an initial set of Information Elements for SACM in Section 7. An Internet Assigned Numbers Authority (IANA) registry will be created and populated with the Information Elements in Section 7. New assignments for SACM Information Elements will be administered by IANA through Expert Review [RFC2434]. The designated experts MUST check the requested Information Elements for completeness and accuracy of the submission with respect to the template and requirements expressed in Section 3.3 and Section 3.1. Requests for Information Elements that duplicate the functionality of existing Information Elements SHOULD be declined. The smallest available Information Element identifier SHOULD be assigned to a new Information Element. The definition of new Information Elements MUST be published using a well-established and persistent publication medium.

10. Security Considerations

Posture Assessments need to be performed in a safe and secure manner. In that regard, there are multiple aspects of security that apply to the communications between components as well as the capabilities themselves. This information model only contains an initial listing of items that need to be considered with respect to security and will need to be augmented as the model continues to be developed.

Security considerations include:

Authentication: Every SACM Component and asset needs to be able to identify itself and verify the identity of other SACM Components and assets.

Confidentiality: Communications between SACM Components need to be protected from eavesdropping or unauthorized collection. Some communications between SACM Components and assets may need to be protected as well.

Integrity: The information exchanged between SACM Components needs to be protected from modification. Some exchanges between assets and SACM Components will also have this requirement.

Restricted Access: Access to the information collected, evaluated, reported, and stored should only be viewable and consumable to authenticated and authorized entities.

Considerations with respect to the operational aspects of collection, evaluation, and storage security automation information can be found in Section 11.

Considerations concerning the privacy of security automation information can be found in Section 12.

11. Operational Considerations

The following sections outline a series of operational considerations for SACM deployments within an organization. This section may be expanded to include other considerations as the WG gains additional operational experience with SACM deployments and extending the information model.

11.1. Endpoint Designation

In order to successfully carry out endpoint posture assessment, it is necessary to be able to identify the endpoints on a network and track the changes to them over time. Specifically, enabling SACM Components to:

o Tell whether two endpoint attribute assertions concern the same endpoint

o Respond to compliance measurements, for example by reporting, remediating, and quarantining (SACM does not specify these responses, but SACM exists to enable them).

Ideally, every endpoint would be identified by a unique identifier present on the endpoint, but, this is complicated due to different factors such as the variety of endpoints on a network, the ability of tools to reliably access such an identifer, and the ability of tools to correlate disparate identifiers. As a result, it is necessary for an endpoint to be identified by a set of attributes that uniquely identify it on a network. The set of attributes that uniquely identify an endpoint on a network will likely vary by organization; however, there are a number of properties to consider when selecting identifying attributes as some are better suited for identification purposes than others.

Multiplicity: Is the attribute typically associated with a single endpoint or with multiple endpoints? If the attribute is associated with a single endpoint, it is better for identifying an endpoint on a network.

Persistence: How likely is the attribute to change? Does it never change? Does it only change when the endpoint is reprovisioned? Does it only change due to an event? Does it change on an ad-hoc and often unpredictable basis? Does it constantly change? The less likely it is for an attribute to change over time, the better it is for identifying an endpoint on a network.

Immutability: How difficult is it to change the attribute? Is the attribute hardware rooted and never changes? Can the attribute be changed by a user/process with the appropriate access? Can the attribute be changed without controlled access. The less likely an attribute is to change over time, the better chance it will be usable to identify an endpoint over time.

Verifiable: Can the attribute be corroborated? Can the attribute be externally verified with source authentication? Can the attribute be externally verified without source authentication? Is it impossible to externally verify the attribute. Attributes that can be externally verified are more likely to be accurate and are better for identifying endpoints on a network.

With that said, requiring SACM Components and end users to constantly refer to a set of attributes to identify an endpoint, is particularly burdensome. As a result, SACM supports the concept of a target endpoint label which associates an identifier (unique to a SACM domain) with the set of attributes used by an organization to identify endpoints on a network. Once defined for an endpoint, the target endpoint label can be used in place of the set of identifying attributes.

