

SACM
Internet-Draft
Intended status: Informational
Expires: December 10, 2014

N. Cam-Winget
Cisco Systems
June 8, 2014

Secure Automation and Continuous Monitoring (SACM) Requirements
draft-camwinget-sacm-requirements-04

Abstract

This document defines the scope and set of requirements for the Secure Automation and Continuous Monitoring working group. The requirements and scope are based on the agreed upon use cases.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 10, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements	2
2.1. General SACM requirements	2
2.2. Requirements based on Use Cases	4
3. Acknowledgements	5
4. IANA Considerations	5
5. Security Considerations	5
6. References	5
6.1. Normative References	5
6.2. Informative References	6
Author's Address	6

1. Introduction

Today's challenges of evolving threats and improved analytics to address such threats highlight a need to automate the securing of both information and the systems that store, process and transmit the information. SACM's charter focuses on addressing some of these challenges in a narrower scope by bounding the task to address use cases that pertain to the posture assessment of endpoints.

This document focuses on describing the requirements for facilitating the exchange of posture assessment information, in particular, for the use cases as exemplified in [I-D.ietf-sacm-use-cases]. Also, this document uses terminology defined in [I-D.ietf-sacm-terminology].

2. Requirements

This document defines requirements based on the SACM use cases defined in [I-D.ietf-sacm-use-cases]. This section describes the requirements used by SACM to assess and compare candidate information models and protocols to suit the architecture. These requirements express characteristics or features that a candidate protocol or data model must be capable of offering so as to ensure security and interoperability.

2.1. General SACM requirements

The use cases defined in [I-D.ietf-sacm-use-cases] apply to many deployment scenarios. To ensure interoperability, scalability and flexibility in any of these deployments, the following requirements are defined for all use cases:

G-001 Extensibility: the data models, protocols and transports defined by SACM must be extensible to allow support for non-standard and future extensions. The transport protocol must support easily

adding new operations while maintaining backwards compatibility. The query language must allow general inquiries as well as expression of specific paths to follow; retrieval of specific information based on an event, as well as on a continuous basis; and the ability to retrieve specific pieces of information, specific classes of information, and/or the entirety of available information. The information model must accommodate the addition of new data types and/or schemas in a backwards compatible fashion.

G-002 Interoperability: The data models, protocols and transports must be specified with enough details and state machine to ensure interoperability.

G-003 Scalability: The data models, protocols and transports must be scalable. SACM must support a broad set of deployment scenarios. As such, it is possible that the size or posture assessment information can vary from a single assessment that is small in (record or datagram) size to a very large datagram or a very large set of assessments and must be addressed by the SACM specifications defined.

G-004 Agility: The agility requirement is to ensure that the data model, protocols, transports and its implementations are suitable to fit in different deployment models and scenarios. Considerations for the lightweight implementations of data models and transports is required. Use cases, especially in the vulnerability assessment and threat defense applications require time criticality in both obtaining the information as well as consuming (e.g. parsing) the data.

G-005 Transport variability: Different transports must be supported to address different deployment and time constraints. Supporting transports at the Layer 2, Layer 3 and higher application layers.

G-006 Extensibility: a method for expressing both standard and non-standard (implementer-specific) data attributes while avoiding collisions should be defined. For interoperability and scope boundary, an explicit set of data attributes as mandatory to implement should be defined and focused on Posture Assessment should be described to allow for interoperability too.

G-007 Access Control: To address security and privacy considerations, the data model, protocols and transport must consider authorization based on roles to only allow authorized requestors and publishers to access the information being requested or published.

2.2. Requirements based on Use Cases

This section describes the requirements that may apply to information models, data models, protocols or transports as identified by the use cases in [I-D.ietf-sacm-use-cases] and referenced by the section numbers from that draft.

REQ-001 Attribute Dictionary: Use Cases in the whole of Section 2 describe the need for an Attribute Dictionary. With SACM's scope focused on Posture Assessment, the attribute collection and aggregation must have a well understood set of attributes inclusive of their meaning or usage intent.

REQ-002 Information Model: Use Case 2.1.1 describes the need for an Information Model to drive content definition. As SACM endeavors to reuse already existing standards which may have their own data models defined by instantiating an information model, the data models can be mapped to SACM's information model. See [RFC3444] for a description and distinctions between an information and data model.

REQ-003 Data Model to Protocol mapping: Use Case 2.1.1 describes the need to instantiate a data model that can map to the SACM protocols for posture content operations such as publication, query, change detection and asynchronous notifications.

REQ-004 Endpoint Discovery: Use Case 2.1.2 describes the need to discover endpoints and their composition.

REQ-005 Attribute based query: Use Case 2.1.2 describes the need for the data model to support a query operation based on a set of attributes to facilitate collection of information such as posture assessment, inventory (of endpoints or endpoint components) and configuration checklist. .

REQ-006 Information based query with filtering: Use Case 2.1.3 describes the need for the data model to support the means for the information to be collected through a query mechanism. Furthermore, the query operation requires filtering capabilities to allow for only a subset of information to be retrieved. The query operation may be a synchronous request or asynchronous request.

REQ-007 Asynchronous publication, updates or change modifications with filtering: Use Cases 2.1.3, 2.1.4 and 2.1.5 describe the need for the data model to support the means for the information to be published asynchronously. Similarly, the data model must support the means for a requestor to obtain updates or change modifications asynchronously. Like the query operation, these update

notifications can be set up with a filter to allow for only a subset of posture assessment information to be obtained.

REQ-008 Data model scalability: Use Cases 2.1.4 and 2.1.5 describes the need for the data model to support scalability. For example, the query operation may result in a very large set of attributes as well as a large set of targets.

REQ-009 Separation of Collection Request and Collection Action: the data model must distinguish the means to request for a data item to include enough information to properly identify the item to collect but the request could be separate and distinct from the actual method or process used to fulfill the request.

3. Acknowledgements

The authors would like to thank Barbara Fraser, Jim Bieda and Adam Montville for reviewing and contributing to this draft.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

This document defines the requirements for SACM. As such, it is expected that several data models, protocols and transports may be defined or reused from already existing standards. This section will highlight security considerations that may apply to SACM based on the architecture and standards applied in SACM.

6. References

6.1. Normative References

[I-D.ietf-sacm-terminology]

Waltermire, D., Montville, A., Harrington, D., and N. Cam-Winget, "Terminology for Security Assessment", draft-ietf-sacm-terminology-04 (work in progress), May 2014.

[I-D.ietf-sacm-use-cases]

Waltermire, D. and D. Harrington, "Endpoint Security Posture Assessment - Enterprise Use Cases", draft-ietf-sacm-use-cases-07 (work in progress), April 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.
- [RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, June 2008.

Author's Address

Nancy Cam-Winget
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
US

Email: ncamwing@cisco.com

SACM Working Group
Internet-Draft
Intended status: Informational
Expires: June 17, 2019

H. Birkholz
Fraunhofer SIT
J. Lu
Oracle Corporation
J. Strassner
Huawei Technologies
N. Cam-Winget
Cisco Systems
A. Montville
CIS
December 14, 2018

Security Automation and Continuous Monitoring (SACM) Terminology
draft-ietf-sacm-terminology-16

Abstract

This memo documents terminology used in the documents produced by SACM (Security Automation and Continuous Monitoring).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 17, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terms and Definitions	2
3. IANA Considerations	21
4. Security Considerations	21
5. Acknowledgements	22
6. Change Log	22
7. Contributors	26
8. References	27
8.1. Normative References	28
8.2. Informative References	28
Appendix A. The Attic	29
Authors' Addresses	29

1. Introduction

Our goal with this document is to improve our agreement on the terminology used in documents produced by the IETF Working Group for Security Automation and Continuous Monitoring. Agreeing on terminology should help reach consensus on which problems we're trying to solve, and propose solutions and decide which ones to use.

2. Terms and Definitions

This section describes terms that have been defined by other RFC's and defines new ones. The predefined terms will reference the RFC and where appropriate will be annotated with the specific context by which the term is used in SACM. Note that explanatory or informational augmentation to definitions are segregated from the definitions themselves. The definition for the term immediately follows the term on the same line, whereas expository text is contained in subsequent paragraphs immediately following the definition.

Assertion: Defined by the ITU in [X.1252] as "a statement made by an entity without accompanying evidence of its validity".

In the context of SACM, an assertion is the output of a SACM Component in the form of a SACM Statement (including metadata about the data source and data origin, e.g. timestamps). While the validity of an assertion about Content and Content Metadata cannot be verified without, for example, Integrity Proofing of the

Data Source, an assertion (and therefore a SACM statement, respectively) of the validity of Statement Metadata can be enabled by including corresponding Integrity Evidence created by the Data Origin.

Assessment: Defined in [RFC5209] as "the process of collecting posture for a set of capabilities on the endpoint (e.g., host-based firewall) such that the appropriate validators may evaluate the posture against compliance policy."

Attribute: Is a data element, as defined in [RFC5209], that is atomic.

In the context of SACM, attributes are "atomic" information elements and an equivalent to attribute-value-pairs. Attributes can be components of Subjects, the basic composite definitions that are defined in the SACM Information Model.

Capability: A set of features that are available from a SACM Component.

See also "capability" in [I-D.ietf-i2nsf-terminology].

In the context of SACM, the extent of a SACM component's ability is enabled by the functions it is composed of. Capabilities are registered at a SACM broker (potentially also at a proxy or a repository component if it includes broker functions) by a SACM component via the SACM component registration task and can be discovered by or negotiated with other SACM components via the corresponding tasks. For example, the capability of a SACM provider may be to provide target endpoint records (declarative guidance about well-known or potential target endpoints), or only a subset of that data.

A capability's description is in itself imperative guidance on what functions are exposed to other SACM components in a SACM domain and how to use them in workflows.

The SACM Vulnerability Assessment Scenario [I-D.ietf-sacm-vuln-scenario] defines the terms Endpoint Management Capabilities, Vulnerability Management Capabilities, and Vulnerability Assessment Capabilities, which illustrate specific sets of SACM capabilities on an enterprise IT department's point of view and therefore compose sets of declarative guidance.

Collection Result: Is a composition of one or more content elements carrying information about a target endpoint, that is produced by a collector when conducting a collection task.

Collection Task: A targeted task that collects attributes and/or corresponding attribute values from target endpoint.

There are four types of frequency collection tasks can be conducted with:

ad-hoc, e.g. triggered by a unsolicited query

conditional, e.g. triggered in accordance with policies included in the compositions of workflows

scheduled, e.g. in regular intervals, such as every minute or weekly

continuously, e.g. a network behavior observation

There are three types of collection methods, each requiring an appropriate set of functions to be included in the SACM component conducting the collection task:

Self-Reporting: A SACM component located on the target endpoint itself conducts the collection task.

