SACM
Internet-Draft
Intended status: Informational
Expires: July 25, 2016

C. Coffin
B. Cheikes
C. Schmidt
D. Haynes
The MITRE Corporation
J. Fitzgerald-McKay
Department of Defense
D. Waltermire
National Institute of Standards and Technology

January 22, 2016

# SACM Vulnerability Assessment Scenario
# draft-coffin-sacm-vuln-scenario-01

## Abstract

This document provides a core narrative that walks through an automated enterprise vulnerability assessment scenario. It is aligned with the SACM use cases and begins with an enterprise ingesting vulnerability description data, followed by identifying endpoints on the network and collecting and storing information about them to enable posture assessment, and finally ends with assessing these endpoints against the vulnerability description data to determine which ones are affected. Processes that specifically overlap between this scenario and SACM use cases will be noted where applicable. Specifically, the relationship between this document and the SACM use case building block capabilities and the usage scenarios will be covered.

## 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 July 25, 2016.

Table of Contents

## 1. Scope

The purpose of this document is to describe a detailed scenario for vulnerability assessment, and identify aspects of this scenario that could be used in the development of an information model. This includes classes of data, major roles, and a high-level description of role interactions. Additionally, this scenario intends to inform engineering work on protocol and data model development. The focus of the document is entirely intra-organizational and covers enterprise handling of vulnerability description data. The document does not attempt to cover the security disclosure itself and any prior activities of the security researcher or discloser, nor does it attempt to cover the specific activities of the vendor whose software is the focus of the vulnerability description data (i.e., the vulnerable software).

For the purposes of this document, the term "vulnerability description data" is intended to mean: "Data intended to alert enterprise IT resources to the existence of a flaw or flaws in software, hardware, and/or firmware, which could potentially have an impact on enterprise functionality and/or security." For the purpose of this scenario, such data also includes information that can be used to determine (to some level of accuracy, although possibly not conclusively) whether or not the flaw is present within an enterprise, when compared to information about the state of the enterprise's endpoints. For those who are familiar with current security practices and terminology, the use of vulnerability description data is also synonymnous with security bulletin or advisory.

This document makes no attempt to provide a definition of a normalized data format (e.g. industry standard) for vulnerability description data although there is nothing precluding the development of such a normalized data format. Also, it does not attempt to define procedures by which a vulnerability discoverer coordinates the release of vulnerability description data to other parties.

## 2. Assumptions

A number of assumptions must be stated in order to further clarify the position and scope of this document.

- The document begins with the assumption that the enterprise has received vulnerability description data, and that the data has already been processed into a format that the enterprise's security software tools can understand and use. In particular, this document:
  - Does not discuss how the enterprise identifies potentially relevant vulnerability description data.
  - Does not discuss how the enterprise collects the vulnerability description data.
  - Does not discuss how the enterprise assesses the authenticity of the vulnerability description data.
  - Does not discuss parsing of the vulnerability description data into a usable format.
- The document assumes that the enterprise has a means of identifying enterprise endpoints. This could mean identifying endpoints as they join the network, actively scanning for connected endpoints, passive scanning of network traffic to identify connected endpoints, or some other method of accounting for the presence of all endpoints in the enterprise. The document also does not distinguish between physical endpoints and virtualized endpoints.
- The document assumes that the enterprise has a means of extracting relevant information about enterprise endpoints. Moreover, this extracted information is expressed in a format that is compatible with the information extracted from the vulnerability description data. The document:
  - Does not specify how relevant information is identified.
  - Does not specify the mechanics of how relevant information is extracted from the data sources (such as the endpoint itself).
  - Does not specify how extracted endpoint information and vulnerability description data is normalized to be compatible.

  Note that having a means of extracting relevant information about enterprise endpoints is within the scope of the SACM Endpoint

Security Posture Assessment process. In the case of this document, this sub-process is assumed to be existent.

- The document assumes that all information described in the steps below is available in the vulnerability description data and serves as the basis of this assessment. Likewise, the document assumes that the enterprise can provide all relevant information about any endpoint needed to perform the described analysis. The authors recognize that this will not always be the case, but these assumptions are taken in order to show the breadth of data utilization in this scenario. Less complete information may require variations to the described steps.

- The document assumes that the enterprise has a policy by which assessment of endpoints based on vulnerability description data is prioritized. The document:

  * Does not specify how prioritization occurs.

  * Does not specify how prioritization impacts assessment behaviors.

- The document assumes that the enterprise has a mechanism for long-term storage of vulnerability description data and endpoint assessment results (e.g., a data repository).

- This document assumes that the enterprise has a procedure for reassessment of endpoints at some point after initial assessment. The document:

  * Does not specify how a reassessment would impact individual assessment behaviors. (i.e., it is agnostic as to whether the assessment procedure is the same regardless of whether this is the first or a subsequent assessment for some set of vulnerability description data.)

  * Does not provide recommendations or specifics on reassessment intervals.

