

Security Automation and Continuous Monitoring WG
Internet-Draft
Intended status: Informational
Expires: May 24, 2014

D. Waltermire
NIST
D. Harrington
Effective Software
November 20, 2013

Endpoint Security Posture Assessment - Enterprise Use Cases
draft-ietf-sacm-use-cases-05

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 May 24, 2014.

Copyright Notice

Copyright (c) 2013 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 Content	5
2.1.2.	Endpoint Identification and Assessment Planning	7
2.1.3.	Endpoint Posture Attribute Value Collection	8
2.1.4.	Posture Evaluation	8
2.1.5.	Mining the Database	9
2.2.	Usage Scenarios	10
2.2.1.	Definition and Publication of Automatable Configuration Guides	10
2.2.2.	Automated Checklist Verification	11
2.2.3.	Detection of Posture Deviations	13
2.2.4.	Endpoint Information Analysis and Reporting	14
2.2.5.	Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra	15
2.2.6.	Identification and Retrieval of Repository Content	17
2.2.7.	Content Change Detection	18
2.2.8.	Others...	18
3.	IANA Considerations	18
4.	Security Considerations	18
5.	Acknowledgements	18
6.	Change Log	19
6.1.	-04- to -05-	19
6.2.	-03- to -04-	20
6.3.	-02- to -03-	20
6.4.	-01- to -02-	21
6.5.	-00- to -01-	21
6.6.	draft-waltermire-sacm-use-cases-05 to draft-ietf-sacm- use-cases-00	22
6.7.	waltermire -04- to -05-	23
7.	References	24
7.1.	Normative References	24
7.2.	Informative References	24
	Authors' Addresses	24

[1.](#) Introduction

Our goal with this document is to improve our agreement on which problems we're trying to solve. We need to start with short, simple problem statements and discuss those by email and in person. Once we agree on which problems we're trying to solve, we can move on to propose various solutions and decide which ones to use.

This document describes example use cases for endpoint posture assessment for enterprises. It provides a sampling of use cases for securely aggregating configuration and operational data and evaluating that data to determine the security posture of individual endpoints, and, in the aggregate, the security posture of an enterprise.

These use cases cross many IT security information domains. From these operational use cases, we can derive common concepts, common information expressions, functional capabilities and requirements to guide development of vendor-neutral, interoperable standards for 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 content that is needed to drive data collection and analysis.

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.

QUESTION: Since this is about security automation, can we drop the User and just use Application? Is there a better term to use here? Once the policy is selected, the rest seems like something we definitely would want to automate, so I dropped the User part.

2. An operator/application selects one or more target endpoints to be assessed.
3. A 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.

QUESTION: It was suggested that mentioning several common acquisition methods, such as local API, WMI, Puppet, DCOM, SNMP, CMDB query, and NEA, without forcing any specific method would be good. I have concerns this could devolve into a "what about my favorite?" contest. OTOH, the charter does specifically call for use of existing standards where applicable, so the use cases document might be a good neutral location for such information, and might force us to consider what types of external interfaces we might need to support when we consider the requirements. It appears that the generic workflow sequence would be a good place to mention such common acquisition methods.

- 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.

QUESTION: Evaluator generates a report or log or notification of some type?

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 content.

2.1.1. Define, Publish, Query and Retrieve Content

This use case describes the need for content to be defined and published to a data store, as well as queried and retrieved from the

data store for the explicit use of posture collection and evaluation. It is expected that multiple information models will be supported to address the information needed to support the exchange of endpoint metadata, and the collection and evaluation of endpoint posture attribute values. It is likely that multiple data models will be used to express these information models requiring specialized or extensible content data stores.

The building blocks of this use case are:

Content Definition: Defining the content to drive collection and evaluation. This may include evaluating existing stores of content to find content to reuse and the creation of new content. Developed content will be based on available data models which may be standardized or proprietary.

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

Content Query: An operator or application should be able to query a content 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 content, a summary, snippet, or the location to retrieve the content.

Content Retrieval: The act of acquiring one or more specific content entries. This capability is useful if the location of the content is known a priori, perhaps as the result of request based on decisions made using information from a previous query.

Content Change Detection: An operator or application needs to identify content of interest that is new, updated, or deleted in a content data store which they have been authorized to access.

These building blocks are used to enable acquisition of various instances of content 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: The purpose of discovery is to determine the type of endpoint to be posture assessed.

QUESTION: Is it just the type? Or is it to identify what endpoint instances to target for assessment using metadata such as the endpoint's organizationally expected type (e.g., expected function/role, etc.)

Identify Endpoint Targets Determine the candidate endpoint target(s) to perform the assessment against. Depending on the assessment trigger, a single endpoint may be targeted or multiple endpoints may be targeted based on discovered endpoint metadata. This may be driven by content that describes the applicable targets for assessment. In this case the Content Query and/or Content Retrieval building blocks (see [section 2.1.1](#)) may be used to acquire this content.

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 dependant on the component inventory, 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 component inventory is known, it is then necessary to calculate what posture attributes are required to be collected to perform the evaluation. If this is driven by content, then the Content Query and/or Content Retrieval building blocks (see [section 2.1.1](#)) may be used to acquire this content.