Remote-Acquisition: A SACM component located on an Endpoint different from the target endpoint conducts the collection task via interfaces available on the target endpoint, e.g. SNMP/NETCONF or WMI.

Behavior-Observation: A SACM component located on an Endpoint different from the target endpoint observes network traffic related to the target endpoint and conducts the collection task via interpretation of that network traffic.

Collector: A piece of software that acquires information about one or more target endpoints by conducting collection tasks.

A collector can be distributed across multiple endpoints, e.g. across a target endpoint and a SACM component. The separate parts of the collector can communicate with a specialized protocol, such as PA-TNC [RFC5792]. At least one part of a distributed collector has to take on the role of a provider of information by providing SACM interfaces to propagate capabilities and to provide SACM content in the form of collection results.

Configuration: A non-volatile subset of the endpoint attributes of a endpoint that is intended to be unaffected by a normal reboot-cycle.

Configuration is a type of imperative guidance that is stored in files (files dedicated to contain configuration and/ or files that are software components), directly on block devices, or on specific hardware components that can be accessed via corresponding software components. Modification of configuration can be conducted manually or automatically via management (plane) interfaces that support management protocols, such as SNMP or WMI. A change of configuration can occur during both run-time and down-time of an endpoint. It is common practice to schedule a change of configuration during or directly after the completion of a boot-cycle via corresponding software components located on the target endpoint itself.

Examples: The static association of an IP address and a MAC address in a DHCP server configuration, a directory-path that identifies a log-file directory, a registry entry.

Configuration Drift: The disposition of endpoint characteristics to change over time.

Configuration drift exists for both hardware components and software components. Typically, the frequency and scale of configuration drift of software components is significantly higher than the configuration drift of hardware components.

Consumer: A SACM Role that requires a SACM Component to include SACM Functions enabling it to receive information from other SACM Components.

Content Element: Content elements constitute the payload data (SACM content) transferred via statement Subjects emitted by providers of information. Every content element Subject includes a specific content Subject and a corresponding content metadata Subject.

Content Metadata: Data about content Subjects. Every content-element includes a content metadata Subject. The Subject can include any information element that can annotate the content transferred. Examples include time stamps or data provenance Subjects.

Control Plane: An architectural component that provides common control functions to all SACM components.

Typically used as a term in the context of routing, e.g. [RFC6192]. SACM components may include authentication, authorization, (capability) discovery or negotiation, registration and subscription. The control plane orchestrates the flow on the data plane according to imperative guidance (i.e. configuration) received via the management plane. SACM components with interfaces to the control plane have knowledge of the capabilities of other SACM components within a SACM domain.

Controller: A controller is a SACM Role that is assigned to a SACM component containing control plane functions managing and facilitating information sharing or execute on security functions.

There are three types of SACM controllers: Broker, Proxy, and Repository. Depending on its type, a controller can also contain functions that have interfaces on the data plane.

Data Confidentiality: Defined in [RFC4949] as "the property that data is not disclosed to system entities unless they have been authorized to know the data."

Data In Motion: Data that is being transported via a network; also referred to as "Data in Transit" or "Data in Flight".

Data in motion requires a data model to transfer the data using a specific encoding. Typically, data in motion is serialized (marshalling) into a transport encoding by a provider of information and deserialized (unmarshalling) by a consumer of information. The termination points of provider of information and consumer of information data is transferred between are interfaces. In regard to data in motion, the interpretation of the roles consumer of information and provider of information depends on the corresponding OSI layer (e.g. on layer2: between interfaces connected to a broadcast domain, on layer4: between interfaces that maintain a TCP connection). In the context of SACM, consumer of information and provider of information are SACM components.

Data At Rest: Data that is stored.

Data at rest requires a data model to encode the data to be stored. In the context of SACM, data at rest located on a SACM component can be provided to other SACM components via discoverable capabilities.

Data Integrity: Defined in [RFC4949] as "the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner."

Data Origin: The SACM Component that initially acquired or produced data about an endpoint.

Data Origin enables a SACM component to identify the SACM component that initially acquired or produced data about a (target) endpoint (e.g. via collection from a data source) and made it available to a SACM domain via a SACM statement. Data Origin can be expressed by an endpoint label information element (e.g. to be used as metadata in statement).

Data Plane: Is an architectural component providing operational functions enabling information exchange that is not command and control or management related.

Typically used as a term in the context of routing (and used as a synonym for forwarding plane, e.g. [RFC6192]). In the context of SACM, the data plane is an architectural component providing operational functions to enable a SACM component to provide and consume SACM statements and therefore SACM content, which composes the actual SACM content. The data plane in a SACM domain is used to conduct distributed SACM tasks by transporting SACM content via specific transport encodings and corresponding operations defined by SACM data models.

Data Provenance: An historical record of the sources, origins and evolution, as it pertains to data, that is influenced by inputs, entities, functions and processes.

Additional Information - In the context of SACM, data provenance is expressed as metadata that identifies SACM statements and corresponding content elements a new statement is created from. In a downstream process, this references can cascade, creating a data provenance tree that enables SACM components to trace back the original data sources involved in the creation of SACM statements and take into account their characteristics and trustworthiness.

Data Source: Is an endpoint from which a particular set of attributes and/or attribute values have been collected.

Data Source enables a SACM component to identify - and potentially characterize - a (target) endpoint that is claimed to be the original source of endpoint attributes in a SACM statement. Data Source can be expressed as metadata by an endpoint label information element or a corresponding subject of identifying endpoint attributes.

Endpoint: Defined in [RFC5209] as "any computing device that can be connected to a network."

Additional Information - The [RFC5209] definition continues, "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 the [RFC5209] definition, 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).

Based on the definition of an asset, an endpoint is a type of asset.

Endpoint Attribute: Is a discreet endpoint characteristic that is computably observable.

Endpoint Attributes typically constitute Attributes that can be bundled into Subject (e.g. information about a specific network interface can be represented via a set of multiple AVP).

Endpoint Characteristics: The state, configuration and composition of the software components and (virtual) hardware components a target endpoint is composed of, including observable behavior, e.g. sys-calls, log-files, or PDU emission on a network.

In SACM work-flows, (Target) Endpoint Characteristics are represented via Information Elements.

Endpoint Characterization Task: The task of endpoint characterization that uses endpoint attributes that represent distinct endpoint characteristics.

Endpoint Classification: The categorization of of the endpoint into one or more taxonomic structures.

Endpoint classification requires declarative guidance in the form of an endpoint profile, discovery results and potentially collection results. Types, classes or the characteristics of an individual target endpoint are defined via endpoint profiles.

Endpoint Classification Task: The task of endpoint classification that uses an endpoint's characteristics to determine how to categorize the given endpoint into one or more taxonomic structures.

Endpoint Label: A unique label associated with a unique endpoint.

Endpoint specializations have corresponding endpoint label specializations. For example, an endpoint label used on a SACM Component is a SACM Component Label.

Endpoint Management Capabilities: Enterprise IT management capabilities that are tailored to manage endpoint identity, endpoint information, and associated metadata.

Evaluation Task: A task by which an endpoint's asserted attribute value is evaluated against a policy-compliant attribute value.

Evaluation Result: The resulting value from having evaluated a set of posture attributes.

Expected Endpoint Attribute State: The policy-compliant state of an endpoint attribute that is to be compared against.

Sets of expected endpoint attribute states are transported as declarative guidance in target endpoint profiles via the management plane. This, for example, can be a policy, but also a recorded past state. An expected state is represented by an Attribute or a Subject that represents a set of multiple attribute value pairs.

Guidance: Machine-processable input directing SACM processes or tasks.

Examples of such processes/tasks include automated device management, remediation, collection, evaluation. Guidance influences the behavior of a SACM Component and is considered content of the management plane. In the context of SACM, guidance is machine-readable and 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.

There are two types of guidance:

Declarative Guidance: Guidance that defines the configuration or state an endpoint is supposed to be in, without providing specific actions or methods to produce that desired state. Examples include Target Endpoint Profiles or network topology based requirements.

Imperative Guidance: Guidance that prescribes specific actions to be conducted or methods to be used in order to achieve an outcome. Examples include a targeted Collection Task or the IP-Address of a SACM Component that provides a registration function.

Prominent examples include: modification of the configuration of a SACM component or updating a target endpoint profile that resides on an evaluator. In essence, guidance is transported via the management plane.

Endpoint Hardware Inventory: The set of hardware components that compose a specific endpoint representing its hardware configuration.

Hardware Component: A distinguishable physical component used to 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. FRU in a chassis are connected via redundant busses). In general, a hardware component can be distinguished by its serial number. Occasionally, hardware components are referred to as power sucking aliens.

Information Element: A representation of information about physical and virtual "objects of interest".

Information elements are the building blocks that constitute the SACM information model. In the context of SACM, an information element that expresses a single value with a specific name is referred to as an Attribute (analogous to an attribute-value-pair). A set of attributes that is bundled into a more complex composite information element is referred to as a Subject. Every

information element in the SACM information model has a unique name. Endpoint attributes or time stamps, for example, are represented as information elements in the SACM information model.

Information Model: An abstract representation of data, their properties, relationships between data and the operations that can be performed on the data.

While there is some overlap with a data model, [RFC3444] distinguishes an information model as being protocol and implementation neutral whereas a data model would provide such details. The purpose of the SACM information model is to ensure interoperability between SACM data models (that are used as transport encoding) and to provide a standardized set of information elements for communication between SACM components.

Interaction Model: The definition of specific sequences regarding the exchange of messages (data in motion), including, for example, conditional branching, thresholds and timers.

An interaction model, for example, can be used to define operations, such as registration or discovery, on the control plane. A composition of data models for data in motion and a corresponding interaction model is a protocol.

Internal Collector: A collector that runs on a target endpoint to acquire information from that target endpoint.

Management Plane: An architectural component providing common functions to steer the behavior of SACM components, e.g. their behavior on the control plane.

Typically, a SACM component can fulfill its purpose without continuous input from the management plane. In contrast, without continuous availability of control plane functions a typical SACM component could not function properly. In general, interaction on the management plane is less frequent and less regular than on the control plane. Input via the management plane can be manual (e.g. via a CLI), or can be automated via management plane functions that are part of other SACM components.

Network Address: A layer-specific address that follows a layer-specific address scheme.

