INTERNET-DRAFT   Mark Davidson
Intended Status: Proposed Standard   The MITRE Corporation
                                                          David Oliva
Expires: March 28, 2013   September 24, 2012

Asset Summary Reporting
draft-davidson-sacm-asr-00

Abstract

This specification defines the Asset Summary Reporting (ASR), a data model for expressing the data exchange format of summary information relative to one or more metrics. ASR reduces the bandwidth requirement to report information about assets in the aggregate since it allows for reporting aggregates relative to metrics, as opposed to reporting data about each individual asset, which can lead to a bloated data exchange. ASR is vendor neutral and leverages widely adopted, open specifications; it is flexible, and suited for a wide variety of reporting applications.

Status of this Memo

Copyright and License Notice

Table of Contents

1 Introduction

The Asset Summary Reporting (ASR) format is a data model to express the transport format of summary information about one or more sets of assets. The data model facilitates the interchange of aggregated asset information throughout and between organizations. ASR is vendor neutral and leverages widely adopted, open specifications; it is flexible, and suited for a wide variety of reporting applications. The primary goal of the ASR format is to describe summary information about one or more arbitrarily large and complex asset-related data sets in a standardized manner. Second, ASR seeks to allow content producers the ability to choose an appropriate level of detail depending on their needs and data set size requirements. Finally, ASR seeks to reduce the complexity of producing and consuming summary result documents. For the purposes of this specification, an asset is considered to be anything that has value to an organization. Computing devices are one form of asset that many organizations track. Additional examples are networks, people, and organizations. This specification, however, does not limit asset summary reporting to those examples; information about any set of assets may be summarized. While this specification was developed to support the immediate needs of the security automation and the continuous monitoring communities, it is expected that this specification will be valuable to any process where producing or consuming summary data is desired.

1.1 Purpose and Scope

The purpose of this report is to define the ASR specification. The report gives an introduction to ASR version 1.0, defines ASR’s data model, and documents the conformance requirements to comply with ASR. Other versions of ASR and the associated component specifications, including emerging specifications and future versions, are not addressed here. Future versions of ASR will be defined in distinct revisions of this report, each clearly labeled with a revision number and the appropriate ASR version number. This report does not describe the queries, instructions, methods, processes, or data required to produce an ASR document. This report does not describe how to transform any specific data model or data set into an ASR document. This report normatively describes only the ASR format. The appendices contain additional information about how to use ASR.

1.2 Audience

The intended audience for this specification is product vendors who are developing applications that either produce or consume aggregated asset information, particularly those that will interoperate with other producers or consumers of aggregated asset information.

1.3 Document Structure

The remainder of this document is organized into the following major sections:

Section 2 defines the terms used within this specification and provides a list of common abbreviations.

Section 3 describes how this specification relates to other standards and specifications.

Section 4 defines the conformance requirements for ASR.

Section 5 provides an overview of the ASR data model constructs and key concepts.

Section 6 documents the ASR data model.

Section 7 specifies the requirements for defining a record set type.

Appendix A describes use cases for ASR.

Appendix B indicates how to integrate ASR with the Asset Reporting Format (ARF).

Appendix C provides some ASR examples.

Appendix D provides examples of record set type definitions.

Appendix E lists pre-defined record attributes.

Appendix F documents normative references.

1.4 Document Conventions

Throughout this specification, when referencing a specification listed in Appendix F, the name will be written between brackets, such as [XSD].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

XML elements [XML] are referred to using qualified names when they are not in the ASR namespace. Elements with no prefix can be assumed to be in the ASR namespace, unless otherwise noted. A qualified name associates a named element with a namespace. The namespace identifies the specific XML schema that defines (and consequently may be used to validate) the syntax of the element instance. A qualified name declares this schema to element association using the format ’prefix:element-name’. The association of prefix to namespace is defined in the metadata of an XML document and varies from document to document. In this specification, the conventional mappings listed in Table 1-1 are used.

Table 1-1: Conventional XML Mappings

Mappings Prefix	Namespace URI	Schema
ai	http://scap.nist.gov/schema/asset-identification/1.1	Asset Identification 1.1
arf-rel	http://scap.nist.gov/specifications/arf/vocabulary/relationships/1.0#	ARF 1.1 Relationships
asr	http://scap.nist.gov/schema/asset-summary-reporting/1.0	ASR 1.0
asr-attr	http://scap.nist.gov/schema/asset-summary-reporting/1.0/attr	ASR 1.0 Common Attributes

2 Terms and Abbreviations

This section defines a set of common terms and abbreviations used within this specification.

