SCAP v2 Data Collection Architecture

August 18, 2020

This document describes the SCAP v2 architecture that supports collection of data from the enterprise and delivery of this information to enterprise components responsible for interpreting and acting on it.

Contents

Use of this document

The document provides a high-level technical description of the SCAP data collection architecture as of the time this document was written. It also documents open questions that remain to be addressed. It does not provide syntactic details regarding data elements and interfaces and, as such, it does not provide sufficient detail to allow two parties to create mutually interoperable implementations of architecture elements. However, it is expected that an implementer would be able to use this document to create a proof-of-concept that conformed to this design without needing to devise solutions for any significant design elements.

This document is written to support design review and, ideally, development of simplistic prototypes that can prove out elements of the design prior to final standardization.

Overview of SCAP v2 Data Collection Architecture

This section provides a high-level overview of the SCAP v2 Data Collection architecture. Subsequent sections go into greater detail on all components.

Terminology

Enterprise – The collection of network-connected, managed assets of an organization as well as the network infrastructure that connects these assets.

Managed asset – A device or other networked computing element that an organization has the ability to directly assess. In many cases, this means that the organization has sufficient access to allow for the installation of agents, as well as sufficient access for those agents to inspect the asset's system settings. Note, however, that some assets might be constrained or otherwise locked down by their vendor (e.g., some IoT devices), limiting on-asset collection capabilities. These assets remain "managed" for the purposes of this discussion if the enterprise has control over their deployment and configuration.

Asset information – Information collected from and/or about an asset. This includes both information directly collected from an asset as well as information collected from external monitors about an asset. Asset information includes compliance information (determinations as to whether certain conditions are met for a given asset) as well as state information (information collected about the state of an enterprise asset). From a technical perspective there is little to distinguish compliance and state information as both might represent the results of evaluations - the only difference is whether the final evaluation leads

to a determination that the asset is in a "good" or "bad" state. As such, this document will generally not distinguish between the two types of asset information.

Scope

The SCAP v2 Data Collection architecture focuses on the capabilities needed to:

- Task the architecture with collection of asset information from/about a managed asset.
- Store collected asset information so that it may be used to quickly respond to future queries without engaging with an asset.
- Compiling and delivering reports containing requested asset information.

The following activities are of importance to the SCAP v2 architecture, but are considered outside the scope of the data collection architecture described in this document:

- Collection of assessment instructions from external sources.
- Responses to reports containing asset information (e.g., execution of mitigation processes).
- Further distribution and utilization of delivered reports containing asset information after their initial delivery by the data collection architecture.
- The creation, modification, and tailoring of assessment instructions.

Key use cases and capabilities

There are two main use cases driving the design of the SCAP v2 architecture. These are supported by multiple sub-capabilities.

Main Use Case 1: Point-in-time Information Collection

The data collection architecture is tasked to collect specific information about certain enterprise assets. The data collection architecture collects the information, to the extent that its capabilities allow, and then compiles its findings into a report that is returned to the requester. At this point the action is complete – the data collection architecture will retain the information it collected for a period of time so it can be used to support future requests, but otherwise has no further responsibilities regarding the request.

This use case reflects current SCAP assessment practices.

Main Use Case 2: Ongoing Monitoring Against a Baseline

The data collection architecture is tasked with collecting information about certain enterprise assets, just as in use case 1, but then monitors those assets for changes in the collected information. If changes in the reported information are detected, the tasking party is informed of these changes. In this way, data collection architecture provides an ongoing picture of the changing state of the enterprise.

Sub-Capabilities

To facilitate the proper functioning of these use cases, the data collection architecture supports a set of sub-capabilities.

Subscription management – The architecture uses subscriptions to ensure that updates regarding enterprise information are delivered to appropriate parties.

Liveness monitoring – Because failure of components can impact the architecture's ability to meet use cases, the architecture monitors components and connections for liveness. This allows failure alerts to

be raised to the appropriate parties and allows the architecture to distinguish between a lack of updates caused by component failure and lack of updates because there are no changes to report.

Storage and query of asset information – Collected information can be stored by the data collection architecture. This data store can be queried during new assessments to avoid needing to reach out to assets, assuming the stored data is sufficiently recent. This store can potentially also be used to provide long-term tracking of asset information that can support forensic and other historical analysis.

Information collection – The data collection architecture supports the gathering of information from and about a range of enterprise assets. The architecture allows this collection capability to be easily extensible to accommodate new capabilities and asset types.

Information normalization – The data collection architecture supports normalization of collected data. This ensures that key metadata is always present (timestamps, identity of asset, identity of collecting agent, etc.). It also ensures that stored and reported data is always in a uniform format that can be used by any tool compatible with the SCAP data collection architecture.

Applicability mapping to assets – Different assessment instructions apply to different assets. For example, an assessment of the configuration of a Windows machine will include different checks than an assessment of a Linux machine. Given a set of assessment instructions, the data collection architecture will be able to determine which enterprise assets are applicable so that assessment instructions can be directed appropriately.

Assessment scope tracking – As noted above, different assets require different assessment capabilities. Once a given set of assessment instructions has been mapped to assets, the data collection architecture needs to identify the components of the architecture than can assess the appropriate assets that are applicable to the assessment instructions.