The following characteristics are a summary derived from the Common Information Model and ITU-T X.213. Each Network Interface of a specific layer can be associated with one or more addresses appropriate for that layer. There is no guarantee that a network

address is globally unique. A dedicated authority entity can provide a level of assurance that a network address is unique in its given scope. In essence, there is always a scope to a network address, in which it is intended to be unique.

Examples include: physical Ethernet port with a MAC address, layer 2 VLAN interface with a MAC address, layer 3 interface with multiple IPv6 addresses, layer 3 tunnel ingress or egress with an IPv4 address.

Network Interface: An Endpoint is connected to a network via one or more Network Interfaces. Network Interfaces can be physical (Hardware Component) or logical (virtual Hardware component, i.e. a dedicated Software Component). Network Interfaces of an Endpoint can operate on different layers, most prominently what is now commonly called layer 2 and 3. Within a layer, interfaces can be nested.

In SACM, the association of Endpoints and Network Addresses via Network Interfaces is vital to maintain interdependent autonomous processes that can be targeted at Target Endpoints, unambiguously.

Examples include: physical Ethernet port, layer 2 VLAN interface, a MC-LAG setup, layer 3 Point-to-Point tunnel ingress or egress.

Metadata: Data about data.

In the SACM information model, data is referred to as Content. Metadata about the content is referred to as Content-Metadata, respectively. Content and Content-Metadata are combined into Subjects called Content-Elements in the SACM information model. Some information elements defined by the SACM information model can be part of the Content or the Content-Metadata. Therefore, if an information element is considered data or data about data depends on which kind of Subject it is associated with. The SACM information model also defines metadata about the data origin via the Subject Statement-Metadata. Typical examples of metadata are time stamps, data origin or data source.

Posture: Defined in [RFC5209] as "configuration and/or status of hardware or software on an endpoint as it pertains to an organization's security policy."

This term is used within the scope of SACM to represent the configuration and state information that is collected from a target endpoint in the form of endpoint attributes (e.g. software/hardware inventory, configuration settings, dynamically assigned

addresses). This information may constitute one or more posture attributes.

Posture Attributes: Defined in [RFC5209] as "attributes describing the configuration or status (posture) of a feature of the endpoint. A Posture Attribute represents a single property of an observed state. For example, a Posture Attribute might describe the version of the operating system installed on the system."

Within this document this term represents a specific assertion about endpoint configuration or state (e.g. configuration setting, installed software, hardware) represented via endpoint attributes. The phrase "features of the endpoint" highlighted above refers to installed software or software components.

Provider: A provider is a SACM role assigned to a SACM component that provides role-specific functions to provide information to other SACM components.

Repository: A repository is a controller that contains functions to consume, store and provide information of a particular kind.

Such information is typically data transported on the data plane, but potentially also data and metadata from the control and management plane. A single repository may provide the functions of more than one specific repository type (i.e. configuration baseline repository, assessment results repository, etc.)

SACM Broker Controller: A SACM Broker Controller is a controller that contains control plane functions to provide and/or connect services on behalf of other SACM components via interfaces on the control plane.

A broker may provide, for example, authorization services and find, upon request, SACM components providing requested services.

SACM Component: Is a component, as defined in [I-D.ietf-i2nsf-terminology], that is composed of SACM capabilities.

In the context of SACM, a set of SACM functions composes a SACM component. A SACM component conducts SACM tasks, acting on control plane, data plane and/or management plane via corresponding SACM interfaces. SACM defines a set of standard components (e.g. a collector, a broker, or a data store). A SACM component contains at least a basic set of control plane functions and can contain data plane and management plane functions. A SACM component residing on an endpoint assigns one or more SACM roles

to the corresponding endpoint due to the SACM functions it is composed of. A SACM component "resides on" an endpoint and an endpoint "contains" a SACM component, correspondingly. For example, a SACM component that is composed solely of functions that provide information would only take on the role of a provider.

SACM Component Discovery: The task of discovering the capabilities provided by SACM components within a SACM domain.

This is likely to be performed via an appropriate set of control plane functions.

SACM Component Label: A specific endpoint label that is used to identify a SACM component.

In content-metadata, this label is called data origin.

SACM Content: The payload provided by SACM components to the SACM domain on the data plane.

SACM content includes the SACM data models.

SACM Domain: Endpoints that include a SACM component compose a SACM domain.

(To be revised, additional definition content TBD, possible dependencies to SACM architecture)

SACM Function: A behavioral aspect of a SACM component that provides external SACM Interfaces or internal interfaces to other SACM Functionse.

For example, a SACM Function with SACM Interfaces on the Control Plane can provide a brokering function to other SACM Components. Via Data Plane interfaces, a SACM Function can act as a provider and/or as a consumer of information. SACM Functions can be propagated as the Capabilities of a SACM Component and can be discovered by or negotiated with other SACM Components.

SACM Interface: An interface, as defined in [I-D.ietf-i2nsf-terminology], that provides SACM-specific operations.

[I-D.ietf-i2nsf-terminology] defines interface as a "set of operations one object knows it can invoke on, and expose to, another object," and further defines interface by stating that an interface "decouples the implementation of the operation from its

specification. An interface is a subset of all operations that a given object implements. The same object may have multiple types of interfaces to serve different purposes."

In the context of SACM, SACM Functions provide SACM Interfaces on the management, control, or data plane. Operations a SACM Interface provides are based on corresponding data model defined by SACM. SACM Interfaces are used for communication between SACM components.

SACM Proxy Controller: A SACM Proxy Controller is a controller that provides data plane and control plane functions, information, or services on behalf of another component, which is not directly participating in the SACM architecture.

SACM Role: Is a role, as defined in [I-D.ietf-i2nsf-terminology], that requires the SACM Component assuming the role to bear a set of SACM functions or interfaces.

SACM Roles provide three important benefits. First, it enables different behavior to be supported by the same Component for different contexts. Second, it enables the behavior of a Component to be adjusted dynamically (i.e., at runtime, in response) to changes in context, by using one or more Roles to define the behavior desired for each context. Third, it decouples the Roles of a Component from the Applications that use that Component."

In the context of SACM, SACM roles are associated with SACM components and are defined by the set of functions and interfaces a SACM component includes. There are three SACM roles: provider, consumer, and controller. The roles associated with a SACM component are determined by the purpose of the SACM functions and corresponding SACM interfaces the SACM component is composed of.

SACM Statement: Is an assertion that is made by a SACM Component.

Security Automation: The process of which security alerts can be automated through the use of different components to monitor, analyze and assess endpoints and network traffic for the purposes of detecting misconfigurations, misbehaviors or threats.

Security Automation is intended to identify target endpoints that cannot be trusted (see "trusted" in [RFC4949]). This goal is achieved by creating and processing evidence (assessment statements) that a target endpoint is not a trusted system [RFC4949].

Software Package: A generic software package (e.g. a text editor).

Software Component: A software package installed on an endpoint.

The software component may include a unique serial number (e.g. a text editor associated with a unique license key).

Software Instance: A running instance of a software component.

For example, 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.

State: A volatile set of endpoint attributes of a (target) endpoint that is affected by a reboot-cycle.

Local state is created by the interaction of components with other components via the control plane, via processing data plane payload, or via the functional properties of local hardware and software components. Dynamic configuration (e.g. IP address distributed dynamically via an address distribution and management services, such as DHCP) is considered state that is the result of the interaction with another component (e.g. provided by a DHCP server with a specific configuration).

Examples: The static association of an IP address and a MAC address in a DHCP server configuration, a directory-path that identifies a log-file directory, a registry entry.

Statement: A statement is the root/top-level subject defined in the SACM information model.

A statement is used to bundle Content Elements into one subject and includes metadata about the data origin.

Subject: A semantic composite information element pertaining to a system entity that is a target endpoint.

Like Attributes, subjects have a 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 (e.g. zero-one, none-unbounded). The content IE of a subject can be an unordered or an ordered list.

In contrast to the definitions of subject provided by [RFC4949], a subject in the scope of SACM is neither "a system entity that

causes information to flow among objects or changes the system state" nor "a name of a system entity that is bound to the data items in a digital certificate".

In the context of SACM, a subject is a semantic composite of information elements about a system entity that is a target endpoint. Every acquirable subject-as defined in the scope of SACM-about a target endpoint represents and therefore identifies every subject-as defined by [RFC4949]-that is a component of that target endpoint. The semantic difference between both definitions can be subtle in practice and is in consequence important to highlight.

Supplicant: A component seeking to be authenticated via the control plane for the purpose of participating in a SACM domain.

System Resource: Defined in [RFC4949] as "data contained in an information system; or a service provided by a system; or a system capacity, such as processing power or communication bandwidth; or an item of system equipment (i.e., hardware, firmware, software, or documentation); or a facility that houses system operations and equipment."

Target Endpoint: Is an endpoint that is under assessment at some point in, or region of, time.

Every endpoint that is not specifically designated as an excluded endpoint is a target endpoint. A target endpoint is not part of a SACM domain unless it contains a SACM component (e.g. a SACM component that publishes collection results coming from an internal collector).

A target endpoint is similar to a device that is a Target of Evaluation (TOE) as defined in Common Criteria and as referenced by {{RFC4949}}.

Target Endpoint Address: An address that is layer specific and which follows layer specific address schemes.

Each interface of a specific layer can be associated with one or more addresses appropriate for that layer. There is no guarantee that an address is globally unique. In general, there is a scope to an address in which it is intended to be unique.

Examples include: physical Ethernet port with a MAC address, layer 2 VLAN interface with a MAC address, layer 3 interface with multiple IPv6 addresses, layer 3 tunnel ingress or egress with an IPv4 address.

Target Endpoint Characterization: The description of the distinctive nature of a target endpoint, that is based on its characteristics.

Target Endpoint Characterization Record: A set of endpoint attributes about a target endpoint that was encountered in a SACM domain, which are associated with that target endpoint as a result of a Target Endpoint Characterization Task.

A characterization record is intended to be a representation of an endpoint. It cannot be assured that a record distinctly represents a single target endpoint unless a set of one or more endpoint attributes that compose a unique set of identifying endpoint attributes are included in the record. Otherwise, the set of identifying attributes included in a record can match more than one target endpoints, which are - in consequence - indistinguishable to a SACM domain until more qualifying endpoint attributes can be acquired and added to the record. A characterization record is maintained over time in order to assert that acquired endpoint attributes are either about an endpoint that was encountered before or an endpoint that has not been encountered before in a SACM domain. A characterization record can include, for example, acquired configuration, state or observed behavior of a specific target endpoint. Multiple and even conflicting instances of this information can be included in a characterization record by using timestamps and/or data origins to differentiate them. The endpoint attributes included in a characterization record can be used to re-identify a distinct target endpoint over time. Classes or profiles can be associated with a characterization record via the Classification Task in order to guide collection, evaluation or remediation tasks.