## 3. Endpoint Identification and Initial (Pre-Assessment) Data Collection

The first step in this scenario involves identifying endpoints and collecting the basic or minumum set of system information attributes from them such as operating system type and version. Further examples of system information and attributes can be found below in the section titled Endpoint Data Collection. This identification occurs prior to the receipt of any specific vulnerability description data and is part of the regular, ongoing monitoring of endpoints

within an enterprise. This process is not meant to report on, or gather data for any specific vulnerabilities. The information gathered during this step could be applied in many enterprise automation efforts. Specifically, in addition to vulnerability management, it could be used by configuration and license management tasks. All of the information collected during this step is stored in a central location such as a Repository.

This activity involves the following sub-steps:

### 3.1. Identification

Prior to any other steps, the identification of endpoints must occur. This involves locating (at least virtually) and distinguishing between endpoints on the network in a way that allows each endpoint to be recognized in future interactions and selected for specific treatment. This not only allows later steps to determine the scope of what endpoints need to be assessed, but also allows for the unique identification of each endpoint. Unique and persistent endpoint IDs are used to allow for endpoints to be tracked over time and between sensors as well as allow for proper counts of assets during inventories and other similar collections. Endpoint identity can be established by collecting certain attributes that allow for unique and persistent tracking of endpoints on the enterprise network. Examples include, but are not limited to, IP address, MAC address, FQDNs, pre-provisioned identifiers such as GUIDs or copies of serial numbers, certificates, hardware identity values, or similar attributes. It is important to note that the persistency of these attributes will likely vary depending on the enterprise. For example, a statically assigned IP address is much more persistent than an IP address assigned via DHCP.

#### 3.1.1. SACM Use Case Alignment

This sub-step aligns with the Endpoint Discovery, Endpoint Characterization, and Endpoint Target Identification building block capabilities. The alignment is due to the fact that the purpose of this sub-step is to discover, identify, and characterize all endpoints on an enterprise network.

### 3.2. Processing Artifacts

Processing artifacts, such as the date and time the collection was performed, should be collected and stored. This timestamp is extremely important when performing later assessments, as it is needed for data freshness computations. The organization may develop rules for stale data and when a new data collection is required. This metadata is also helpful in correlating information across

multiple data collections. This includes correlating both pre-assessment data and secondary assessment data (sections 4.3 Endpoint Data Collection and 6.2 Secondary Assessment).

### 3.3. Endpoint Data Collection

The enterprise should perform ongoing collection of basic endpoint information such as operating system and version information, and an installed software inventory. This information is collected for general system monitoring as well as its potential use in activities such as vulnerability assessment.

Some examples of basic information to collect about endpoints in this pre-assessment process could include:

- Endpoint type - traditional (e.g., workstation, server, etc.) network infrastructure (e.g., switches, routers, etc.), mobile (e.g., cell phones, tablets, laptops, etc.), and constrained (e.g., industrial control systems, Internet of Things, etc.)

- Hardware version/firmware - e.g., BIOS version, firmware revision, etc.

- Operating system - e.g., Windows, Linux, Mac OS, Android

- Operating system attributes - e.g., version, patch level, service pack level, internationalized or localized version, etc.

- Installed software inventory - Would include the software names and versions and possibly other high-level attributes. Could be used to quickly determine endpoint applicability when new vulnerability description data arrives.

Some additional and more advanced information to collect from endpoints in this pre-assessment process could include:

- Open ports and enabled services - This would include applications listening for incoming connections on open ports as well as services that are starting, running, suspended, or enabled to run pending some event.

- Operating system optional component inventory - some OS' have optional components that can be installed which may not show up as separate pieces of software (e.g., web and ftp servers, demo web pages, shared libraries, etc.). Note that this could also occur within third-party applications as well.

- Endpoint location - physical location (e.g., department, room, Global Positioning System (GPS), etc.), logical location (e.g., what network infrastructure endpoints (e.g. switches, wireless access point, etc.) an endpoint is connected to, etc.

- Purpose - describes how the endpoint is used within the enterprise (e.g., end-user system, database server, public web server, etc.)

- Criticality - enterprise defined rating (possibly a score) that helps determine the criticality of the endpoint. If this endpoint is attacked or lost, what is the impact to the overall enterprise?

It is important to note that some of these attributes may exist natively on the endpoint whereas other attributes may be assigned by a human, computed, or derived from other data and may or may not be available for collection on the endpoint.

Furthermore, the possibility should be left open for enterprises to define their own custom queries and algorithms to gather and derive enterprise-specific attributes that are deemed of interest to regular enterprise operations.