11.2. Timestamp Accuracy

An organization will likely have different collectors deployed across the network that will be configured to collect posture attributes on varying frequencies (periodic, ad-hoc, event-driven, on endpoint, off endpoint, etc.). Some collectors will detect changes as soon as they occur whereas others will detect them at a later point during a periodic scan or when an event has triggered the collection of posture attributes. Furthermore, some changes will be detected on the endpoint and others will be observed off of the endpoint. As a result of these differences, the accuracy of the timestamp associated with the collected information will vary. For example, if a collector is only running once every 12 hours, the change probably happened at some point in time prior to the scan and the timestamp is likely not accurate. Due to this, it is important for system administrators to determine if the accuracy of a timestamp is good enough for their intended purposes.

12. Privacy Considerations

In the IETF, there are privacy concerns with respect to endpoint identity and monitoring. This is especially true when the activity on an endpoint can be linked to a particular person. For example, by correlating endpoint attributes such as usernames, certificates, etc. with browser activity, it may be possible to gain insight in to user behavior and trends beyond what is required to carry out endpoint posture assessments. In the hands of the wrong person, this information could be used to negatively influence a user’s behavior or to plan attacks against the organization’s infrastructure.

As a result, SACM data models should incorporate a mechanism by which an organization can designate which endpoint attributes are considered sensitive with respect to privacy. This will allow SACM Components to handle endpoint attributes in a manner consistent with the organization’s privacy policies. Furthermore, organization’s should put the proper mechanism in place to ensure endpoint attributes are protected when transmitted, stored, and accessed to ensure only authorized parties are granted access.

It should also be noted that some of this is often mitigated by organizational policies that require a user of an organization’s network to consent to some level of monitoring in return for access to the network and other resources. The information that is monitored and collected will vary by organization and further highlights the need for a mechanism by which an organization can specify what constitutes privacy sensitive information for them.

13. References

13.1. Normative References

[PEN] Internet Assigned Numbers Authority, "Private Enterprise Numbers", July 2016, <https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

13.2. Informative References

[I-D.ietf-sacm-requirements] Cam-Winget, N. and L. Lorenzin, "Secure Automation and Continuous Monitoring (SACM) Requirements", draft-ietf-sacm-requirements-01 (work in progress), October 2014.

[I-D.ietf-sacm-terminology] Waltermire, D., Montville, A., Harrington, D., and N. Cam-Winget, "Terminology for Security Assessment", draft-ietf-sacm-terminology-05 (work in progress), August 2014.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, DOI 10.17487/RFC2434, October 1998, <http://www.rfc-editor.org/info/rfc2434>.

[RFC3580] Congdon, P., Aboba, B., Smith, A., Zorn, G., and J. Roese, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines", RFC 3580, DOI 10.17487/RFC3580, September 2003, <http://www.rfc-editor.org/info/rfc3580>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

[RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, DOI 10.17487/RFC5209, June 2008, <http://www.rfc-editor.org/info/rfc5209>.

[RFC5793] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5793, DOI 10.17487/RFC5793, March 2010, <http://www.rfc-editor.org/info/rfc5793>.

[RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <http://www.rfc-editor.org/info/rfc7012>.

[RFC7632] Waltermire, D. and D. Harrington, "Endpoint Security Posture Assessment: Enterprise Use Cases", RFC 7632, DOI 10.17487/RFC7632, September 2015, <http://www.rfc-editor.org/info/rfc7632>.

Appendix A. Change Log

A.1. Changes in Revision 01

Added some proposed normative text.

For provenance:

Added a class "Method"

Added the produced-using relationship between an AVP and a method

Added the produced-by relationship between a Guidance and a SACM Component

Added the hosted-by relationship between a SACM Component and an Endpoint

asserted-by and summarized-by have been renamed to produced-by.

"User" is now "Account". If a user has different credentials, SACM cannot know that they belong to the same user. But, per Kim W, many organizations do have accounts that associate credentials.

The multiplicity of the based-on relationships has been corrected.

More relationships now have labels, per UML convention.

The diagram no longer has causal arrow. They had become redundant and were nonstandard and clutter.

Renamed "credential" to "identity", following industry usage. A credential includes proof, such as a key or password. A username or a distinguished name is called an "identity".