Target Endpoint Characterization Task: An ongoing task of continuously adding acquired endpoint attributes to a corresponding record. The TE characterization task manages the representation of encountered target endpoints in the SACM domain in the form of characterization records. For example, the output of a target endpoint discovery task or a collection task can be processed by the characterization task and added to the record. The TE characterization Task also manages these representations of target endpoints encountered in the SACM domain by splitting or merging the corresponding records as new or more refined endpoint attributes become available.

Target Endpoint Classification Task: The task of associating a class from an extensible list of classes with an endpoint characterization record. TE classes function as imperative and declarative guidance for collection, evaluation, remediation and security posture assessment in general.

Target Endpoint Discovery Task: The ongoing task of detecting previously unknown interaction of a potential target endpoint in the SACM domain. TE Discovery is not directly targeted at a specific target endpoint and therefore an un-targeted task. SACM Components conducting the discovery task as a part of their function are typically distributed and located, for example, on infrastructure components or collect from those remotely via appropriate interfaces. Examples of infrastructure components that are of interest to the discovery task include routers, switches, VM hosting or VM managing components, AAA servers, or servers handling dynamic address distribution.

Target Endpoint Identifier: The target endpoint discovery task and the collection tasks can result in a set of identifying endpoint attributes added to a corresponding Characterization Record. This subset of the endpoint attributes included in the record is used as a target endpoint identifier, by which a specific target endpoint can be referenced. Depending on the available identifying attributes, this reference can be ambiguous and is a "best-effort" mechanism. Every distinct set of identifying endpoint attributes can be associated with a target endpoint label that is unique in a SACM domain.

Target Endpoint Label: An endpoint label that identifies a specific target endpoint.

Target Endpoint Profile: A bundle of expected or desired component composition, configurations and states that is associated with a target endpoint.

The corresponding task by which the association with a target endpoint takes places is the endpoint classification task. The task by which an endpoint profile is created is the endpoint characterization task. A type or class of target endpoints can be defined via a target endpoint profile. Examples include: printers, smartphones, or an office PC.

In respect to [RFC4949], a target endpoint profile is a protection profile as defined by Common Criteria (analogous to the target endpoint being the target of evaluation).

SACM Task: Is a task conducted within the scope of a SACM domain by one or more SACM functions that achieves a SACM-defined outcome.

A SACM task can be triggered by other operations or functions (e.g. a query from another SACM component or an unsolicited push on the data plane due to an ongoing subscription). A task is part of a SACM process chain. A task starts at a given point in time

and ends in a deterministic state. With the exception of a collection task, a SACM task consumes SACM statements provided by other SACM components. The output of a task is a result that can be provided (e.g. published) on the data plane.

The following tasks are defined by SACM:

Target Endpoint Discovery

Target Endpoint Characterization

Target Endpoint Classification

Collection

Evaluation [TBD]

Information Sharing [TBD]

SACM Component Discovery

SACM Component Authentication [TBD]

SACM Component Authorization [TBD]

SACM Component Registration [TBD]

Timestamps : Defined in [RFC4949] as "with respect to a data object, a label or marking in which is recorded the time (time of day or other instant of elapsed time) at which the label or marking was affixed to the data object".

A timestamp always requires context, i.e. additional information elements that are associated with it. Therefore, all timestamps wrt information elements are always metadata. Timestamps in SACM Content Elements may be generated outside a SACM Domain and may be encoded in an unknown representation. Inside a SACM domain the representation of timestamps is well-defined and unambiguous.

Virtual Endpoint: An endpoint composed entirely of logical system components (see [RFC4949]).

The most common example is a virtual machine/host running on a target endpoint. Effectively, target endpoints can be nested and at the time of this writing the most common example of target endpoint characteristics about virtual components is the EntLogicalEntry in [RFC6933].

Vulnerability Assessment: An assessment specifically tailored to determining whether a set of endpoints is vulnerable according to the information contained in the vulnerability description information.

Vulnerability Description Information: Information pertaining to the existence of a flaw or flaws in software, hardware, and/or firmware, which could potentially have an adverse impact on enterprise IT functionality and/or security.

Vulnerability description information should contain enough information to support vulnerability detection.

Vulnerability Detection Data: A type of imperative guidance extracted or derived from vulnerability description information that describes the specific mechanisms of vulnerability detection that is used by an enterprise's vulnerability management capabilities to determine if a vulnerability is present on an endpoint.

Vulnerability Management Capabilities: An IT management capability tailored toward managing endpoint vulnerabilities and associated metadata on an ongoing basis by ingesting vulnerability description information and vulnerability detection data, and performing vulnerability assessments.

Vulnerability assessment capabilities: An assessment capability that is tailored toward determining whether a set of endpoints is vulnerable according to vulnerability description information.

Workflow: A workflow is a modular composition of tasks that can contain loops, conditionals, multiple starting points and multiple endpoints.

The most prominent workflow in SACM is the assessment workflow.

3. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

This memo documents terminology for security automation. While it is about security, it does not affect security.

5. Acknowledgements

6. Change Log

Changes from version 00 to version 01:

- o Added simple list of terms extracted from UC draft -05. It is expected that comments will be received on this list of terms as to whether they should be kept in this document. Those that are kept will be appropriately defined or cited.

Changes from version 01 to version 02:

- o Added Vulnerability, Vulnerability Management, xposure, Misconfiguration, and Software flaw.

Changes from version 02 to version 03:

- o Removed Section 2.1. Cleaned up some editing nits; broke terms into 2 sections (predefined and newly defined terms). Added some of the relevant terms per the proposed list discussed in the IETF 89 meeting.

Changes from version 03 to version 04:

- o TODO

Changes from version 04 to version 05:

- o TODO

Changes from version 05 to version 06:

- o Updated author information.
- o Combined "Pre-defined Terms" with "New Terms and Definitions".
- o Removed "Requirements language".
- o Removed unused reference to use case draft; resulted in removal of normative references.
- o Removed introductory text from Section 1 indicating that this document is intended to be temporary.
- o Added placeholders for missing change log entries.

Changes from version 06 to version 07:

- o Added Contributors section.
- o Updated author list.
- o Changed title from "Terminology for Security Assessment" to "Secure Automation and Continuous Monitoring (SACM) Terminology".
- o Changed abbrev from "SACM-Terms" to "SACM Terminology".
- o Added appendix The Attic to stash terms for future updates.
- o Added Authentication, Authorization, Data Confidentiality, Data Integrity, Data Origin, Data Provenance, SACM Component, SACM Component Discovery, Target Endpoint Discovery.
- o Major updates to Building Block, Function, SACM Role, Target Endpoint.
- o Minor updates to Broker, Capability, Collection Task, Evaluation Task, Posture.
- o Relabeled Role to SACM Role, Endpoint Target to Target Endpoint, Endpoint Discovery to Endpoint Identification.
- o Moved Asset Targeting, Client, Endpoint Identification to The Attic.
- o Endpoint Attributes added as a TODO.
- o Changed the structure of the Change Log.

Changes from version 07 to version 08:

- o Added Assertion, Collection Result, Collector, Excluded Endpoint, Internal Collector, Network Address, Network Interface, SACM Domain, Statement, Target Endpoint Identifier, Target Endpoint Label, Timestamp.
- o Major updates to Attributes, Broker, Collection Task, Consumer, Controller, Control Plane, Endpoint Attributes, Expected Endpoint State, SACM Function, Provider, Proxy, Repository, SACM Role, Target Endpoint.
- o Minor updates to Asset, Building Block, Data Origin, Data Source, Data Provenance, Endpoint, Management Plane, Posture, Posture Attribute, SACM Component, SACM Component Discovery, Target Endpoint Discovery.

- o Relabeled Function to SACM Function.

Changes from version 08 to version 09:

- o Updated author list.
- o Added Data Plane, Endpoint Characterization, Endpoint Classification, Guidance, Interaction Model, Software Component, Software Instance, Software Package, Statement, Target Endpoint Profile, SACM Task.
- o Removed Building Block.
- o Major updates to Control Plane, Endpoint Attribute, Expected Endpoint State, Information Model, Management Plane.
- o Minor updates to Attribute, Capabilities, SACM Function, SACM Component, Collection Task.
- o Moved Asset Characterization to The Attic.

Changes from version 09 to version 10:

- o Added Configuration Drift, Data in Motion, Data at Rest, Endpoint Management Capability, Hardware Component, Hardware Inventory, Hardware Type, SACM Interface, Target Endpoint Characterization Record, Target Endpoint Characterization Task, Target Endpoint Classification Task, Target Endpoint Discovery Task, Vulnerability Description Information, Vulnerability Detection Data, Vulnerability Management Capability, Vulnerability Assessment
- o Added references to i2nsf definitions in Capability, SACM Component, SACM Interface, SACM Role.
- o Added i2nsf Terminology I-D Reference.
- o Major Updates to Endpoint, SACM Task, Target Endpoint Identifier.
- o Minor Updates to Guidance, SACM Component Discovery, Target Endpoint Label, Target Endpoint Profile.
- o Relabeled SACM Task
- o Removed Target Endpoint Discovery

Changes from version 10 to version 11:

- o Added Content Element, Content Metadata, Endpoint Label, Information Element, Metadata, SACM Component Label, Workflow.
- o Major Updates to Assessment, Capability, Collector, Endpoint Management Capabilities, Guidance, Vulnerability Assessment Capabilities, Vulnerability Detection Data, Vulnerability Assessment Capabilities.
- o Minor updates to Collection Result, Control Plane, Data in Motion, Data at Rest, Data Origin, Network Interface, Statement, Target Endpoint Label.
- o Relabeled Endpoint Management Capability, Vulnerability Management Capability, Vulnerability Assessment.

Changes from version 11 to version 12:

- o Added Configuration, Endpoint Characteristic, Event, SACM Content, State, Subject.
- o Major Updates to Assertion, Data in Motion, Data Provenance, Data Source, Interaction Model.
- o Minor Updates to Attribute, Control Plane, Data Origin, Data Provenance, Expected Endpoint State, Guidance, Target Endpoint Classification Task, Vulnerability Detection Data.

Changes from version 12 to version 13:

- o Added Virtual Component.
- o Major Updates to Capability, Collection Task, Hardware Component, Hardware Type, Security Automation, Subject, Target Endpoint, Target Endpoint Profile.
- o Minor Updates to Assertion, Data Plane, Endpoint Characteristics.