In addition to collecting these attributes, metadata about the attributes should also be collected which could include:

Data origin - where the data originated from

Data source - what provided the data

Date and time of collection - when the data was collected

#### 3.3.1. SACM Use Case Alignment

This sub-step aligns with the Data Publication building block capability because this section involves storage of endpoint attributes within an enterprise Repository. This sub-step also aligns with the Endpoint Characterization and Endpoint Target Identification building block capabilities because it further characterizes the endpoint through automated and possibly manual means. There is direct alignment with the Endpoint Component Inventory, Posture Attribute Identification, and Posture Attribute Value Collection building block capabilities since the purpose of this sub-step is to perform an initial inventory of the endpoint and collect basic attributes and their values. Last, there is alignment with the Collection Guidance Acquisition building block capabilities as the inventory and collection of endpoint attributes would be directed by some type of enterprise or third-party guidance.

### 3.4. Implementation Examples

Within the SACM Architecture, the Internal and External Collector components could be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

The SWID Message and Attributes for IF-M standard defines collection and validation of software identities using the ISO Software Identification Tag Standard. Using this standard, the identity of all installed software including the endpoint operating system, could be collected and used for later assessment.

The OVAL Definitions Model provides a data model that can be used to specify what posture attributes to collect as well as their expected values which can be used to drive an assessment.

The OVAL System Characteristics Model can be used to capture information about an endpoint. The model is specifically suited to expressing OS information, endpoint identification information (such as IP and MAC addresses), and other endpoint metadata.

## 4. Vulnerability Description Data

The next step in the Vulnerability Assessment scenario begins after vulnerability description data has been received and processed into a form that can be used in the assessment of the enterprise. As a part of the enterprise process for managing vulnerability description data, the enterprise should store all received and processed vulnerability description data in a Repository. The stored vulnerability description data can be used and compared with later vulnerability description data for the purpose of duplicate detection and in some cases, guidance on how to handle similar issues.

All vulnerability description data should be assigned an internal tracking ID by the enterprise as a first step as this helps compensate for the fact that incoming vulnerability description data might not have a global identifier when it is received, and might never be assigned one.

High-level vulnerability description data metadata to store would include:

- Ingest date and time - the date and time that the vulnerability description data was received by the enterprise.

- Date and time of vulnerability description data release (i.e., publication or disclosure date and time) - Some older vulnerability description data may be ingested long after publication. This can be useful when reviewing historical enterprise information to (potentially) identify the period when a particular endpoint was first assessed as vulnerable. Sometimes this information will help to differentiate between similar vulnerability description data.

- Version - the version or iteration of the vulnerability description data according to the author, if applicable.

- External Vulnerability Description Data ID(s) (if applicable) - any external or third-party IDs assigned to the vulnerability description data should be tracked. There could be multiple IDs in some cases (e.g., vendor bug id, global ID, discoverer's local ID, third-party vulnerability database ID, etc.).

- Severity Score (if available) - these may be useful for later mitigation prioritization.

In addition to the described metadata, the raw or original vulnerability description data would be stored along with the specific information extracted from it that is to be used in the applicability and assessment process.

### 4.1. SACM Use Case Alignment

This step aligns with the Data Publication and Data Retrieval building block capabilities because this section details storage of vulnerability description data within an enterprise Repository and later retrieval of the same.

### 4.2. Implementation Examples

The Common Vulnerability Reporting Framework (CVRF) is an XML-based language that attempts to standardize the creation of vulnerability report documentation. Using CVRF, the enterprise could create automated tools based on the standardized schema which would obtain the needed and relevant information useful for later assessments and assessment results.

## 5. Endpoint Applicability and Assessment

When new vulnerability description data is received by the enterprise, applicable enterprise endpoints must be identified and assessed. Endpoints are first examined using the already obtained pre-assessment data. If this is not sufficient to determine endpoint

applicability, a secondary data collection for additional data and attributes may be performed to determine status with regard to the vulnerability description data.

## 5.1. Applicability

The applicability of an endpoint and its vulnerability status can, in many cases, be determined entirely by the existence of a particular version of installed software on the endpoint. This data may have been collected in the pre-assessment data collection. If the applicability and vulnerability status of an endpoint can be determined entirely by the pre-collected data attribute set, no further data collection is required.

Other cases may require specific data (i.e., file system attributes, specific configuration parameters, etc.) to be collected for the assessment of a particular vulnerability description data. In these cases, a secondary, targeted vulnerability assessment is required. Administrators may want to evaluate applicability to the vulnerability description data iteratively. Specifically, the process would compare against pre-collected data first (easy to do and the data is stored in a Repository), and then if needed, query endpoints that are not already excluded from applicability for additional required data. (I.e., A "fast-fail" model). To do this, the criteria for determining applicability must be separable, so that some conclusions can be drawn based on the possession of partial data.

### 5.1.1. SACM Use Case Alignment

This sub-step aligns with the Data Retrieval, Data Query, and Posture Attribute Value Query building block capabilities because, in this sub-step, the process is attempting to determine the vulnerability status of the endpoint using the data that has previously been collected.

## 5.2. Secondary Assessment

If the applicability and vulnerability status of an endpoint cannot be determined by the pre-assessment data collection, a secondary and targeted assessment of the endpoint will be required. A secondary assessment may also be required in the case that data on-hand (either from pre-assessment or from prior secondary assessments) is stale or out-of-date.