Assessment capability tracking – The data collection architecture is designed to support an extensible set of assessment capabilities. As a result, some assessment instructions might need assessment capabilities that have not been built into the data collection architecture. The data collection architecture will understand what capabilities it is able to employ and utilize them, and also recognize when assessment instructions cannot be mapped to assessment capabilities and note this capability gap in returned results.

Design Requirements

In the course of designing the SCAP v2 data collection architecture, the team identified some key requirements and design goals that the architecture needs to support.

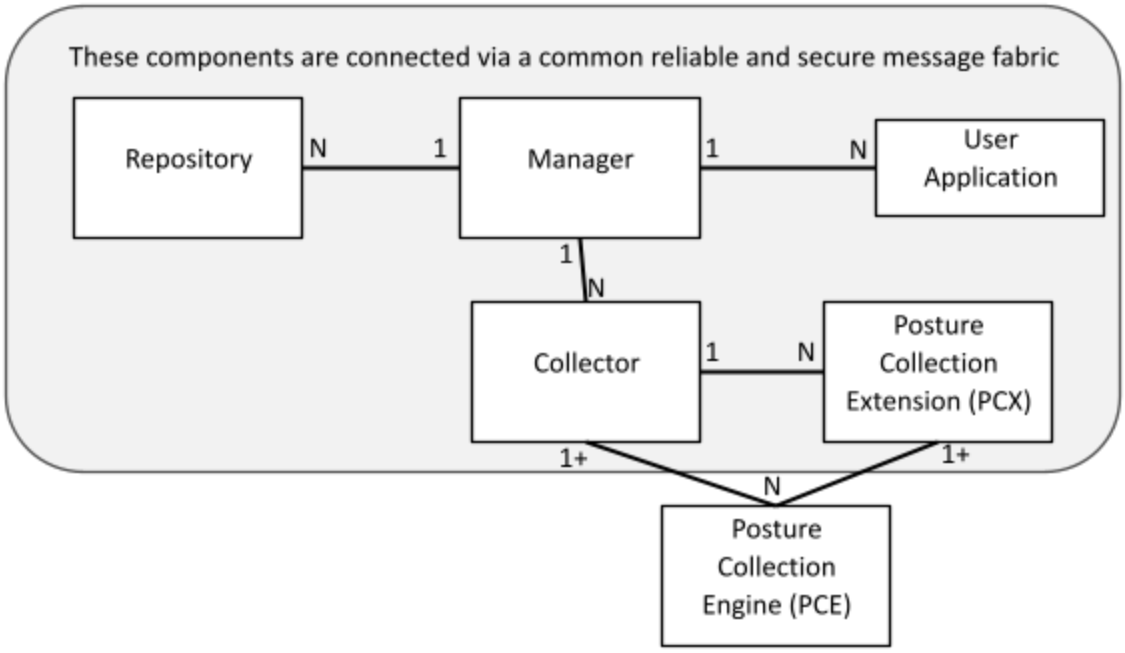
- Don't lose SCAP v1 capabilities – There is a large community utilizing SCAP v1 as part of their security solution. All the capabilities that they depend upon in SCAP v1 should be supported in SCAP v2.
- Support integration of current capabilities – Most organizations already have some security tools. Adoption of SCAP v2 will be slowed if this adoption requires removal and replacement of these old tools. The SCAP v2 data collection architecture should, to the greatest extent possible, be able to overlay and incorporate existing tools rather than requiring their replacement.
- Be transparent in all operations – Enterprises depend upon reliable understanding of their security posture. As such, there can be no ambiguity as to what is being conveyed. This includes

making it clear how old all reported assessment results are and how they were gathered. It also means that absence of information needs to have unambiguous meaning (rather than having it be a question as to whether the lack of reporting is intentional or the result of some failure of the reporting system). Finally, this also means that the system needs to be aware of the health of its own operations. Specifically, errors during activities or loss of access to a component need to be recognized, recorded, and reported so that any impacts these situations might have on assessments is clearly identifiable.

- Support extensible capabilities – SCAP has always sought to support extension of scanning capabilities. Moreover, because of the goal to support integration of existing tools, it is probable that proprietary tools will include capabilities outside of standard SCAP scanning practices. The supporting infrastructure of the data collection architecture needs to be able to cleanly and transparently support scanning capabilities and requests for scanning capabilities that extend beyond those standardized in SCAP.
- Expand to support timely monitoring of changes – Point-in-time assessments (such as those performed in most SCAP v1 deployments) have their value, but there is a need for a more real-time understanding of the state of, and changes to, enterprise security posture. Specifically, certain enterprise information needs to be checked on a regular basis, or even in real-time. However, this needs to be done in a way that does not overwhelm data consumers with unnecessary updates and does not overwhelm assets with repeated queries regarding their state.

The SCAP v2 Architecture and Roles

This section presents the data collection architecture, including its roles and interfaces.



The architecture consists of the following roles:

User Applications (Applications) – These represent the various tools and user interfaces that will send instructions to the data collection architecture and receive results therefrom. In effect, the data collection architecture exists solely to support Application activity. The behaviors of Applications will vary considerably and the data collection architecture does not dictate any Application behaviors. However, Applications will need to implement a standardized interface in order to interact with the data collection architecture.