Changes from version 13 to version 14:

- o Handled a plethora of issues listed in GitHub.
- o Pruned some commonly understood terms.
- o Narrowing term labels per their definitions.
- o In some cases, excised expositional text.

- o Where expository text was left intact, it has been separated from the actual definition of a term.

Changes from version 14 to version 16:

- o moved obsolete definitions into the Appendix (attic).

7. Contributors

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20877
USA

Email: david.waltermire@nist.gov

Adam W. Montville
Center for Internet Security
31 Tech Valley Drive
East Greenbush, NY 12061
USA

Email: adam.w.montville@gmail.com

David Harrington
Effective Software
50 Harding Rd
Portsmouth, NH 03801
USA

Email: ietfdbh@comcast.net

Brian Ford
Lancope
3650 Brookside Parkway, Suite 500
Alpharetta, GA 30022
USA

Email: bford@lancope.com

Merike Kaeo
Double Shot Security
3518 Fremont Avenue North, Suite 363
Seattle, WA 98103
USA

Email: merike@doubleshotsecurity.com

8. References

8.1. Normative References

- [RFC5792] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5792, DOI 10.17487/RFC5792, March 2010, <<https://www.rfc-editor.org/info/rfc5792>>.
- [RFC6933] Bierman, A., Romascanu, D., Quittek, J., and M. Chandramouli, "Entity MIB (Version 4)", RFC 6933, DOI 10.17487/RFC6933, May 2013, <<https://www.rfc-editor.org/info/rfc6933>>.

8.2. Informative References

- [I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-06 (work in progress), July 2018.
- [I-D.ietf-netmod-entity]
Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", draft-ietf-netmod-entity-08 (work in progress), January 2018.
- [I-D.ietf-sacm-vuln-scenario]
Coffin, C., Cheikes, B., Schmidt, C., Haynes, D., Fitzgerald-McKay, J., and D. Waltermire, "SACM Vulnerability Assessment Scenario", draft-ietf-sacm-vuln-scenario-02 (work in progress), September 2016.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <<https://www.rfc-editor.org/info/rfc3444>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://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, <<https://www.rfc-editor.org/info/rfc5209>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.

[X.1252] "ITU-T X.1252 (04/2010)", n.d..

Appendix A. The Attic

The following terms are stashed for now and will be updated later:

Asset: Is a system resource, as defined in [RFC4949], that may be composed of other assets.

Examples of Assets include: Endpoints, Software, Guidance, or X.509 public key certificates. An asset is not necessarily owned by an organization.

Asset Management: The IT process by which assets are provisioned, updated, maintained and deprecated.

Asset Characterization: Asset characterization is the process of defining attributes that describe properties of an identified asset.

Asset Targeting: Asset targeting is the use of asset identification and categorization information to drive human-directed, automated decision making for data collection and analysis in support of endpoint posture assessment.

Client: An architectural component receiving services from another architectural component.

Endpoint Identification (TBD per list; was "Endpoint Discovery"):
The process by which an endpoint can be identified.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Jarrett Lu
Oracle Corporation
4180 Network Circle
Santa Clara, CA 95054
USA

Email: jarrett.lu@oracle.com

John Strassner
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95138
USA

Email: john.sc.strassner@huawei.com

Nancy Cam-Winget
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
USA

Email: ncamwing@cisco.com

Adam Montville
Center for Internet Security
31 Tech Valley Drive
East Greenbush, NY 12061
USA

Email: adam.w.montville@gmail.com

Security Automation and Continuous Monitoring WG
Internet-Draft
Intended status: Informational
Expires: January 2, 2016

D. Waltermire
NIST
D. Harrington
Effective Software
July 1, 2015

Endpoint Security Posture Assessment - Enterprise Use Cases
draft-ietf-sacm-use-cases-10

Abstract

This memo documents a sampling of use cases for securely aggregating configuration and operational data and evaluating that data to determine an organization's security posture. From these operational use cases, we can derive common functional capabilities and requirements to guide development of vendor-neutral, interoperable standards for aggregating and evaluating data relevant to security posture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Endpoint Posture Assessment	4
2.1. Use Cases	5
2.1.1. Define, Publish, Query and Retrieve Security Automation Data	5
2.1.2. Endpoint Identification and Assessment Planning	9
2.1.3. Endpoint Posture Attribute Value Collection	10
2.1.4. Posture Attribute Evaluation	11
2.2. Usage Scenarios	12
2.2.1. Definition and Publication of Automatable Configuration Checklists	12
2.2.2. Automated Checklist Verification	13
2.2.3. Detection of Posture Deviations	16
2.2.4. Endpoint Information Analysis and Reporting	17
2.2.5. Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra	18
2.2.6. Identification and Retrieval of Guidance	20
2.2.7. Guidance Change Detection	21
3. IANA Considerations	21
4. Security Considerations	21
5. Acknowledgements	22
6. Change Log	22
6.1. -08- to -09-	22
6.2. -07- to -08-	22
6.3. -06- to -07-	23
6.4. -05- to -06-	23
6.5. -04- to -05-	23
6.6. -03- to -04-	24
6.7. -02- to -03-	24
6.8. -01- to -02-	25
6.9. -00- to -01-	25
6.10. draft-waltermire-sacm-use-cases-05 to draft-ietf-sacm-use-cases-00	26
6.11. waltermire -04- to -05-	27
7. Informative References	28
Authors' Addresses	28

1. Introduction

This document describes the core set of use cases for endpoint posture assessment for enterprises. It provides a discussion of

these use cases and associated building block capabilities. The described use cases support:

- o securely collecting and aggregating configuration and operational data, and
- o evaluating that data to determine the security posture of individual endpoints.

Additionally, this document describes a set of usage scenarios that provide examples for using the use cases and associated building blocks to address a variety of operational functions.

These operational use cases and related usage scenarios cross many IT security domains. The use cases enable the derivation of common:

- o concepts that are expressed as building blocks in this document,
- o characteristics to inform development of a requirements document
- o information concepts to inform development of an information model document, and
- o functional capabilities to inform development of an architecture document.

Together these ideas will be used to guide development of vendor-neutral, interoperable standards for collecting, aggregating, and evaluating data relevant to security posture.

Using this standard data, tools can analyze the state of endpoints, user activities and behaviour, and evaluate the security posture of an organization. Common expression of information should enable interoperability between tools (whether customized, commercial, or freely available), and the ability to automate portions of security processes to gain efficiency, react to new threats in a timely manner, and free up security personnel to work on more advanced problems.

The goal is to enable organizations to make informed decisions that support organizational objectives, to enforce policies for hardening systems, to prevent network misuse, to quantify business risk, and to collaborate with partners to identify and mitigate threats.

It is expected that use cases for enterprises and for service providers will largely overlap. When considering this overlap, there are additional complications for service providers, especially in handling information that crosses administrative domains.

The output of endpoint posture assessment is expected to feed into additional processes, such as policy-based enforcement of acceptable state, verification and monitoring of security controls, and compliance to regulatory requirements.

2. Endpoint Posture Assessment

Endpoint posture assessment involves orchestrating and performing data collection and evaluating the posture of a given endpoint. Typically, endpoint posture information is gathered and then published to appropriate data repositories to make collected information available for further analysis supporting organizational security processes.

Endpoint posture assessment typically includes:

- o Collecting the attributes of a given endpoint;
- o Making the attributes available for evaluation and action; and
- o Verifying that the endpoint's posture is in compliance with enterprise standards and policy.

As part of these activities, it is often necessary to identify and acquire any supporting security automation data that is needed to drive and feed data collection and evaluation processes.

The following is a typical workflow scenario for assessing endpoint posture:

1. Some type of trigger initiates the workflow. For example, an operator or an application might trigger the process with a request, or the endpoint might trigger the process using an event-driven notification.
2. An operator/application selects one or more target endpoints to be assessed.
3. An operator/application selects which policies are applicable to the targets.
4. For each target:
 - A. The application determines which (sets of) posture attributes need to be collected for evaluation. Implementations should be able to support (possibly mixed) sets of standardized and proprietary attributes.

- B. The application might retrieve previously collected information from a cache or data store, such as a data store populated by an asset management system.
- C. The application might establish communication with the target, mutually authenticate identities and authorizations, and collect posture attributes from the target.
- D. The application might establish communication with one or more intermediary/agents, mutually authenticate their identities and determine authorizations, and collect posture attributes about the target from the intermediary/agents. Such agents might be local or external.
- E. The application communicates target identity and (sets of) collected attributes to an evaluator, possibly an external process or external system.
- F. The evaluator compares the collected posture attributes with expected values as expressed in policies.
- G. The evaluator reports the evaluation result for the requested assessment, in a standardized or proprietary format, such as a report, a log entry, a database entry, or a notification.

2.1. Use Cases

The following subsections detail specific use cases for assessment planning, data collection, analysis, and related operations pertaining to the publication and use of supporting data. Each use case is defined by a short summary containing a simple problem statement, followed by a discussion of related concepts, and a listing of associated building blocks which represent the capabilities needed to support the use case. These use cases and building blocks identify separate units of functionality that may be supported by different components of an architectural model.

2.1.1. Define, Publish, Query and Retrieve Security Automation Data

This use case describes the need for security automation data to be defined and published to one or more data stores, as well as queried and retrieved from these data stores for the explicit use of posture collection and evaluation.

Security automation data is a general concept that refers to any data expression that may be generated and/or used as part of the process of collecting and evaluating endpoint posture. Different types of

security automation data will generally fall into one of three categories:

Guidance: Instructions and related metadata that guide the attribute collection and evaluation processes. The purpose of this data is to allow implementations to be data-driven enabling their behavior to be customized without requiring changes to deployed software.

This type of data tends to change in units of months and days. In cases where assessments are made more dynamic, it may be necessary to handle changes in the scope of hours or minutes. This data will typically be provided by large organizations, product vendors, and some 3rd-parties. Thus, it will tend to be shared across large enterprises and customer communities. In some cases access may be controlled to specific authenticated users. In other cases, the data may be provided broadly with little to no access control.

This includes:

- * Listings of attribute identifiers for which values may be collected and evaluated
- * Lists of attributes that are to be collected along with metadata that includes: when to collect a set of attributes based on a defined interval or event, the duration of collection, and how to go about collecting a set of attributes.
- * Guidance that specifies how old collected data can be to be used for evaluation.
- * Policies that define how to target and perform the evaluation of a set of attributes for different kinds or groups of endpoints and the assets they are composed of. In some cases it may be desirable to maintain hierarchies of policies as well.
- * References to human-oriented data that provide technical, organizational, and/or policy context. This might include references to: best practices documents, legal guidance and legislation, and instructional materials related to the automation data in question.