QUESTION: Are we missing a building block that determines what previously collected data, if any, is suitable for evaluation and what data needs to be actually collected?

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. Due to a scheduled or ad hoc collection task.

The building blocks of this use case are:

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

Posture Attribute Value Collection: The accumulation of posture attribute values. This may be based on collection content 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 Evaluation

This use case describes the process of evaluating collected posture attribute values representing 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. Due to a scheduled or ad hoc evaluation task.

The building blocks of this use case are:

Posture Attribute Value Query: If previously collected posture attribute values are needed, the appropriate data stores are queried to retrieve them. If all posture attribute values are provided directly for evaluation, then this capability may not be needed.

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

Posture Attribute Evaluation: The comparison of posture attribute values against their expected results as expressed in the specified content. The result of this comparison is output as a set of posture evaluation results.

Completion of this process represents a complete assessment cycle as defined in section [Section 2](#).

[2.1.5](#). Mining the Database

This use case describes the need to analyze previously collected posture attribute values from one or more endpoints. This is an alternate use case to Posture Evaluation (see [section 2.1.4](#) that uses collected posture attributes values for analysis processes that may do more than evaluating expected vs. actual state(s).

The building blocks of this use case are:

Query: Query a data store for specific posture attribute values.

Change Detection: An operator should have 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.

QUESTION: Does this warrant a separate use case, or should this be incorporated into the previous use case?

[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 Guides

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 checklist 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 content repository 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:

Content Definition: To allow content to be defined using standardized or proprietary data models that will drive Collection and Evaluation.

Content Publication: Providing a mechanism to publish created content to a content data store.

Content Query: To locate and select existing content that may be reused.

Content Retrieval To retrieve specific content from a content 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 content 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.

Using the gathered hardware and software inventory data and associated asset management data that may indicate the organizational defined functions of each endpoint, checklist content is queried, located and downloaded from the appropriate vendor and 3rd-party content repositories for the appropriate checklists. This content is cached locally to reduce the need to download the checklist content 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 each endpoint. Specific data is gathered based on the defined enterprise function and software inventory of each endpoint. The data collection paths 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 in each applicable checklist. The results of this evaluation are provided to appropriate operators and applications to drive additional business logic.

Checklists could include searching for indicators of compromise on the endpoint (e.g., file hashes); identifying malicious activity (e.g. command and control traffic); detecting presence of unauthorized/malicious software, hardware, and configuration items; and other indicators.

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.

Checklists could also come from sources other than the application or OS vendor, such as industry groups or regulatory authorities, or enterprises could develop their own checklists.

While specific applications for checklists results are out-of-scope for current SACM efforts, how the data is used may illuminate specific latency and bandwidth requirements. For this purpose use of checklist assessment results may include, but are not limited to:

- o Detecting endpoint posture deviations as part of a change management program to include changes to hardware and software inventory including patches, changes to configuration items, and other posture aspects.
- o Determining compliance with organizational policies governing endpoint posture.
- o Searching for current and historic signs of infection by malware and determining the scope of infection within an enterprise.
- o Informing configuration management, patch management, and vulnerability mitigation and remediation decisions.
- 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 Content Acquisition: Based on the identified posture attributes, the application will query appropriate content data stores to find the "applicable" data collection content 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 Content Acquisition: Any content 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 content 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 of 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 a delta assessment 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 are queried from the appropriate data stores for the target endpoint(s).

QUESTION: Should we include other building blocks here?

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 content repository 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 content repository, 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 exposure 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 of checklists. It differs from that scenario in how content, collected posture 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 Content Acquisition: Due to intermittent communication windows and bandwidth constraints, changes to collection content will need to be batched and transmitted during the next communication window. Content 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 in the same way.

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

Evaluation Content Acquisition: Due to intermittent communication windows and bandwidth constraints, changes to evaluation content will need to be batched and transmitted during the next communication window. Content 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 content, 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 Repository Content

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

To query content 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 content-specific criteria elements. Once the criteria is defined, one or more content 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 content directly, a snippet of the content matching the query, or a resolvable location to retrieve the content at a later time. The content matching the query will be restricted based the authorized level of access allowed to the requester.

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

Alternately, content can be discovered by iterating over content published with a given context within a content repository. Specific content can be selected and retrieved as needed.

This usage scenario employs the following building blocks defined in [Section 2.1.1](#) above:

Content Query: Enables an operator or application to query one or more content data stores for content using a set of specified criteria.

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

2.2.7. Content Change Detection

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

Once content 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:

Content Change Detection: Allows an operator or application to identify content changes in a content data store which they have been authorized to access.

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

2.2.8. Others...

Additional use cases will be identified as we work through other domains.

3. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

This memo documents, for Informational purposes, use cases for security automation. While it is about security, it does not affect security.

5. Acknowledgements

The National Institute of Standards and Technology (NIST) and/or the MITRE Corporation have developed specifications under the general term "Security Automation" including languages, protocols, enumerations, and metrics.

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. -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.

[6.2.](#) -03- to -04-

Added four new use cases regarding content repository.

[6.3.](#) -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

[6.4.](#) -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).

[6.5.](#) -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.

6.6. [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.

6.7. 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 I 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 I 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. References

7.1. Normative References

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

7.2. Informative References

- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", [RFC 2865](#), June 2000.

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