The following data types and attributes are examples of what might be required in the case of a secondary and targeted assessment:

- Specific files and attributes - i.e., file name, versions, size, write date, modified date, checksum, etc. Some vulnerabilities may only be distinguishable through the presence or absence of specific files or their attributes.

- Shared libraries - Some vulnerabilities will affect many products across multiple vendors. In these cases the vulnerability may apply to a shared library. Under these circumstances, product versions may be less helpful than looking for the presence of one or more specific files and their attributes.

- Other software configuration information (if applicable) - e.g., Microsoft Windows registry queries, Apple configuration profiles, GConf, Proc filesystem, text configuration files and their parameters, and the installation paths. Sometimes vulnerabilities only affect certain software configurations and in some cases these are not the default configurations. Certain configuration attributes can be used to determine the current configuration state.

Note that the secondary assessment described here does not need to be a pull assessment that is initiated by the server. The secondary assessment could also be part of a push to the server when the endpoint detects a change to a vulnerability assessment baseline.

#### 5.2.1. SACM Use Case Alignment

This sub-step aligns with the Data Publication building block capability because this section details storage of endpoint attributes within an enterprise Repository. The sub-step also aligns with the Collection Guidance Acquisition building block capability since the vulnerability description data (guidance) drives the collection of additional endpoint attributes.

This sub-step aligns with the Endpoint Characterization (both manual and automated) and Endpoint Target Identification building block capabilities because it could further characterize the endpoint through automated and possibly manual means. There is direct alignment with the Endpoint Component Inventory, Posture Attribute Identification, and Posture Attribute Value Collection building block capabilities since the purpose of this sub-step is to perform additional and more specific component inventories and collections of endpoint attributes and their values.

### 5.3. Implementation Examples

Within the SACM Architecture, the assessment task would be handled by the Evaluator component. If pre-assessment data is used, this would be stored on and obtained from a Data Store component.

Within the SACM Architecture, the Internal and External Collector components could be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

The SWID Message and Attributes for IF-M standard defines collection and validation of software identities using the ISO Software Identification Tag Standard. Using this standard, all installed software including the endpoint operating system could be collected and stored for later assessment.

The OVAL Definitions Model provides a data model that can be used to specify what posture attributes to collect as well as their expected values which can be used to drive an assessment.

The OVAL System Characteristics Model can be used to capture information about an endpoint. The model is specifically suited to expressing OS information, endpoint identification information (such as IP and MAC addresses), and other endpoint metadata.

The SACM Internal and External Attribute Collector components can be used to allow enterprises to collect posture attributes that demonstrate compliance with enterprise policy. Endpoints can be required to provide posture attributes, which may include identification attributes to enable persistent communications.

## 6. Assessment Results

Assessment results present the results of an assessment, along with sufficient context so a human or machine can make the appropriate response. This context might include a description of the issue provided by the vulnerability description data, the endpoint attributes that indicate applicability, or other information needed to respond to the results of the assessment. Data in this step is stored for auditing and forensic purposes.

The following details are important to track in assessment results. Note that information may be "included" by providing pointers to other records stored in a Repository (e.g., vulnerability description data, endpoint data, etc.).

- Date and time of assessment - The date and time that the assessment was performed. To understand when the data was compared against the vulnerability description data and what conclusions were drawn.

- Data collection/attribute age - The age of the data used in the assessment to make the endpoint status determination.

- Endpoint ID - The endpoint itself must be identified for tracking results over time.

- Vulnerability description data ID(s) - May include both the internally defined ID as well as one or more externally defined IDs if they exist. The internally assigned ID allows linkage to the correct vulnerability description data. If available, external IDs provide a "pivot point" to additional external information.

- Vulnerable software product(s) - Identifies the software products on the endpoint that resulted in the endpoint being declared applicable. Since some vulnerability description data identify vulnerabilities in multiple products, this will help identify the specific product (or products) found to be vulnerable in the endpoint assessment.

- Endpoint vulnerability status - The endpoint status based on the vulnerability description data. Does the vulnerability exist on the endpoint?

- Vulnerability description - Not needed for automated assessment but probably should be included for human review. The reason for inclusion is to support the human user understanding of the vulnerability assessment results within the application front-end or interface.

- Vulnerability remediation - Similar to the above, remediation or vendor patch information would be useful for a human response. In many cases, this information may be a part of the description information described above. Note that patch information may change over time due to supercession of the vendor patches.

### 6.1. SACM Use Case Alignment

This step aligns with the Data Publication and Data Retrieval building block capabilities because this section details storage of vulnerability assessment results within an enterprise Repository and later retrieval of the same.

### 6.2. Implementation Examples

The OVAL Results Model provides a data model to encode the results of the assessment, which could then be stored in a Repository and later accessed. The assessment results described in this scenario could be stored and later accessed using the OVAL Results Model. Note that the use of the OVAL Results Model for sharing results is not recommended per section 7.3 of the OVAL and the SACM Information Model [draft-hansbury-sacm-oval-info-model-mapping-01].