Attribute Data: Data collected through automated and manual mechanisms describing organizational and posture details pertaining to specific endpoints and the assets that they are

composed of (e.g., hardware, software, accounts). The purpose of this type of data is to characterize an endpoint (e.g., endpoint type, organizationally expected function/role) and to provide actual and expected state data pertaining to one or more endpoints. This data is used to determine what posture attributes to collect from which endpoints and to feed one or more evaluations.

This type of data tends to change in units of days, minutes, a seconds with posture attribute values typically changing more frequently than endpoint characterizations. This data tends to be organizationally and endpoint specific, with specific operational groups of endpoints tending to exhibit similar attribute profiles. This data will generally not be shared outside an organizational boundary and will generally require authentication with specific access controls.

This includes:

- * Endpoint characterization data that describes the endpoint type, organizationally expected function/role, etc.
- * Collected endpoint posture attribute values and related context including: time of collection, tools used for collection, etc.
- * Organizationally defined expected posture attribute values targeted to specific evaluation guidance and endpoint characteristics. This allows a common set of guidance to be parameterized for use with different groups of endpoints.

Processing Artifacts: Data that is generated by, and is specific to, an individual assessment process. This data may be used as part of the interactions between architectural components to drive and coordinate collection and evaluation activities. Its lifespan will be bounded by the lifespan of the assessment. It may also be exchanged and stored to provide historic context around an assessment activity so that individual assessments can be grouped, evaluated, and reported in an enterprise context.

This includes:

- * The identified set of endpoints for which an assessment should be performed.
- * The identified set of posture attributes that need to be collected from specific endpoints to perform an evaluation.

- * The resulting data generated by an evaluation process including the context of what was assessed, what it was assessed against, what collected data was used, when it was collected, and when the evaluation was performed.

The information model for security automation data must support a variety of different data types as described above, along with the associated metadata that is needed to support publication, query, and retrieval operations. It is expected that multiple data models will be used to express specific data types requiring specialized or extensible security automation data repositories. The different temporal characteristics, access patterns, and access control dimensions of each data type may also require different protocols and data models to be supported furthering the potential requirement for specialized data repositories. See [RFC3444] for a description and discussion of distinctions between an information and data model. It is likely that additional kinds of data will be identified through the process of defining requirements and an architectural model. Implementations supporting this building block will need to be extensible to accommodate the addition of new types of data, both proprietary or (preferably) using a standard format.

The building blocks of this use case are:

Data Definition: Security automation data will guide and inform collection and evaluation processes. This data may be designed by a variety of roles - application implementers may build security automation data into their applications; administrators may define guidance based on organizational policies; operators may define guidance and attribute data as needed for evaluation at runtime, and so on. Data producers may choose to reuse data from existing stores of security automation data and/or may create new data. Data producers may develop data based on available standardized or proprietary data models, such as those used for network management and/or host management.

Data Publication: The capability to enable data producers to publish data to a security automation data store for further use. Published data may be made publicly available or access may be based on an authorization decision using authenticated credentials. As a result, the visibility of specific security automation data to an operator or application may be public, enterprise-scoped, private, or controlled within any other scope.

Data Query: An operator or application should be able to query a security automation data store using a set of specified

criteria. The result of the query will be a listing matching the query. The query result listing may contain publication metadata (e.g., create date, modified date, publisher, etc.) and/or the full data, a summary, snippet, or the location to retrieve the data.

Data Retrieval: A user, operator, or application acquires one or more specific security automation data entries. The location of the data may be known a priori, or may be determined based on decisions made using information from a previous query.

Data Change Detection: An operator or application needs to know when security automation data they interested in has been published to, updated in, or deleted from a security automation data store which they have been authorized to access.

These building blocks are used to enable acquisition of various instances of security automation data based on specific data models that are used to drive assessment planning (see section 2.1.2), posture attribute value collection (see section 2.1.3), and posture evaluation (see section 2.1.4).

2.1.2. Endpoint Identification and Assessment Planning

This use case describes the process of discovering endpoints, understanding their composition, identifying the desired state to assess against, and calculating what posture attributes to collect to enable evaluation. This process may be a set of manual, automated, or hybrid steps that are performed for each assessment.

The building blocks of this use case are:

Endpoint Discovery: To determine the current or historic presence of endpoints in the environment that are available for posture assessment. Endpoints are identified in support of discovery using information previously obtained or by using other collection mechanisms to gather identification and characterization data. Previously obtained data may originate from sources such as network authentication exchanges.

Endpoint Characterization: The act of acquiring, through automated collection or manual input, and organizing attributes associated with an endpoint (e.g., type, organizationally expected function/role, hardware/software versions).

Identify Endpoint Targets: Determine the candidate endpoint target(s) against which to perform the assessment. Depending on the assessment trigger, a single endpoint or multiple

endpoints may be targeted based on characterized endpoint attributes. Guidance describing the assessment to be performed may contain instructions or references used to determine the applicable assessment targets. In this case the Data Query and/or Data Retrieval building blocks (see section 2.1.1) may be used to acquire this data.

Endpoint Component Inventory: To determine what applicable desired states should be assessed, it is first necessary to acquire the inventory of software, hardware, and accounts associated with the targeted endpoint(s). If the assessment of the endpoint is not dependent on these details, then this capability is not required for use in performing the assessment. This process can be treated as a collection use case for specific posture attributes. In this case the building blocks for Endpoint Posture Attribute Value Collection (see section 2.1.3) can be used.

Posture Attribute Identification: Once the endpoint targets and their associated asset inventory is known, it is then necessary to calculate what posture attributes are required to be collected to perform the desired evaluation. When available, existing posture data is queried for suitability using the Data Query building block (see section 2.1.1). Such posture data is suitable if it is complete and current enough for use in the evaluation. Any unsuitable posture data is identified for collection.

If this is driven by guidance, then the Data Query and/or Data Retrieval building blocks (see section 2.1.1) may be used to acquire this data.

At this point the set of posture attribute values to use for evaluation are known and they can be collected if necessary (see section 2.1.3).

2.1.3. Endpoint Posture Attribute Value Collection

This use case describes the process of collecting a set of posture attribute values related to one or more endpoints. This use case can be initiated by a variety of triggers including:

1. A posture change or significant event on the endpoint.
2. A network event (e.g., endpoint connects to a network/VPN, specific netflow is detected).
3. A scheduled or ad hoc collection task.

The building blocks of this use case are:

Collection Guidance Acquisition: If guidance is required to drive the collection of posture attributes values, this capability is used to acquire this data from one or more security automation data stores. Depending on the trigger, the specific guidance to acquire might be known. If not, it may be necessary to determine the guidance to use based on the component inventory or other assessment criteria. The Data Query and/or Data Retrieval building blocks (see section 2.1.1) may be used to acquire this guidance.

Posture Attribute Value Collection: The accumulation of posture attribute values. This may be based on collection guidance that is associated with the posture attributes.

Once the posture attribute values are collected, they may be persisted for later use or they may be immediately used for posture evaluation.

2.1.4. Posture Attribute Evaluation

This use case represents the action of analyzing collected posture attribute values as part of an assessment. The primary focus of this use case is to support evaluation of actual endpoint state against the expected state selected for the assessment.

This use case can be initiated by a variety of triggers including:

1. A posture change or significant event on the endpoint.
2. A network event (e.g., endpoint connects to a network/VPN, specific netflow is detected).
3. A scheduled or ad hoc evaluation task.

The building blocks of this use case are:

Collected Posture Change Detection: An operator or application has a mechanism to detect the availability of new, or changes to existing, posture attribute values. The timeliness of detection may vary from immediate to on-demand. Having the ability to filter what changes are detected will allow the operator to focus on the changes that are relevant to their use and will enable evaluation to occur dynamically based on detected changes.

Posture Attribute Value Query: If previously collected posture attribute values are needed, the appropriate data stores are queried to retrieve them using the Data Query building block (see section 2.1.1). If all posture attribute values are provided directly for evaluation, then this capability may not be needed.

Evaluation Guidance Acquisition: If guidance is required to drive the evaluation of posture attributes values, this capability is used to acquire this data from one or more security automation data stores. Depending on the trigger, the specific guidance to acquire might be known. If not, it may be necessary to determine the guidance to use based on the component inventory or other assessment criteria. The Data Query and/or Data Retrieval building blocks (see section 2.1.1) may be used to acquire this guidance.

Posture Attribute Evaluation: The comparison of posture attribute values against their expected values as expressed in the specified guidance. The result of this comparison is output as a set of posture evaluation results. Such results include metadata required to provide a level of assurance with respect to the posture attribute data and, therefore, evaluation results. Examples of such metadata include provenance and or availability data.

While the primary focus of this use case is around enabling the comparison of expected vs. actual state, the same building blocks can support other analysis techniques that are applied to collected posture attribute data (e.g., trending, historic analysis).

Completion of this process represents a complete assessment cycle as defined in Section 2.

2.2. Usage Scenarios

In this section, we describe a number of usage scenarios that utilize aspects of endpoint posture assessment. These are examples of common problems that can be solved with the building blocks defined above.

2.2.1. Definition and Publication of Automatable Configuration Checklists

A vendor manufactures a number of specialized endpoint devices. They also develop and maintain an operating system for these devices that enables end-user organizations to configure a number of security and operational settings. As part of their customer support activities,

they publish a number of secure configuration guides that provide minimum security guidelines for configuring their devices.

Each guide they produce applies to a specific model of device and version of the operating system and provides a number of specialized configurations depending on the device's intended function and what add-on hardware modules and software licenses are installed on the device. To enable their customers to evaluate the security posture of their devices to ensure that all appropriate minimal security settings are enabled, they publish an automatable configuration checklists using a popular data format that defines what settings to collect using a network management protocol and appropriate values for each setting. They publish these checklists to a public security automation data store that customers can query to retrieve applicable checklist(s) for their deployed specialized endpoint devices.

Automatable configuration checklist could also come from sources other than a device vendor, such as industry groups or regulatory authorities, or enterprises could develop their own checklists.

This usage scenario employs the following building blocks defined in Section 2.1.1 above:

Data Definition: To allow guidance to be defined using standardized or proprietary data models that will drive collection and evaluation.

Data Publication: Providing a mechanism to publish created guidance to a security automation data store.

Data Query: To locate and select existing guidance that may be reused.

Data Retrieval To retrieve specific guidance from a security automation data store for editing.