Removed Session, because an endpoint’s network activity is not SACM’s initial focus

Removed Authorization, for the same reason

Added many-to-many relationship between Hardware Component and Endpoint, for clarity

Added many-to-many relationship between Software Component and Endpoint, for clarity

Added "contains" relationship between Network Interface and Network Interface

Removed relationship between Network Interface and Account. The endpoint knows the identity it used to gain network access. The PDP also knows that. But they probably do not know the account.

Added relationship between Network Interface and Identity. The endpoint and the PDP will typically know the identity.

Made identity-to-account a many-to-one relationship.

A.2. Changes in Revision 02

Added Section Identifying Attributes.

Split the figure into Figure Model of Endpoint and Figure Information Elements.

Added Figure Information Elements Take 2, proposing a triple-store model.

Some editorial cleanup

A.3. Changes in Revision 03

Moved Appendix A.1, Appendix A.2, and Mapping to SACM Use Cases into the Appendix. Added a reference to it in Section 1

Added the Section 3.3 section. Provided notes for the type of information we need to add in this section.

Added the Section 6 section. Moved sections on Endpoint, Hardware Component, Software Component, Hardware Instance, and Software Instance there. Provided notes for the type of information we need to add in this section.

Removed the Provenance of Information Section. SACM is not going to solve provenance rather give organizations enough information to figure it out.

Updated references to the Endpoint Security Posture Assessment: Enterprise Use Cases document to reflect that it was published as an RFC.

Fixed the formatting of a few figures.

Included references to [RFC3580] where RADIUS is mentioned.

A.4. Changes in Revision 04

Integrated the IPFIX [RFC7012] syntax into Section 3.3.

Converted many of the existing SACM Information Elements to the IPFIX syntax.

Included existing IPFIX Information Elements and datatypes that could likely be reused for SACM in Section 7 and Section 3.3 respectively.

Removed the sections related to reports as described in https://github.com/sacmwg/draft-ietf-sacm-information-model/issues/30.

Cleaned up other text throughout the document.

A.5. Changes in Revision 05

Merged proposed changes from the I-D IM into the WG IM (https://github.com/sacmwg/draft-ietf-sacm-information-model/issues/41).

Fixed some formatting warnings.

Removed a duplicate IE and added a few IE datatypes that were missing.

A.6. Changes in Revision 06

Clarified that the SACM statement and content-element subjects are conceptual and that they do not need to be explicitly defined in a data model as long as the necessary information is provided.

Updated the IPFIX syntax used to define Information Elements. There are still a couple of open issues that need to be resolved.

Updated some of the Information Elements contained in Section 7 to use the revised IPFIX syntax. The rest of the Information Elements will be converted in a later revision.

Performed various clean-up and refactoring in Sections 6 and 7. Still need to go through Section 8.

Removed appendices that were not referenced in the body of the draft. The text from them is still available in previous revisions of this document if needed.

A.7. Changes in Revision 07

Made various changes to the IPFIX syntax based on discussions at the IETF 96 Meeting. Changes included the addition of a structure property to the IE specification template, the creation of an enumeration datatype, and the specification of an IE naming convention.

Provided text to define Collection Guidance, Evaluation Guidance, Classification Guidance, Storage Guidance, and Evaluation Results.

Included additional IEs related to software, configuration, and the vulnerability assessment scenario.

Added text for the IANA considerations, security considerations, operational considerations, and privacy considerations sections.

Performed various other editorial changes and clean-up.

Authors' Addresses

David Waltermire (editor)
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Email: david.waltermire@nist.gov

Kim Watson
United States Department of Homeland Security
DHS/CS&C/FNR
245 Murray Ln. SW, Bldg 410
MS0613
Washington, DC 20528
USA

Email: kimberly.watson@hq.dhs.gov

Clifford Kahn
Pulse Secure, LLC
2700 Zanker Road, Suite 200
San Jose, CA 95134
USA

Email: cliffordk@pulsesecure.net

Lisa Lorenzin
Pulse Secure, LLC
2700 Zanker Road, Suite 200
San Jose, CA 95134
USA

Email: llorenzin@pulsesecure.net

Michael Cokus
The MITRE Corporation
903 Enterprise Parkway, Suite 200
Hampton, VA 23666
USA

Email: msc@mitre.org

Daniel Haynes
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: dhaynes@mitre.org

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de