Manager – The Manager is responsible for taking the instructions received from Applications, identifying the components of the data collection architecture needed to service these instructions, and invoking these components with the appropriate information and context. All interactions with the Manager occur using standardized interfaces.

Repository – This represents a persistent storage capability capable of recording previously collected enterprise information. It also provides persistent storage for other information, such as assessment instructions received and the set of known assessment targets. The exact implementation of this capability is not specified – it could be supported by a dedicated database or the data might be distributed across multiple storage mechanisms. This component implements a standardized, front-end interface and must be capable of executing the instructions received over that interface, but the precise mechanisms by which these instructions are carried out are not specified in recognition of the many ways that persistent storage may be achieved. The Repository helps avoid duplicative data gathering by allowing prior collections to be used to respond to new requests (assuming those prior collections are recent enough to meet needs.) The Repository might also be configured to support long-term data retention that can be used for forensic, retrospective, and trending analysis.

Collector – Each Collector is responsible for information collection regarding a specific subset of the enterprise, called its scope. The scope consists of a set of assets that the Collector is capable of assessing. Assets associated with a Collector are considered "bound" to it. Note that just because an asset is bound to a collector does not imply it is available for assessment at any given time – a bound asset could be off or disconnected when a Collector attempts to perform an assessment, but the asset remains "bound" to the Collector. The set of assets bound to each Collector is stored in the Repository. If a new asset becomes bound to a Collector, the Collector is expected to update this information in the Repository. Collector scopes are expected not to overlap with each other and the union of all scopes encompasses all enterprise assets that are capable of being assessed.

Collectors receive tasking instructions from the Manager to collect information about the assets within their scope. To do this, each Collector is responsible for engaging with one or more PCEs (see below). These PCEs are capable of gathering information about the assets in the Collector's scope, and the entire Collector's scope is covered by the collection of PCEs with which it communicates. Since the interfaces with PCEs are not standardized in SCAP, a Collector is responsible for interpreting the instructions received from the Manager and using them to build appropriate instructions to be sent to the PCE to cause the PCE to collect the necessary information. The Collector is responsible for taking the information returned by the PCE and converting it into a standard result format, which may include annotating these results with additional information. Collectors are also responsible for conducting some amount of evaluation of the collected information based on the instructions received from the Manager. Finally, Collectors are responsible for delivering this information (after normalization, annotation, and possible evaluation) back to the Manager.

Posture Collection Extension (PCX) – The goal of the SCAP data collection architecture is to support any form of assessment, which means that a Collector's assessment capabilities need to be readily extensible. This is accomplished through standardization of an interface between Collectors and PCXs. When a Collector receives instructions from a Manager, it is possible that some extensions to the SCAP instructions might not be actionable by the Collector and its attendant PCEs. However, if the Collector is connected to a PCX that does understand those extended instructions, the extended instructions can be handed over to the PCX for processing. A PCX supports most of the same behaviors as a Collector: It is associated with one or more PCEs, it converts SCAP instructions into the appropriate PCE commands, and it normalizes and augments the PCE responses. The PCX only differs from the Collector in that the Collector is invoked directly by the Manager while the PCX is invoked by a Collector. A PCX also has a scope, which will be a subset of the scope of its controlling Collector. In summary, a PCX is a way to add on capabilities to a Collector to allow the SCAP data collection architecture to grow its capabilities.

Posture Collection Engine (PCE) – These are components capable of directly collecting information about enterprise assets. Many PCEs will exist as agents on the assets about which they report, but a PCE could also report on other assets, acting as an external scanner for those assets. Each PCE reports to a single Collector but a Collector may receive information from many PCEs. The SCAP data collection architecture does not specify any requirements governing the behavior of posture collection engines beyond their definitional capability of collecting information about enterprise assets. Similarly, the data collection architecture does not define standards regarding the interfaces used to interact with PCEs or the format of the data they return.

Message Fabric – The Application, Manager, Repository, Collector, and PCX are all connected using a common message fabric. PCEs might be connected to their Collector or PCX via the message fabric, but this is not required. This message fabric provides numerous services. These include:

- Reliable message delivery (including retry of failed delivery and alerting when retry has been exhausted)
- Subscription queues – These are used to ensure reports of changes during monitoring are distributed to interested parties.

Key Supporting Data Sets

The SCAP data collection architecture employs several key data sets in its operation. The syntax and semantics of these will be standardized in the future. This document seeks to identify the key data sets at a high level in order to better clarify how the architecture is intended to operate.

Assessment Instructions

Assessment instructions will be passed from Applications to the Manager. These instructions will guide any queries the Manager makes to the Repository element. In the initial release of the SCAP data collection architecture standard, it will be the case that these instructions will outline specific mechanisms for data collection and thus will be platform-specific. Today's OVAL, as well as many other checking languages, operate at this level. Future versions of the SCAP data collection architecture might allow specification of assessment instructions in a more abstract, platform-agnostic manner.

The Manager will also send assessment instructions to relevant Collectors, which in turn will use them to task PCE and PCX elements. The specific format of instructions sent from Collectors or PCXs to their PCEs is not specified and might be different from the assessment instructions received by the Collector/PCX.