While each building block can be used in a manual fashion by a human operator, it is also likely that these capabilities will be implemented together in some form of a guidance editor or generator application.

2.2.2. Automated Checklist Verification

A financial services company operates a heterogeneous IT environment. In support of their risk management program, they utilize vendor provided automatable security configuration checklists for each operating system and application used within their IT environment.

Multiple checklists are used from different vendors to insure adequate coverage of all IT assets.

To identify what checklists are needed, they use automation to gather an inventory of the software versions utilized by all IT assets in the enterprise. This data gathering will involve querying existing data stores of previously collected endpoint software inventory posture data and actively collecting data from reachable endpoints as needed utilizing network and systems management protocols. Previously collected data may be provided by periodic data collection, network connection-driven data collection, or ongoing event-driven monitoring of endpoint posture changes.

Appropriate checklists are queried, located and downloaded from the relevant guidance data stores. The specific data stores queried and the specifics of each query may be driven by data including:

- o collected hardware and software inventory data, and
- o associated asset characterization data that may indicate the organizational defined functions of each endpoint.

Checklists may be sourced from guidance data stores maintained by an application or OS vendor, an industry group, a regulatory authority, or directly by the enterprise.

The retrieved guidance is cached locally to reduce the need to retrieve the data multiple times.

Driven by the setting data provided in the checklist, a combination of existing configuration data stores and data collection methods are used to gather the appropriate posture attributes from (or pertaining to) each endpoint. Specific posture attribute values are gathered based on the defined enterprise function and software inventory of each endpoint. The collection mechanisms used to collect software inventory posture will be used again for this purpose. Once the data is gathered, the actual state is evaluated against the expected state criteria defined in each applicable checklist.

A checklist can be assessed as a whole, or a specific subset of the checklist can be assessed resulting in partial data collection and evaluation.

The results of checklist evaluation are provided to appropriate operators and applications to drive additional business logic. Specific applications for checklist evaluation results are out-of-scope for current SACM efforts. Irrespective of specific applications, the availability, timeliness, and liveness of results

is often of general concern. Network latency and available bandwidth often create operational constraints that require trade-offs between these concerns and need to be considered.

Uses of checklists and associated evaluation results may include, but are not limited to:

- o Detecting endpoint posture deviations as part of a change management program to:
 - * identify missing required patches,
 - * unauthorized changes to hardware and software inventory, and
 - * unauthorized changes to configuration items.
- o Determining compliance with organizational policies governing endpoint posture.
- o Informing configuration management, patch management, and vulnerability mitigation and remediation decisions.
- o Searching for current and historic indicators of compromise.
- o Detecting current and historic infection by malware and determining the scope of infection within an enterprise.
- o Detecting performance, attack and vulnerable conditions that warrant additional network diagnostics, monitoring, and analysis.
- o Informing network access control decision making for wired, wireless, or VPN connections.

This usage scenario employs the following building blocks defined in Section 2.1.1 above:

Endpoint Discovery: The purpose of discovery is to determine the type of endpoint to be posture assessed.

Identify Endpoint Targets: To identify what potential endpoint targets the checklist should apply to based on organizational policies.

Endpoint Component Inventory: Collecting and consuming the software and hardware inventory for the target endpoints.

Posture Attribute Identification: To determine what data needs to be collected to support evaluation, the checklist is evaluated

against the component inventory and other endpoint metadata to determine the set of posture attribute values that are needed.

Collection Guidance Acquisition: Based on the identified posture attributes, the application will query appropriate security automation data stores to find the "applicable" collection guidance for each endpoint in question.

Posture Attribute Value Collection: For each endpoint, the values for the required posture attributes are collected.

Posture Attribute Value Query: If previously collected posture attribute values are used, they are queried from the appropriate data stores for the target endpoint(s).

Evaluation Guidance Acquisition: Any guidance that is needed to support evaluation is queried and retrieved.

Posture Attribute Evaluation: The resulting posture attribute values from previous collection processes are evaluated using the evaluation guidance to provide a set of posture results.

2.2.3. Detection of Posture Deviations

Example corporation has established secure configuration baselines for each different type of endpoint within their enterprise including: network infrastructure, mobile, client, and server computing platforms. These baselines define an approved list of hardware, software (i.e., operating system, applications, and patches), and associated required configurations. When an endpoint connects to the network, the appropriate baseline configuration is communicated to the endpoint based on its location in the network, the expected function of the device, and other asset management data. It is checked for compliance with the baseline indicating any deviations to the device's operators. Once the baseline has been established, the endpoint is monitored for any change events pertaining to the baseline on an ongoing basis. When a change occurs to posture defined in the baseline, updated posture information is exchanged, allowing operators to be notified and/or automated action to be taken.

Like the Automated Checklist Verification usage scenario (see section 2.2.2), this usage scenario supports assessment based on automatable checklists. It differs from that scenario by monitoring for specific endpoint posture changes on an ongoing basis. When the endpoint detects a posture change, an alert is generated identifying the specific changes in posture allowing assessment of the delta to be performed instead of a full assessment in the previous case. This

usage scenario employs the same building blocks as Automated Checklist Verification (see section 2.2.2). It differs slightly in how it uses the following building blocks:

Endpoint Component Inventory: Additionally, changes to the hardware and software inventory are monitored, with changes causing alerts to be issued.

Posture Attribute Value Collection: After the initial assessment, posture attributes are monitored for changes. If any of the selected posture attribute values change, an alert is issued.

Posture Attribute Value Query: The previous state of posture attributes are tracked, allowing changes to be detected.

Posture Attribute Evaluation: After the initial assessment, a partial evaluation is performed based on changes to specific posture attributes.

This usage scenario highlights the need to query a data store to prepare a compliance report for a specific endpoint and also the need for a change in endpoint state to trigger Collection and Evaluation.

2.2.4. Endpoint Information Analysis and Reporting

Freed from the drudgery of manual endpoint compliance monitoring, one of the security administrators at Example Corporation notices (not using SACM standards) that five endpoints have been uploading lots of data to a suspicious server on the Internet. The administrator queries data stores for specific endpoint posture to see what software is installed on those endpoints and finds that they all have a particular program installed. She then queries the appropriate data stores to see which other endpoints have that program installed. All these endpoints are monitored carefully (not using SACM standards), which allows the administrator to detect that the other endpoints are also infected.

This is just one example of the useful analysis that a skilled analyst can do using data stores of endpoint posture.

This usage scenario employs the following building blocks defined in Section 2.1.1 above:

Posture Attribute Value Query: Previously collected posture attribute values for the target endpoint(s) are queried from the appropriate data stores using a standardized method.

This usage scenario highlights the need to query a repository for attributes to see which attributes certain endpoints have in common.

2.2.5. Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra

A university team receives a grant to do research at a government facility in the arctic. The only network communications will be via an intermittent, low-speed, high-latency, high-cost satellite link. During their extended expedition, they will need to show continue compliance with the security policies of the university, the government, and the provider of the satellite network as well as keep current on vulnerability testing. Interactive assessments are therefore not reliable, and since the researchers have very limited funding they need to minimize how much money they spend on network data.

Prior to departure they register all equipment with an asset management system owned by the university, which will also initiate and track assessments.

On a periodic basis -- either after a maximum time delta or when the security automation data store has received a threshold level of new vulnerability definitions -- the university uses the information in the asset management system to put together a collection request for all of the deployed assets that encompasses the minimal set of artifacts necessary to evaluate all three security policies as well as vulnerability testing.

In the case of new critical vulnerabilities, this collection request consists only of the artifacts necessary for those vulnerabilities and collection is only initiated for those assets that could potentially have a new vulnerability.

(Optional) Asset artifacts are cached in a local CMDB. When new vulnerabilities are reported to the security automation data store, a request to the live asset is only done if the artifacts in the CMDB are incomplete and/or not current enough.

The collection request is queued for the next window of connectivity. The deployed assets eventually receive the request, fulfill it, and queue the results for the next return opportunity.

The collected artifacts eventually make it back to the university where the level of compliance and vulnerability exposed is calculated and asset characteristics are compared to what is in the asset management system for accuracy and completeness.

Like the Automated Checklist Verification usage scenario (see section 2.2.2), this usage scenario supports assessment based on checklists. It differs from that scenario in how guidance, collected posture attribute values, and evaluation results are exchanged due to bandwidth limitations and availability. This usage scenario employs the same building blocks as Automated Checklist Verification (see section 2.2.2). It differs slightly in how it uses the following building blocks:

Endpoint Component Inventory: It is likely that the component inventory will not change. If it does, this information will need to be batched and transmitted during the next communication window.

Collection Guidance Acquisition: Due to intermittent communication windows and bandwidth constraints, changes to collection guidance will need to be batched and transmitted during the next communication window. Guidance will need to be cached locally to avoid the need for remote communications.

Posture Attribute Value Collection: The specific posture attribute values to be collected are identified remotely and batched for collection during the next communication window. If a delay is introduced for collection to complete, results will need to be batched and transmitted.

Posture Attribute Value Query: Previously collected posture attribute values will be stored in a remote data store for use at the university

Evaluation Guidance Acquisition: Due to intermittent communication windows and bandwidth constraints, changes to evaluation guidance will need to be batched and transmitted during the next communication window. Guidance will need to be cached locally to avoid the need for remote communications.

Posture Attribute Evaluation: Due to the caching of posture attribute values and evaluation guidance, evaluation may be performed at both the university campus as well as the satellite site.

This usage scenario highlights the need to support low-bandwidth, intermittent, or high-latency links.

2.2.6. Identification and Retrieval of Guidance

In preparation for performing an assessment, an operator or application will need to identify one or more security automation data stores that contain the guidance entries necessary to perform data collection and evaluation tasks. The location of a given guidance entry will either be known a priori or known security automation data stores will need to be queried to retrieve applicable guidance.

To query guidance it will be necessary to define a set of search criteria. This criteria will often utilize a logical combination of publication metadata (e.g. publishing identity, create time, modification time) and guidance data-specific criteria elements. Once the criteria is defined, one or more security automation data stores will need to be queried generating a result set. Depending on how the results are used, it may be desirable to return the matching guidance directly, a snippet of the guidance matching the query, or a resolvable location to retrieve the data at a later time. The guidance matching the query will be restricted based the authorized level of access allowed to the requester.

If the location of guidance is identified in the query result set, the guidance will be retrieved when needed using one or more data retrieval requests. A variation on this approach would be to maintain a local cache of previously retrieved data. In this case, only guidance that is determined to be stale by some measure will be retrieved from the remote data store.