Within the SACM Architecture, the generation of the assessment results would occur in the Report Generator component. Those results might then be moved to a Data Store component for later sharing and retrieval as defined by SACM.

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

This document provides a core narrative that walks through an automated enterprise vulnerability assessment scenario and is aligned with SACM "Endpoint Security Posture Assessment: Enterprise Use Cases" [RFC7632]. As a result, the security considerations for [RFC7632] apply to this document. Furthermore, the vulnerability description data may provide attackers with useful information such as what software an enterprise is running on their endpoints. As a result, organizations should properly protect the vulnerability description data it ingests.***TODO IS THIS COVERED BY RFC7632???***

## 9. Informative References

[charter-ietf-sacm-01]
Security Automation and Continuous Monitoring, "Charter, Version 1.0", July 2013.

[critical-controls]
Council on CyberSecurity, "Critical Security Controls, Version 5.1".

[draft-hansbury-sacm-oval-info-model-mapping-01]
Security Automation and Continuous Monitoring, "OVAL and the SACM Information Model", November 2015.

[I-D.ietf-sacm-requirements]
Cam-Winget, N. and L. Lorenzin, "Secure Automation and Continuous Monitoring (SACM) Requirements", draft-ietf-sacm-requirements-11 (work in progress), November 2015.

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

## Appendix A. Change Log

### A.1. Changes in Revision 01

Clarification of the vulnerability description data IDs in sections 4 and 6.

Added "vulnerability remediation" to the Assessment Results and Data Attribute Table and Definitions sections.

Added Implementation Examples to Endpoint Identification and Initial (Pre-Assessment) Data Collection, Vulnerability Description Data, Endpoint Applicability and Assessment, and Assessment Results sections.

Added an example to vulnerability description data in the scope section.

Added a sentence to clarify vulnerability description data definition in the scope section.

Added data repository example for long-term storage scope item.

Added sentence to direct reader to examples of basic system information in endpoint identification section.

Split the examples of information to collect in the pre-assessment collection section into a basic and advanced list.

Added examples of data stored in the repository in the Assessment Results section.

Added sentence for human-assigned attributes in the Future Work section.

Replaced "vulnerability report" to "vulnerability description data" because the term report was causing confusion. Similarly, replaced "assessment report" with "assessment results".

Replaced "Configuration Management Database (CMDB)" with "Repository" which is SACM's term for a data store.

Replaced endpoint "Role" with "Purpose" because "Role" is already defined in SACM. Also, removed "Function" because it too is already defined in SACM.

Clarified that the document does not try to define a normalized data format for vulnerability description data although it does not preclude the creation of such a format.

Included additional examples of software configuration information.

Clarified the section around endpoint identification to make it clear designation attributes used to correlate and identify endoints are both persistent and unique. Furthermore, text was added to explain how the persistency of attributes may vary. This was based on knowledge gained from the Endpoint ID Design Team.

Updated the Security Considerations section to mention those described in [RFC7632].

Removed text around Bring Your Own Device (BYOD). While important, BYOD just adds complexity to this initial draft. BYOD should be addressed in a later revision.

Merged the list of "basic endpoint information" and the list of "human-assigned endpoint attributes" as both represent data we want to collect about an endpoint. Whether or not that data is natively available on the endpoint for collection or assigned by a human, computed, or derived from other data which may or may not be available on the endpoint for collection seems arbitrary. With this scenario, we primarily care about expressing information needs rather than how the information is collected or from where.

## Appendix B. Continuous Vulnerability Assessment

It is not sufficient to perform a single assessment when vulnerability description data is published without any further checking. Doing so does not address the possibility that the reported vulnerability might be introduced to the enterprise environment after the intial assessment completes. For example, new endpoints can be introduced to the environment which have old software or are not up-to-date with patches. Another example is where unauthorized or obsolete software is installed on an existing endpoint by enterprise users after vulnerability description data and initial assessment has taken place. Moreover, enterprises might not wish to, or be able to, assess all vulnerability description data

immediately when they come in. Conflicts with other critical activities or limited resources might mean that some alerts, especially those that the enterprise deems as "low priority", are not used to guide enterprise assessments until sometime after the initial receipt.

The scenario above describes a single assessment of endpoints. However, it does not make any assumptions as to when this assessment occurs relative to the original receipt of the vulnerability description data that led to this assessment. The assessment could immediately follow ingest of the vulnerability description data, could be delayed, or the assessment might represent a reassessment of some vulnerability description data against which endpoints had previously been assessed. Moreover, the scenario incorporates long-term storage of collected data, vulnerability description data, and assessment results in order to facilitate meaningful and ongoing reassessment.