2.1 Terms

Asset: Anything that has value to an organization, including, but not limited to, another organization, person, computing device, information technology (IT) system, IT network, IT circuit, software (both an installed instance and a physical instance), virtual computing platform (common in cloud and virtualized computing), and related hardware (e.g., locks, cabinets, keyboards).

Asset Identification: The attributes and methods necessary for uniquely identifying a given asset. A full explanation of asset identification is provided in [Asset Identification].

Data Source: Represents a source of data that could be used to build a summary report.

Population: A defined set of assets from which reporting is based.

Record Set: A grouping of related data about a topic, organized into records and data elements.

Record Set Type: A description of a record set that defines requirements for the record set, and the semantics of the data fields.

Summary Report: A collection of related record sets into a coherent report, as defined by the report creator. A summary report may be represented in multiple pages, which would be manifested as multiple XML documents.

2.2 Acronyms

ARF - Asset Reporting Format

ASR - Asset Summary Reporting Format

CCE - Common Configuration Enumeration

CCSS - Common Configuration Scoring System

CISO - Chief Information Security Officer

CPE - Common Platform Enumeration

CPU - Central Processing Unit

CVE - Common Vulnerabilities and Exposures

CVSS - Common Vulnerability Scoring System

CWE - Common Weakness Enumeration

CWSS - Common Weakness Scoring System

FIPS - Federal Information Processing Standard

IETF - Internet Engineering Task Force

IR - Interagency Report

ISCM - Information Security Continuous Monitoring

IT - Information Technology

ITL - Information Technology Laboratory

NIST - National Institute of Standards and Technology

OCIL - Open Checklist Interactive Language

OS - Operating System

OVAL - Open Vulnerability and Assessment Language

RACI - Responsible, Accountable, Consult, Inform (Responsibility Matrix)

RFC - Request for Comment

SCAP - Security Content Automation Protocol

SIEM - Security Information and Event Management

SP - Special Publication

SWID - Software Identification

URI - Universal Resource Identifier

USGCB - United States Government Configuration Baseline

W3C - World Wide Web Consortium

XCCDF - Extensible Configuration Checklist Description Format

XML - Extensible Markup Language

XSD - XML Schema

XSLT - Extensible Stylesheet Language Transformations

3 Relationships to Existing Standards and Specifications

ASR’s relationships to other selected specifications are described below.

1. Asset Identification - ASR leverages [Asset Identification] to identify assets for the ASR document. Asset Identification is a useful component of ASR, as it enables the correlation and fusion of various ASR documents and ASR content. ASR provides a place to optionally list assets defined using [Asset Identification].

2. Asset Reporting Format (ARF) - ASR may be used as a payload for an Asset Reporting Format (ARF) document [ARF]. ASR is not required to be packaged as an ARF payload; however, Appendix B provides guidance on using ASR in an ARF report.

4 Conformance

Developers and organizations may want to build products in conformance with ASR to foster consistency and interoperability of those products. Users of those products, and consumers of the content generated by those products, would then know the format of the data that the product will produce and can consume. In addition, products that conform to this specification will be better able to interoperate and exchange reporting information with other products that conform to ASR. Products may want to claim conformance with this specification to advertise their interoperability with other ASR compliant tools, as well as to meet requirements set by other specifications or organizations. The following sections define the criteria for content and products to claim conformance with this specification.

4.1 Product Conformance There are two types of ASR products: report producers and report consumers. Report producers are products that generate ASR documents, while report consumers are products that accept an existing ASR document and process it. Products claiming conformance with this specification SHALL adhere to the following requirements: 1. For report producers, generate well-formed content as defined in Section 4.2. 2. For report consumers, consume and process well-formed content as defined in Section 4.2. 3. Make an explicit claim of conformance to this specification in any documentation provided to end users.

4.2 Content Conformance In order for an ASR report to be considered in compliance with this specification, the report MUST adhere to the following requirements: 1. The ASR report SHALL conform to all of the normative guidance provided in Section 6. 2. Each record set within an ASR report SHALL conform to the requirements set by its declared record-set-type, as described in Section 7.

5 ASR Data Model Overview and Key Concepts

This section provides an overview of the ASR data model structure and design philosophy. The data model defines a format for representing one or more collections of data. A collection of data about assets is called a record set. At its simplest, an ASR report is a collection of record sets. The following sections introduce the key concepts of the ASR data model.