An important aspect of the Collector/PCX role is to take assessment instructions received from the Manager, which will be expressed in one of a small set of standardized formats, and convert that into instructions that can be understood by their PCE(s). As such, while Collectors will only accept a small number of formats, a PCE can be integrated into the architecture regardless of the format of the data it accepts. (In cases where supported PCE formats are known by the Application, this process might be facilitated by having one of the supported Collector formats be a "wrapper" that can take any form of instruction and having the Collector "translate" that input by stripping the wrapper and passing along the contained format to a PCE that understands it.)

Bound Asset Lists

Each Collector is bound to a set of assets it is capable of assessing (at least in some ways). The total set of bound assets is the set of assets the SCAP architecture is capable of assessing. This information is needed for two, related purposes. First, when making applicability evaluations, it is necessary to know which potential assessment targets are applicable, not-applicable, or whose applicability is unknown. This requires knowledge of the list of potential targets so they can be aligned with information used in applicability checking (such as software inventories). Secondly, this information is needed to determine the completeness of an assessment – specifically, at what point have all applicable assets been assessed, and whether any assets that are applicable (or whose applicability is unknown) are unavailable for assessment.

Each Collector has a set of bound assets. Each Collector's set of bound assets is expressed as a list of assets about which the Collector (or more precisely, the PCEs associated with this Collector and associated PCXs) can collect information For that Collector, these assets represent the scope of its assessment capabilities in terms of potential targets. This information needs to be stored in the Repository. The list of assets also needed to be tied to assessment results and other information that might be used for applicability determinations. Collectors will need to update this information when new assets are bound to them. Managers will need to use this information to turn applicability statements in assessment instructions into a list of applicable targets and the set of associated Collectors needed to assess those targets.

Assessment Results

Results of assessments are returned from Collectors. These results are passed to the Repository for storage and to the Manager so it can track how much of an assessment has been completed. In some cases, it might make more sense for results to be queued in the message fabric for direct availability to the Application.

The expectation is that data returned by a Collector and stored in the Repository will utilize standardized result formats. It is expected that the specific format will be controllable by Application through their initial tasking instructions. This is necessary since the Application needs to be able to understand the results regardless of how they are returned from a PCE. The Application's selection will have to be one of a limited set of result formats supported by the SCAP architecture and be compatible with the nature of the assessment instructions sent by the application. (E.g., it would not make sense for an Application to request XCCDF results if the Application only sent OVAL check instructions.)

As with assessment instructions, it is expected that the Collector will serve as a translator between the format of results returned by a PCE and the format requested by the Application. As such, Collectors need to know not only how to translate received instructions into instructions that can be understood by

their PCEs, but they need to know how to take the PCE results and translate them into any valid result format.

The Repository will store and organize results of assessments. Results in the Repository will be associated with the target that was assessed and the instructions that guided the assessment. The Manager will query the Repository using identifiers associates with the assessment instructions that produced the results, which is associated with the result objects. For example, the Manager could query for results associated with an XCCDF Benchmark, an XCCDF rule, or an OVAL definition, and receive the results associate with the assessment directed by these assessment instructions. As a result, every distinct assessment instruction (regardless of format) must have a unique identifier, and every result (regardless of format) must capture the unique identifier from the instructions that gave rise to it. The SCAP v2 interfaces do not support querying based on other characteristics, such as querying for specific registry key values that might have been collected in the course of an assessment, although proprietary interfaces that would support this are not prohibited.

Collector/PCX capabilities

Collectors and PCXs have the ability to perform certain types of asset assessments using their associated PCEs. The instructions the Collector receives from the Manager will include the different assessment types (e.g., collecting registry keys, file permissions, analyzing XML files, etc.) and map these to the relevant PCEs or PCXs (which then need to map to their PCEs) that are capable of performing the indicated type of assessment of the targeted assets.

This information is specified using a combination of an identifier for the PCE itself and the check types the PCE can support. The PCE identifier is needed to ensure that any limits on the agents that collect the information, as specified by the Application, are followed. The check type is used to align instructions with PCEs capable of executing those instructions (either directly or after translation by the Collector). This information should be sufficient for a Collector to determine whether one of its PCEs can perform the assessment, one of its PCX's PCEs can perform the assessment, or if the Collector lacks the ability to perform the given assessment.

Interfaces and Activity Contracts

Standardized interfaces will connect the Application, Manager, Repository, Collector, and PCX components. PCEs do not have interfaces standardized in the architecture. Note that all interfaces are tied to a single component. This is because, while certain components are expected to invoke these interfaces most of the time, the SCAP architecture does constrain which components invoke the given interfaces. This allows greater flexibility in terms of architecture composition.

This section looks at some of the exchanges conveyed over these interfaces. It is likely that additional standardized exchanges will be identified as the standard develops.

Manager Interface

This section considers exchanges between Applications and the Manager.