## Appendix C. Priority

Priorities associated with the vulnerability description data, assessment results, and any remedy is important, but is treated as a separate challenge and, as such, has not been integrated into the description of this scenario. Nevertheless, it is important to point out and describe the use of priorities in the overall vulnerability description data scenario as they separable issues with their own sets of requirements.

Priority in regard to vulnerability description data, can be viewed in a couple of different ways within an enterprise. The assessment prioritization involves prioritization of the vulnerability description data assessment process. This determines what vulnerability description data is assessed, and in what order it is assessed in. For instance, a vulnerability affecting an operating system or application used throughout the enterprise would likely be prioritized higher than a vulnerability in an application which is used only on a few, low-criticality endpoints.

The prioritization of remedies relates to the enterprise remediation and mitigation process based on the discovered vulnerabilities. Once an assessment has been performed and applicable endpoints identified, enterprise vulnerability managers must determine where to focus their efforts to apply appropriate remedies. For example, a vulnerability that is easily exploitable and which can allow arbitrary code execution might be remedied before a vulnerability that is more difficult to exploit or which just degrades performance.

Some vulnerability description data include severities and/or other information that places the vulnerability in context. This information can be used in both of the priority types discussed above. In other cases, enterprise administrators may need to prioritize based only on what they know about their enterprise and the description provided in the vulnerability description data.

Examples of data attributes specific to priority of assessments and/or remedies include (but not limited to) the following:

- Enterprise - defined purpose of the device, criticality of the device, exposure of the device, etc.
- Severity attributes - A rating or score that attempts to provide the level of severity or criticality associated with a given vulnerability.
- Cyber threat intelligence - information such as tactics, techniques, and procedures of threat actors, indicators of compromise, incidents, courses of action, etc. that help the enterprise understand relevant threats and how to detect, mitigate, or respond to them.

## Appendix D. Data Attribute Table and Definitions

### D.1. Table

The following table maps all major data attributes against each major process where they are used.

| | vulnerability description data | Endpoint Identification and Initial (Pre-Assessment) Data Collection | Endpoint Applicability and Assessment | Assessment Results |
|---|---|---|---|---|
| *Endpoint* | | | | |
| Collection date/time | | X | X | |
| Endpoint type | | X | X | |
| Hardware ver | X | X | X | |

| sion/firmware | | | | | |
|---|---|---|---|---|---|
| Operating system | X | X | X | | |
| Operating system attributes (e.g., version, service pack level, edition, etc.) | X | X | X | | |
| Installed software name | X | X | X | X | |
| Installed software attributes (e.g., version, patch level, install path, etc.) | X | X | X | X | |
| Open ports/services | X | X | X | | |
| Operating system optional component inventory | X | X | X | | |
| Location | | X | | X | |
| Purpose | | X | | X | |
| Criticality | | X | | X | |
| File system attributes (e.g., versions, | X | | X | | |

| | | | | | |
|---|---|---|---|---|---|
| size, write date, modified date, checksum, etc.) | | | | | |
| Shared libraries | X | | X | | |
| Other software configuration information | X | | X | | |
| *External vulnerability description data* | | | | | |
| Ingest Date | X | | X | | |
| Date of Release | X | | X | | |
| Version | X | | X | | |
| External vuln ID | X | | X | X | |
| Severity Score | | | | X | |
| *Assessment Results* | | | | | |
| Date of assessment | | | X | X | |
| Date of data collection | | X | X | X | |
| Endpoint identification and/or locally assigned ID | | X | X | X | |

| | | | | | |
|---|---|---|---|---|---|
| Vulnerable software product(s) | X | X | X | X | |
| Endpoint vulnerability status | | | X | X | |
| Vulnerability description | X | | | X | |
| Vulnerability rememdiation | X | | | X | |

Table 1: Vulnerability Assessment Attributes

## D.2. Definitions

Endpoint

- Collection date/time - the date and time of data collection
- Endpoint type - the device type of the endpoint (e.g., standard computer, printer, router, mobile device, tablet, etc.)
- Hardware version/firmware - the hardware or firmware version if applicable (e.g., BIOS version, firmware revision, etc.)
- Operating system - Operating system name
- Operating system attributes - Operating system high-level attributes (e.g., version, service pack level, edition, etc.). Would not include configuration details.
- Installed software name - List of all installed software packages (i.e., software inventory). May or may not include software installed by the operating system.
- Installed software attributes - Software high-level attributes (e.g., version, patch level, install path, etc.). Would not include configuration details.
- Open ports/enabled services - Listening network ports (e.g., TCP, UDP, etc.) as well as services that are starting, running, suspended, or enabled to run pending some event.

- Operating system optional component inventory - Operating system specific components and software (when NOT already included in the general software inventory)

- Location - The physical location of an enterprise endpoint (e.g., department, room, etc.)

- Purpose - describes how the endpoint is used within the enterprise (e.g., end user system, database server, public web server, etc.)

- Criticality - An enterprise-defined rating (possibly a score) that helps determine the criticality of the endpoint. If this endpoint is attacked or lost, what is the impact to the overall enterprise?

