          Endpoint Security Posture Assessment - Enterprise Use Cases
                       draft-ietf-sacm-use-cases-08

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

Copyright Notice

Table of Contents

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.

Togther 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, but 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 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: An 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 the 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 cases 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 devices 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 checklist to a public security automation data store that customers can query to retrieve applicable checklist 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 expose 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 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 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 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.

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. -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.

6.2. -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.

6.3. -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.

6.4. **-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.

6.5. -03- to -04-

Added four new use cases regarding content repository.

6.6. -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

6.7. -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).

6.8. -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.

6.9. **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.

6.10. **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