Initiate Assessment

This message is sent to the Manager requesting that an assessment be performed and the results reported back. In short, this message initiates an assessment activity (both point-in-time and continuous

monitoring). In most cases, such a message would come from an Application. The information in this message includes:

- Assessment instructions – These are the instructions that will guide the assessment. This will also include any profile selections or tailoring instructions.
- Targeting expression – An expression to limit the scope of the assessment. This allows the requester to constrain an assessment to specific assets, to specific organizational units, assets known to have certain software installed, or other characteristics.
- Time bounds – This consists of:
    - Oldest acceptable results – This optional information specifies the oldest stored values that are acceptable in response to its request.
    - Latest acceptable return – This optional information specifies a time by which a result needs to be returned, even if the result is incomplete at this time. Missing results are marked as "unavailable" in the report.
    - Attempt fresh results – This is a Boolean value that is only valid if the "latest acceptable return" field is filled it. It specifies that the assessment should get new values if possible, but to fall back on stored values if new values are not available by the time results must be returned. The "oldest acceptable results" value, if provided, remains in effect and any stored results older than that value will still not be returned.
- Collection method – This optional field provides a list of pairs of PCE identifier and check type that is identical to the records used to specify Collector/PCX capabilities. It is used to specify when certain PCEs must or must not be used to service assessments using the named check type. The check type can be wild-carded. (This requirement is necessary because might deploy multiple PCEs with overlapping capabilities, but only some of those PCEs might have been determined acceptable for certain uses.)
- Result format and filters – This field specifies the type of result format to return as well as additional filters on what to include or exclude. The result format type will be one of the types specified for Repository storage (Assessment Results). Filters would allow further narrowing of the return values, such as eliding reporting of certain results (e.g., don't return "Not Applicable" results or results that don't indicate a need for remediative action).
- Collection parameters – This optional field specifies the format and filters of results to be stored in the Repository, which might differ from the results returned to party that requested the assessment. For example, requester might request a result that is succinct, but which provides a relatively low level of detail, while storing a more detailed set of results in the Repository. The requester could then identify the specific portions of the results that are of interest and request those results from the Repository. The collection parameters uses the same syntax as the "result format and filters" field. If absent, the collection will store the same type of result format returned to the requester

Upon receiving this message, the Manager generates a unique transaction ID for this assessment and publishes this so the requester can learn this information.

The Manager then employs the rest of the SCAP data collection architecture to conduct the described assessment. Once that process is completed, the manager publishes this result in conjunction with the transaction ID it assigned to this assessment.

Cancel Assessment

This message terminates an ongoing assessment. The only field in the message is the Transaction ID of the assessment to cancel. This message is ignored if it doesn't come from the same party that requested the assessment being cancelled.

When it receives this message, the Manager must immediately contact all Collectors tasked with this assessment and tell them to cancel their activities related to the identified assessment. No new results will be reported to the requester as a result of the identified assessment. If the Collectors had set up real-time monitors or timers for reassessment, these will all be cancelled unless they are still in use by other active assessments.

Repository Interface

This represents the interface for the Repository, which represents persistent storage within the data collection architecture.

Store Data

This message is sent to the Repository to add data into the Repository. Multiple components can use this interface.

The Manager uses this interface when it receives an Initiate Assessment information to store the assessment instructions and key parameters of the request for later reference. The following information is present in this message:

- Assessment instructions – As received in the Initiate Assessment message.
- Targeting – As received in the Initiate Assessment message.
- Oldest acceptable results – As received from the Time bounds field of in the Initiate Assessment message.
- Collection method - As received in the Initiate Assessment message.
- Result format - As received in the Initiate Assessment message.
- Transaction ID – The unique ID associated with this assessment.
- Requestor ID – The identifier for the party that requested this assessment.

This interface would also be used to store assessment results. This information would come from a Collector. When storing assessment results, in addition to recording the results themselves, the repository would store:

- A reference to the Initiate Assessment information that initiated the data collection, allowing those results to be linked to the instructions and parameters associated with their collection.
- The target from which the results were collected.
- The Collector, PCX (if used), and PCE used to collect the data.
- A timestamp for when the results were collected.

In addition, Collectors would also use the repository to store the set of known targets they can assess as well as the set of PCEs bound to a given Collector.

Finally, Applications might store additional data lists that could be used in applicability calculations. For example, an Application might store a list of targets that belong to a given organizational unit, so that membership in that organizational unit can be used as applicability criteria.

Query

This is used to send arbitrary queries to the Repository. These might be queries passed along from the Application by the Manager or might be queries from the Collector requesting specified assessment instructions during an assessment. This message only includes a query string.

When it receives this message, the Collector must immediately check its stored data for information matching the query. Matching information is returned in a Query Result message.

A Query could request any information stored in the repository. This includes, but is not limited to, results of prior assessments, lists of potential assessment targets (potentially filtered by applicability to certain criteria), list of PCE agents, previous assessment instructions, and other information that is populated within the Repository.