- File system attributes - Attributes that describe the file or directory (e.g., versions, size, write date, modified date, checksum, etc.)

- Shared libraries - libraries that can be used by and installed with many different software applications. A shared library vulnerability could affect multiple software applications in the same way.

- Other software configuration information - operating system or software application configuration attributes that go beyond that basic information already captured (e.g., Microsoft Windows registry, Apple configuration profiles, GConf, Proc filesystem, text configuration files and their parameters, and the installation paths.)

External vulnerability description data

- Ingest Date - the date that the vulnerability description data was received by the enterprise.

- Date of Release - publication or disclosure date of the vulnerability description data

- Version - the version or iteration of the vulnerability description data according to the author, if applicable.

- External vuln ID - external or third-party IDs assigned to the vulnerability description data. Could be multiple IDs in some cases (e.g., vendor bug id, global ID, discoverer’s local ID, third-party vulnerability database ID, etc.).

- Severity Score - the severity of the vulnerability description data according to the vulnerability description data author, if applicable.

Assessment Results

- Date of assessment - The date that the assessment was performed against an endpoint.
- Date of data collection - The age of the data used in the assessment to make the endpoint status determination.
- Endpoint identification and/or locally assigned ID - The ID assigned to the enterprise endpoint. Must be assigned for tracking results over time.
- Vulnerable software product(s) - The vulnerable software products identified as being installed on the endpoint.
- Endpoint vulnerability status - Overall vulnerability status of the enterprise endpoint (i.e., Pass or Fail)
- Vulnerability description - A human-consumable description of a vulnerability. Supports the human user understanding of the vulnerability assessment results within an application front-end or user interface.
- Vulnerability remediation - The fix, workaround, or patch information for a vulnerability. This information may be a part of the vulnerability description described previously. Note that this information can change over time due to vendor patch supercession.

## Appendix E. Alignment with Other Existing Works

### E.1. Critical Security Controls

The Council on CyberSecurity's Critical Security Controls [critical-controls] includes security controls for a number of use scenarios, some of which are covered in this document. This section documents the alignment between the Council's controls and the relevant elements of the scenario.

#### E.1.1. Continuous Vulnerability Assessment

"CSC 4: Continuous Vulnerability Assessment and Remediation," which is described by the Council on CyberSecurity as "Continuously acquire, assess, and take action on new information in order to

identify vulnerabilities, remediate, and minimize the window of opportunity for attackers." The scenario described in this document is aligned with CSC 4 in multiple ways:

CSC 4-1 applies to this scenario in that it calls for running regular, automated scanning to deliver prioritized lists of vulnerabilities with which to respond. The scenario described in this document is intended to be executed on a continuous basis, and the priorities of both vulnerability description data and the remedy of vulnerabilities are discussed in the Priority section earlier in this document.

This scenario assumes that the enterprise already has a source for vulnerability description data as described in CSC 4-4.

Both CSC 4-2 and 4-7 are made possible by writing information to a Repository since this makes previously collected data available for later analysis.

While this scenario does not go into the details of how prioritization would be calculated or applied, it does touch on some of the important ways in which prioritization would impact the endpoint assessment process in the Priority section. As such, the Priority section aligns with CSC 4-10, which deals with vulnerability priority. Vulnerability priority in this scenario is discussed in terms of the vulnerability description data priority during receipt, as well as the vulnerability priority with regards to remedies.

The described scenario does not address the details of applying a remedy based on assessment results. As such, CSC 4-5, 4-8, and 4-9, which all deal with mitigations and patching, are out of scope for this work. Similarly, CSC 4-3 prescribes performing scans in authenticated mode and CSC 4-6 prescribes monitoring logs. This scenario does not get into the means by which data is collected, focusing on "what" to collect rather than "how", and as such does not have corresponding sections, although the procedures described are not incompatible with either of these controls.

The CSC 4 System Entity Relationship diagram and numbered steps directly align with the scenario described in this document with the exception of step 7 (patch response). Steps 1 -6 in CSC 4 describe the overall process for vulnerability management starting with obtaining the vulnerability description data from the source in Step 1, to producing assessment results in step 6.

### E.1.2. Hardware and Software Inventories

This scenario is also aligned with, and describes a process for, collecting and maintaining hardware and software inventories, which are covered by the Council on CyberSecurity CSC 1 "Inventory of Authorized and Unauthorized Devices" and CSC 2 "Inventory of Authorized and Unauthorized Software." This scenario documents a process that is specific to collecting and maintaining hardware and software data attributes for vulnerability assessment purposes, but the collection of the hardware attributes and software inventory documented in the Endpoint Data Collection section that follows can also be used for the purpose of implementing authorized and unauthorized hardware and software management processes (e.g., scanning tools looking for unauthorized software). Moreover, the ability to accurately identify endpoints and, to a lesser degree, applications is integral to effective endpoint data collection and vulnerability management.