Alternately, guidance can be discovered by iterating over data published with a given context within a security automation data store. Specific guidance can be selected and retrieved as needed.

This usage scenario employs the following building blocks defined in Section 2.1.1 above:

Data Query: Enables an operator or application to query one or more security automation data stores for guidance using a set of specified criteria.

Data Retrieval: If data locations are returned in the query result set, then specific guidance entries can be retrieved and possibly cached locally.

2.2.7. Guidance Change Detection

An operator or application may need to identify new, updated, or deleted guidance in a security automation data store for which they have been authorized to access. This may be achieved by querying or iterating over guidance in a security automation data store, or through a notification mechanism that alerts to changes made to a security automation data store.

Once guidance changes have been determined, data collection and evaluation activities may be triggered.

This usage scenario employs the following building blocks defined in Section 2.1.1 above:

Data Change Detection: Allows an operator or application to identify guidance changes in a security automation data store which they have been authorized to access.

Data Retrieval: If data locations are provided by the change detection mechanism, then specific guidance entries can be retrieved and possibly cached locally.

3. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

This memo documents, for informational purposes, use cases for security automation. Specific security and privacy considerations will be provided in related documents (e.g., requirements, architecture, information model, data model, protocol) as appropriate to the function described in each related document.

One consideration for security automation is that a malicious actor could use the security automation infrastructure and related collected data to gain access to an item of interest. This may include personal data, private keys, software and configuration state that can be used to inform an attack against the network and endpoints, and other sensitive information. It is important that security and privacy considerations in the related documents identify methods to both identify and prevent such activity.

For consideration are means for protecting the communications as well as the systems that store the information. For communications between the varying SACM components there should be considerations for protecting the confidentiality, data integrity and peer entity

authentication. For exchanged information, there should be a means to authenticate the origin of the information. This is important where tracking the provenance of data is needed. Also, for any systems that store information that could be used for unauthorized or malicious purposes, methods to identify and protect against unauthorized usage, inappropriate usage, and denial of service need to be considered.

5. Acknowledgements

Adam Montville edited early versions of this draft.

Kathleen Moriarty, and Stephen Hanna contributed text describing the scope of the document.

Gunnar Engelbach, Steve Hanna, Chris Inacio, Kent Landfield, Lisa Lorenzin, Adam Montville, Kathleen Moriarty, Nancy Cam-Winget, and Aron Woland provided use cases text for various revisions of this draft.

6. Change Log

6.1. -08- to -09-

Fixed a number of gramatical nits throughout the draft identified by the SECDIR review.

Added additional text to the security considerations about malicious actors.

6.2. -07- to -08-

Reworked long sentences throughout the document by shortening or using bulleted lists.

Re-ordered and condensed text in the "Automated Checklist Verification" sub-section to improve the conceptual presentation and to clarify longer sentences.

Clarified that the "Posture Attribute Value Query" building block represents a standardized interface in the context of SACM.

Removed the "others" sub-section within the "usage scenarios" section.

Updated the "Security Considerations" section to identify that actual SACM security considerations will be discussed in the appropriate related documents.

6.3. -06- to -07-

A number of edits were made to section 2 to resolve open questions in the draft based on meeting and mailing list discussions.

Section 2.1.5 was merged into section 2.1.4.

6.4. -05- to -06-

Updated the "Introduction" section to better reflect the use case, building block, and usage scenario structure changes from previous revisions.

Updated most uses of the terms "content" and "content repository" to use "guidance" and "security automation data store" respectively.

In section 2.1.1, added a discussion of different data types and renamed "content" to "data" in the building block names.

In section 2.1.2, separated out the building block concepts of "Endpoint Discovery" and "Endpoint Characterization" based on mailing list discussions.

Addressed some open questions throughout the draft based on consensus from mailing list discussions and the two virtual interim meetings.

Changed many section/sub-section names to better reflect their content.

6.5. -04- to -05-

Changes in this revision are focused on section 2 and the subsequent subsections:

- o Moved existing use cases to a subsection titled "Usage Scenarios".
- o Added a new subsection titled "Use Cases" to describe the common use cases and building blocks used to address the "Usage Scenarios". The new use cases are:
 - * Define, Publish, Query and Retrieve Content
 - * Endpoint Identification and Assessment Planning
 - * Endpoint Posture Attribute Value Collection
 - * Posture Evaluation

- * Mining the Database

- o Added a listing of building blocks used for all usage scenarios.
- o Combined the following usage scenarios into "Automated Checklist Verification": "Organizational Software Policy Compliance", "Search for Signs of Infection", "Vulnerable Endpoint Identification", "Compromised Endpoint Identification", "Suspicious Endpoint Behavior", "Traditional endpoint assessment with stored results", "NAC/NAP connection with no stored results using an endpoint evaluator", and "NAC/NAP connection with no stored results using a third-party evaluator".
- o Created new usage scenario "Identification and Retrieval of Repository Content" by combining the following usage scenarios: "Repository Interaction - A Full Assessment" and "Repository Interaction - Filtered Delta Assessment"
- o Renamed "Register with repository for immediate notification of new security vulnerability content that match a selection filter" to "Content Change Detection" and generalized the description to be neutral to implementation approaches.
- o Removed out-of-scope usage scenarios: "Remediation and Mitigation" and "Direct Human Retrieval of Ancillary Materials"

Updated acknowledgements to recognize those that helped with editing the use case text.

6.6. -03- to -04-

Added four new use cases regarding content repository.

6.7. -02- to -03-

Expanded the workflow description based on ML input.

Changed the ambiguous "assess" to better separate data collection from evaluation.

Added use case for Search for Signs of Infection.

Added use case for Remediation and Mitigation.

Added use case for Endpoint Information Analysis and Reporting.

Added use case for Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra.

Added use case for Traditional endpoint assessment with stored results.

Added use case for NAC/NAP connection with no stored results using an endpoint evaluator.

Added use case for NAC/NAP connection with no stored results using a third-party evaluator.

Added use case for Compromised Endpoint Identification.

Added use case for Suspicious Endpoint Behavior.

Added use case for Vulnerable Endpoint Identification.

Updated Acknowledgements

6.8. -01- to -02-

Changed title

removed section 4, expecting it will be moved into the requirements document.

removed the list of proposed capabilities from section 3.1

Added empty sections for Search for Signs of Infection, Remediation and Mitigation, and Endpoint Information Analysis and Reporting.

Removed Requirements Language section and rfc2119 reference.

Removed unused references (which ended up being all references).

6.9. -00- to -01-

- o Work on this revision has been focused on document content relating primarily to use of asset management data and functions.
- o Made significant updates to section 3 including:
 - * Reworked introductory text.
 - * Replaced the single example with multiple use cases that focus on more discrete uses of asset management data to support hardware and software inventory, and configuration management use cases.

- * For one of the use cases, added mapping to functional capabilities used. If popular, this will be added to the other use cases as well.
 - * Additional use cases will be added in the next revision capturing additional discussion from the list.
 - o Made significant updates to section 4 including:
 - * Renamed the section heading from "Use Cases" to "Functional Capabilities" since use cases are covered in section 3. This section now extrapolates specific functions that are needed to support the use cases.
 - * Started work to flatten the section, moving select subsections up from under asset management.
 - * Removed the subsections for: Asset Discovery, Endpoint Components and Asset Composition, Asset Resources, and Asset Life Cycle.
 - * Renamed the subsection "Asset Representation Reconciliation" to "Deconfliction of Asset Identities".
 - * Expanded the subsections for: Asset Identification, Asset Characterization, and Deconfliction of Asset Identities.
 - * Added a new subsection for Asset Targeting.
 - * Moved remaining sections to "Other Unedited Content" for future updating.
- 6.10. draft-waltermire-sacm-use-cases-05 to draft-ietf-sacm-use-cases-00
- o Transitioned from individual I/D to WG I/D based on WG consensus call.
 - o Fixed a number of spelling errors. Thank you Erik!
 - o Added keywords to the front matter.
 - o Removed the terminology section from the draft. Terms have been moved to: draft-dbh-sacm-terminology-00
 - o Removed requirements to be moved into a new I/D.

- o Extracted the functionality from the examples and made the examples less prominent.
- o Renamed "Functional Capabilities and Requirements" section to "Use Cases".
 - * Reorganized the "Asset Management" sub-section. Added new text throughout.
 - + Renamed a few sub-section headings.
 - + Added text to the "Asset Characterization" sub-section.
- o Renamed "Security Configuration Management" to "Endpoint Configuration Management". Not sure if the "security" distinction is important.
 - * Added new sections, partially integrated existing content.
 - * Additional text is needed in all of the sub-sections.
- o Changed "Security Change Management" to "Endpoint Posture Change Management". Added new skeletal outline sections for future updates.

6.11. waltermire -04- to -05-

- o Are we including user activities and behavior in the scope of this work? That seems to be layer 8 stuff, appropriate to an IDS/IPS application, not Internet stuff.
- o Removed the references to what the WG will do because this belongs in the charter, not the (potentially long-lived) use cases document. I removed mention of charter objectives because the charter may go through multiple iterations over time; there is a website for hosting the charter; this document is not the correct place for that discussion.
- o Moved the discussion of NIST specifications to the acknowledgements section.
- o Removed the portion of the introduction that describes the chapters; we have a table of concepts, and the existing text seemed redundant.
- o Removed marketing claims, to focus on technical concepts and technical analysis, that would enable subsequent engineering effort.

- o Removed (commented out in XML) UC2 and UC3, and eliminated some text that referred to these use cases.
- o Modified IANA and Security Consideration sections.
- o Moved Terms to the front, so we can use them in the subsequent text.
- o Removed the "Key Concepts" section, since the concepts of ORM and IRM were not otherwise mentioned in the document. This would seem more appropriate to the arch doc rather than use cases.
- o Removed role=editor from David Waltermire's info, since there are three editors on the document. The editor is most important when one person writes the document that represents the work of multiple people. When there are three editors, this role marking isn't necessary.
- o Modified text to describe that this was specific to enterprises, and that it was expected to overlap with service provider use cases, and described the context of this scoped work within a larger context of policy enforcement, and verification.
- o The document had asset management, but the charter mentioned asset, change, configuration, and vulnerability management, so I added sections for each of those categories.
- o Added text to Introduction explaining goal of the document.
- o Added sections on various example use cases for asset management, config management, change management, and vulnerability management.

7. Informative References

- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

Authors' Addresses

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

Email: david.waltermire@nist.gov

David Harrington
Effective Software
50 Harding Rd
Portsmouth, NH 03801
USA

Email: ietfdbh@comcast.net