One important use of the query function is to check the Repository for the present of relevant, previously-collected data when an Application requests an assessment. If the Initiate Assessment is for a one-time data collection, the Manager queries the Repository to determine whether the Repository has some or all of the requested results. This query uses the identities of the received assessment instructions along with parameters for required freshness, follow any collection method restrictions, and result format. The Repository will need to take these parameters and analyze its data using a process equivalent to the following:

1) It determines whether there have been prior assessments using the same assessment instructions. It does this by searching for prior assessment instructions that use the same unique IDs for assessment instructions. This search is conducted at multiple levels. (E.g., the Repository would look for XCCDF benchmarks with the benchmark ID of the assessment instructions and also would look for the OVAL Definition IDs. The former might indicate the full assessment was done before, while matches only against the latter could indicate at least some parts had previously been performed.)
2) Assuming there is a match, the Repository determines whether the discovered results are suitably recent, with the relevant Query parameter set according to the Oldest acceptable results field from the Initiate Assessment message.
3) Assuming there are still valid matches, the Repository determines whether the discovered results are in a format that can be used to populate requested result type, where the relevant Query parameter comes from the Result format and filters field in the Initiate Assessment message. (A more granular stored result could populate a less granular result type, but not vice versa.)
4) Assuming there are still valid matches, the Repository checks whether these matches conform to collection method restrictions, where the corresponding Query parameter is set based on the Collection method field of the Initiate Assessment message.
5) If any assessment results pass all these tests, they are posted so the requesting party can find the results.

Another important use of the Query interface is for identifying the set of applicable targets associated with a given set of assessment instructions. The Manager sends a Query to the Repository to collect this information when it receives an Initiate Assessment message that cannot be fully satisfied by a Query to the Repository for prior results. In this case, the Manager constructs a query using both targeting information, from the Targeting expression field of the Initiate Assessment message, and any

applicability information included in the assessment instructions. The latter would be expressed in the Query using the unique identifiers of the applicability check instructions. Based on this, the Repository would make one of three possible determinations for each asset bound to some Collector in the architecture:

- The target is applicable – the Repository is able to find results for the applicability checks associated with the target and those results indicate the target is applicable.
- The target is not applicable – the Repository is able to find results for the applicability checks associated with the target and those results indicate the target is not applicable.
- The target's applicability cannot be determined – the Repository either lacks some results for applicability checks needed to determine applicability of the target or it has results for applicability checks but those results are not able to determine applicability.

Both applicable targets and targets whose applicability cannot be determined would need to be posted so the Manager can direct assessment of those targets.

Collector/PCX Interface

A Collector receives assessment instructions and returns the results of these assessments. Normally, these would come from the Manager. Similarly, a PCX would receive the same types of instructions from its associated Collector. Both the Collector and PCX participate in the following exchanges:

Collector Request

This is a message tasking a given Collector or PCX with the collection of information needed for an assessment. The message contains the following information:

- Assessment Instruction IDs – These are references to the assessment instructions to guide assessment. These are only references because the Collector/PCX caches copies of prior assessment instructions to reduce the size of network messages. If the Collector/PCX does not have the referenced assessment instructions, it can query the Repository to collect it. Note that the referenced assessment instructions might be a subset of the assessment instructions sent in an Initiate Assessment if some of the assessment instructions are not applicable to the Collector's/PCX's scope.
- Targeting – These identify the subset of the Collector's/PCX's scope that are to be assessed. If this field is absent, the Collector's/PCX's whole scope is assumed to be targeted.
- Latest acceptable return – This optional field is the latest time by which the Collector/PCX should return a result.
- Collection method – The portion of the Collection method field received in the Initiate Assessment that is applicable to this Collector/PCX.
- Report parameter – As received in the Initiate Assessment message. This determines the format of the information returned by the Collector/PCX.
- Transaction ID – The unique identifier for this assessment.

Note that the message sent from a Collector to a PCX might include only a subset of the assessment instruction IDs, a subset of the targeted assets, and a subset of the collection methods received by the Collector itself in order to limit PCX activities to those needed. (For example, the Collector and PCX might share certain capabilities, so if the Collector will provide a given capability, instructions to the PCX might exclude instructions that use those capabilities and/or might be explicitly excluded in the collection

methods sent to the PCX.) The Latest acceptable time sent to the PCX might also be adjusted so that the Collector has time to process PCX returns in time to return timely results.

Upon receiving this message, a Collector would first determine if it already had the referenced assessment instructions and, if not, query the Repository to retrieve this information. It would then use the assessment instructions, targeting, and collection method fields to determine which of its PCEs it needs to use to make the collection, and which (if any) of its PCXs need to be utilized. Relevant PCX's are invoked by sending them an appropriate Collector Request message. The Collector then converts the relevant portions of assessment instructions to the appropriate instructions to its PCEs and initiates the corresponding assessments.

The steps taken by a PCX in response to receiving this message are the same except that the PCX will not invoke another PCX.

Once collection has been completed, the results are posted, along with the transaction ID so they can be tied to the original Initiate Assessment.

Detailed Processing Flow

The preceding sections have provided an overview of the SCAP v2 data collection architecture, and details as to data sets and interfaces. The current section puts all of this together to describe in detail how the architecture will support its two primary use cases. Note that, while the interfaces only specify the activities of the recipient of instructions, the process flow below describes both the sender and recipient in order to provide a more fleshed-out example.

One-time Enterprise Assessment

This describes the steps by which a one-time assessment is performed on an enterprise.