The Endpoint Data Collection section does not have coverage for the specific details described in CSC 1 and 2 as they are different processes and would be out-of-scope of this scenario, but the section does provide the data necessary to support the controls.

The Endpoint Identification and Endpoint Data Collection sections within this scenario align with CSC 1-1 and 1-4 by identifying enterprise endpoints and collecting their hardware and network attributes. The Endpoint Data Collection section aligns with and supports CSC 2-3 and 2-4 by defining a software inventory process and a method of obtaining operating system and file system attributes. The rest of the items from CSC 1 and 2 deal with implementation details and would be out-of-scope for this document.

CSC 2-9 describes the use of a software ID tag in XML format. SWID tags (https://en.wikipedia.org/wiki/ISO/IEC_19770) would also be a possible implementation for the Endpoint Data Collection section described in this scenario.

## Appendix F. SACM Usage Scenarios

The SACM "Endpoint Security Posture Assessment: Enterprise Use Cases" document ([RFC7632]) defines multiple usage scenarios that are meant to provide examples of implementing the use cases and building block capabilities. Below is a brief summary of some of these usage scenarios and how this document aligns and/or adds additional value to the identified usage scenarios.

- Automated Checklist Verification (2.2.2) - "An enterprise operates a heterogeneous IT environment. 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 ensure adequate coverage of all IT assets." The usage scenario, as defined in the RFC, is targeted at the checklist level and can be interpreted as being specific to endpoint configuration. There is mention of patch assessment and vulnerability mitigation, but the usage scenario could be expanded upon by including vulnerability verification. Replacing the idea of a checklist in the SACM usage scenario with vulnerability would allow the usage scenario to align almost exactly with the scenario described in this document. Instead of collecting automatable security configuration checklists, the enterprise would collect automatable vulnerability description data available from the vendor as described or possibly from other interested third-parties.

- Detection of Posture Deviations (2.2.3) - "An enterprise has established secure configuration baselines for each different type of endpoint within their IT environment. When an endpoint connects to the network, the appropriate baseline configuration is communicated to the endpoint. 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. When the endpoint detects a posture change, an alert is generated identifying the specific changes in posture." This usage scenario would support the concept of endpoints signaling or alerting the enterprise to changes in the posture relates to endpoint vulnerabilities in the same way that it would for configurations. Replacing the idea of a checklist with vulnerability description data allows the SACM usage scenario and the scenario described in this document to align in their objectives.

- Asynchronous Compliance/Vulnerability Assessment at Ice Station Zebra (2.2.5) - "An isolated arctic IT environment that is separated from the main university network. The only network communications are via an intermittent, low-speed, high-latency, high-cost satellite link. Remote network admins will need to show continued 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." This SACM usage scenario describes vulnerability assessment and aligns well with the vulnerability scenario described in this document. The endpoint assets are identified and associated data is published in a Repository. Vulnerability description data is collected and saved in a Repository as it is released. The vulnerability description data is queued for later assessment, then the

assessment results and vulnerability description data are stored after assessment. The only real difference in this SACM usage scenario is the timing of the assessments. The scenario described within this document would have no problems adjusting to the timing of this SACM usage scenario or anything similar.

## Appendix G. SACM Requirements and Charter - Future Work

In the course authoring this document, some additional considerations for possible future work were noted. The following points were taken from the SACM Requirements [I-D.ietf-sacm-requirements], SACM Charter [charter-ietf-sacm-01], and SACM Use Cases ([RFC7632]) documents and represent work that may be necessary to support the tasks or goals of SACM going forward.

- The SACM requirements mentions "Result Reporting" with applications but no detail around what the assessment results data set should include. In the case of vulnerability assessment results, context is important and details beyond just a Pass or Fail result are needed in order to take action. A good example of this might be the Priority of the vulnerability itself and how many systems it affects within the enterprise. With this in mind, it might be worthwhile to investigate a minimum data set or schema for assessment results. The concern here is with vulnerability description data, but this could apply to other enterprise processes as well.

- The "Human-assigned endpoint attributes" mentioned previously in this scenario are touched on in the SACM use cases, but the topic could probably be explored in much more depth. Enterprise policy and behaviors could be greatly influenced by endpoint attributes such as locations, how the endpoint is used, and criticality. When and how these data attributes are collected, as well as what the minimum or common set might look like, would be good topics for future related SACM work. In addition, the storage of these attributes could be central (stored in a data repository) or they could be assigned and stored on the endpoints themselves.

## Authors' Addresses

Christopher Coffin
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: ccoffin@mitre.org

Brant Cheikes
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: bcheikes@mitre.org

Charles Schmidt
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
USA

Email: cmschmidt@mitre.org

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

Email: dhaynes@mitre.org

Jessica Fitzgerald-McKay
Department of Defense
9800 Savage Road
Ft. Meade, Maryland
USA

Email: jmfitz2@nsa.gov

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

Email: david.waltermire@nist.gov