1) An Application sends an Initiate Assessment message to the Manager. This contains the assessment instructions guiding the assessment along with additional parameters.
2) The Manager generates a transaction ID for this assessment and posts it for the requesting Application.
3) The Manager sends a Store Data message to the Repository. This message contains the assessment instructions guiding the assessment and relevant parameters.
4) The Manager sends a Query message to the Repository to check for relevant results. The Repository checks its memory to determine whether it has any relevant and suitably fresh results that can be applied to this assessment.
5) The Repository posts any relevant results it has discovered and the Manager retrieves this information. The Repository also posts the list of assets that are applicable (or whose applicability cannot be determined).
6) If the Repository response is sufficient to meet the assessment, skip to step 18). Otherwise, the Manager sends a Query to the Repository to identify applicable targets for assessment.
7) The Manager takes the list of targeted endpoints and sends another Query to the Repository to find Collectors whose scopes are collectively sufficient to cover the entire set of applicable targets for the assessment.
8) The Manager creates a Collector Request for each needed Collector, tasking it with a portion of the assessment. This message will be based on the contents of the original Initiate Assessment

but might be reduced if certain aspects are not applicable to a given Collector or if the Repository results obviate the need to gather certain results.

9) The Collector will check whether it has a local copy of the referenced SCAP instructions. If not, it will Query the Repository for those instructions an store them.
10) The Collector takes the information in the Collector requests and identifies the assessment capabilities necessary to service the request. Based on this it will identify PCEs and PCXs that can perform the assessment. It is possible that the assessment instructions might require capabilities the Collector does not have access to via its PCEs and PCXs, in which case those portions of the assessment will simply be assigned results indicating they could not be assessed.
11) The Collector will craft and send Collector Request messages to any relevant PCXs, tasking them with their portion of the assessment. These Collector Requests may contain a subset of the Collector Request received by the Collector, reflecting the role the PCX will have in the assessment.
12) All Collectors and PCXs tasked in the assessment will convert their instructions into commands to send to the appropriate PCEs.
13) The Collectors and PCXs will collect the information gathered by their PCEs. The Collectors and PCXs will augment this information as necessary (e.g., adding timestamps, identities of the PCEs that performed the checks, etc.) and format this information into the appropriate type of SCAP result.
14) PCXs will pass their results on to their Collector, which will integrate the PCX results into their own results.
15) When all results have been gathered (or when the Collector has reached the end of its allotted time to perform the assessment), the results will be posted. The Manger and Repository will be alerted to and retrieve these posts.
16) The Repository will store the results (along with the associated transaction ID) for future reference.
17) The Manager will integrate the results received from all contacted Collectors along with any Repository data it received in step 5).
18) The Manager will convert the result set into the appropriate format requested by the Application. It will also apply any specified filters.
19) The Manager will post the results. The Application will be alerted to and retrieve these results.

Ongoing Enterprise Monitoring

This describes the steps by which an enterprise sets up ongoing monitoring based on a set of assessment instructions.

Initial deployment and baseline gathering

1) An Application sends an Initiate Assessment message to the Manager. This contains the assessment instructions guiding the assessment along with additional parameters.
2) The Manager generates a transaction ID for this assessment and posts it for the requesting Application.
3) The Manager sends a Store Data message to the Repository. This message contains the assessment instructions guiding the assessment and relevant parameters.
4) The Manager sends a Query to the Repository to identify applicable targets for assessment. The Repository posts the set of applicable targets (or targets for which applicability cannot be determined) based on the assessment instructions.

5) The Manager takes the list of targeted endpoints and sends another Query to the Repository to find Collectors whose scopes are collectively sufficient to cover the entire set of applicable targets for the assessment.
6) The Manager creates a Collector Request for each needed Collector, tasking it with a portion of the assessment. This message will be based on the assessment instructions of the original Initiate Assessment but might be reduced if certain aspects are not applicable to a given Collector or if the Repository results obviate the need to gather certain results.
7) The Collector will check whether it has a local copy of the referenced SCAP instructions. If not, it will Query the Repository for those instructions an store them.
8) The Collector takes the information in the Collector requests and identifies the assessment capabilities necessary to service the request. Based on this it will identify PCEs and PCXs that can perform the assessment. It is possible that the assessment instructions might require capabilities the Collector does not have access to via its PCEs and PCXs, in which case those portions of the assessment will simply be assigned results indicating they could not be assessed.
9) The Collector will craft and send Collector Request messages to any relevant PCXs, tasking them with their portion of the assessment. These Collector Requests may contain a subset of the Collector Request received by the Collector, reflecting the role the PCX will have in the assessment.
10) All Collectors and PCXs tasked in the assessment will convert their instructions into commands to send to the appropriate PCEs. In addition to collecting the requisite information from the endpoint, if necessary, the commands will also instruct the PCEs to establish monitors to report changes to
11) The Collectors and PCXs will collect the information gathered by their PCEs. The Collectors and PCXs will augment this information as necessary (e.g., adding timestamps, identities of the PCEs that performed the checks, etc.) and format this information into the appropriate type of SCAP result. Collectors and PCXs will locally store the collected information for future reference.
12) The Collectors and PCXs will set up timers to re-check settings per the assessment instructions. The period of these timers will be based on instructions in the assessment instructions. In addition, if attempts to set up real-time monitors on PCEs fail, timers will be set to frequently re-check those settings.
13) When all results have been gathered (or when the Collector has reached the end of its allotted time to perform the assessment), the results will posted. The Repository and Manager will receive alerts and collect this information. These results constitute the baseline against which future changes will be measured.
14) The Repository will store the baseline results (along with the associated transaction ID) for future reference.
15) The Manager will convert the baseline result set into the appropriate format requested by the Application. It will also apply any specified filters.
16) The Manager will post the baseline results. The Application will be alerted to and collect these results.

Reporting of detected changes

1) New information is delivered to a Collector or PCX.
    a. This might constitute a change detected by real-time monitors on a PCE.

    b. This might come about because one or more of the timers running on a Collector or PCX has expired. In this case, the Collector/PCX sends a command to the corresponding PCE to collect the indicated information. The timer is then restarted.
2) The Collector/PCX performs the following checks:
    a. Does the collected value indicate a change from the previously saved value? If no, no further processing is necessary. If yes, the Collector/PCX updates its stored value.
    b. Assessment instructions can consist of complex checks, combining multiple pieces of information. It the monitored instruction is includes an evaluation, the instruction will be evaluated to see if the new information chances its result. If no, no further processing is necessary. If yes, the Collector/PCX updates its stored value for this evaluation. If the instruction is not evaluated (i.e., it is just collecting and reporting a system value), then always proceed to the next step.
3) The Collectors and PCXs will augment the collected information as necessary (e.g., adding timestamps, identities of the PCEs that performed the checks, etc.) and format this information into the appropriate type of SCAP result.
4) PCXs will pass these results on to its Collector.
5) The Collector post these results. The Manager and the Repository will be alerted to these changes and collect the updated results.
6) The Repository will store these results (along with the associated transaction ID) for future reference. Depending on how the Repository is configured, it may replace an old value with the new results (if it is not tracking historical information) or it may simply add the new results to its storage (if it is tracking historical information).
7) The Manager will convert the results into the appropriate format requested by the Application. It will also apply any specified filters.
8) The Manager will post the collected results. The Application will be alerted to and retrieve the results.

End monitoring

1) The Application sends a Cancel Assessment message to the Manager with the Transaction ID of the current monitoring effort.
2) The Manager sends a Collector Request message to all the Collectors associated with the cancelled assessment. This message has the Transaction ID of the cancelled assessment, but the assessment instruction IDs field is blank, indicating that there is no longer any action associated with this assessment.
3) The Collector passes this Collector Request on to any PCXs supporting the cancelled assessment.
4) The Collector and PCX check if the monitors established for the cancelled assessment are also used by other, active assessments. If so, these monitors are left active. (It will likely be necessary to use a counter to track this.) For all real-time PCE monitors that are only used by the cancelled assessment, the Collector or PCX crafts a command to the corresponding PCE and instructs it to halt the real-time monitoring.
5) The Collector and PCX check if the periodic reassessment timers established for the cancelled assessment are also used by other, active assessments. If so, these timers are left active. (Again, it will likely be necessary to use a counter to track this.) Timers that are not used by other assessments are deleted.

Open Questions

1) It is unclear the role the Manager needs to take in distribution of results to the Application. On one hand, having the Manager compile results could help clarify the meaning of results, especially with regard to the completeness of the assessment since an Application is likely to have little visibility into this. On the other hand, the Repository's and Collectors' posts could be directly retrieved by the Application, removing the Manager as a middleman in this exchange.
2) There is a desire to support a wide range of assessment instruction types. PCEs are unrestricted in the types of inputs they can require. The SCAP architecture requires that all instructions received by a Manager have a way to uniquely identify them (to enable lookup of the instructions and their results in the Repository) and have their type clearly identified (so the Collector can determine how to route them). Collectors and PCXs will have instruction translation capabilities tied to their supported PCEs, but would inevitably be limited in the types of inputs they could take. It remains an open question as to what the official instruction formats of the SCAP architecture will be (i.e., the formats that all Collectors and PCXs will be expected to accept and use), and if/how other formats will be "wrapped" in an acceptable format to allow them to be used by Collectors.
3) There is a need to determine how periodicity of monitoring will be controlled in the architecture.
4) There is a need to control the nature of assessment results to ensure that they are usable by Applications. The details of what this control looks like, as well as the set of supported result formats, remains an open question.
    a. Related to this, every result format needs to be able to support a way to indicate "WAS UNABLE TO PERFORM ASSESSMENT" so that Collectors can appropriately indicate this for checks they are unable to assign to any PCE or PCX.
5) Details of how applicability checking is expressed and process remain under-defined and will need to be worked out.

Conclusion

This document represents the latest understanding of the design of the SCAP v2 data collection architecture as of the document's date. This architecture remains a work in progress. Comments and feedback are welcome.

It is hoped that this design can be used to create an initial implementation of the architecture as a proof of concept. While the lack of a standardized serialization for data would prevent interoperability with such an implementation, the work would help identify any remaining design issues requiring resolution, and likely would provide initial insights into reasonable serializations.

A standardized version of this design would include greater detail about syntax and semantics of interfaces and the serialization of data.