5.1 Data Model Overview

The following sections provide a high level overview of the main data model constructs.

5.1.1 Summary Report

An ASR document (i.e., a "summary report") is composed of one or more record sets. A summary report is the highest level notion in the ASR data model. One or more XML documents may compose a summary report, so the concept of a summary report is not confined to the physical boundaries of an XML document.

5.1.2 Record Set

A record set is a collection of data organized into individual records. A record set optionally provides a reference to information about the source of the data for the records. The context of a record set is communicated by declaring a "type", known as a record set type. Section 5.2.2 provides more details regarding the record set type.

5.1.3 Record

The record construct is the primary mechanism for conveying data in a summary report. All record data is contained in the attributes of the record construct. The properties of a record are defined by the record set type. The record construct may have any number of properties, as permitted by the record set type. In general, properties are expected to be qualified XML attributes as described in Section 5.2.1.

5.1.4 Data Source

In addition to capturing a collection of records, a record set may capture information about the origin of the data in its records. Data source information is captured separately from the record set, but each record set may reference a data source. The same data source may be referenced by multiple record sets if that is appropriate for the summary report.

5.1.5 Metadata

A summary report may have metadata about its creation. The metadata should include the name of the tool or person that generated the report, the time the report was generated, and any other pertinent information.

5.2 Key Concepts

The following sections provide an overview of the key concepts in composing a summary report.

5.2.1 Record Attributes

Record attributes are the core construct for construing information in ASR. A record set type defines the allowable and required attributes on a record in a particular record set. The attributes must be namespace qualified in order to give context as to the meaning of the attribute. In XML, attributes that are not namespace qualified belong to the "no namespace" realm, which does not give context to the meaning of the attribute. The namespace and local-name must be defined in the corresponding record set type.

5.2.2 Record Set Type

The concept of a record set is generic in nature, and it is based on the expectation that a report creator must define a record set type or adopt an existing record set type. A record set type is a definition of a record set. The record set type defines all properties of a record set. In object-oriented programming, a class is the functional equivalent of a record set type, and an object is the functional equivalent of a record set instance. The record set type describes the attributes that "SHOULD", "MUST", and "MAY" be included in a record in that record set. It also describes the semantics of each attribute. Section 7 gives specific requirements for defining a record set type. While a record set type must be defined, the format of the record set type is not strictly defined. A record set type definition is an agreement between producer and consumer. It is anticipated that record set type definitions may take the form of prose, XML, spoken word, or any other form of communication capable of conveying the requirements of a record set type. The authors of this specification have provided a data model and XML schema for record-set-type-definitions that may be used for this purpose. The record-settype-definition data model is covered in Section 7, and that section also provides a pointer to the recordset-type-definition XML schema.

5.2.3 Paging

Since a summary report can grow too large for available resources (e.g., network bandwidth, memory, CPU), a summary report may be divided into multiple pages. A paged summary report means that a single summary report is represented as two or more separate XML documents. This allows report creators the flexibility to reduce the resources required to produce and exchange a single large summary report by breaking it up into many smaller reports. Paging exists to support use cases where the amount of data contained in an ASR exceeds reasonable computing thresholds. Paging can be used to send multiple, smaller segments of information instead of one large block of information. Each page of an ASR should be consumable without the other pages when possible. All paging information is contained in the root element of each XML document.

5.2.4 Referencing Assets

A record may be allowed or required to capture an asset list as a child. If an asset list is captured, there must be a count attribute in the record that is identified as the "count for the asset list", as defined in Section 7. That attribute must hold a value equal to the number of assets listed within the record. The "count for the asset list" attribute represents the count of assets that meet certain criteria. The criteria are specified in the description of the count attribute.

6 ASR Data Model Description

This section describes the requirements for the ASR data model manifested as Extensible Markup Language (XML). Section 6.1 provides a conceptual overview of the data model, while Section 6.2 examines the actual XML data model in detail.

6.1 XML Data Model Introduction

Figure 6-1 provides a logical view of a sample summary report spread across two pages. For brevity some attributes have been omitted. In the diagram, each record set claims a type and a data source description. Those items give additional context to understand the meaning of the data captured in the record set.

```
+---------------------+
| summary-report      |
|---------------------|
|                     |
| metadata (0, 1)     |
|                     |
| data-source (0, *)  |
|                     |
| record-set (1, *)   |
|                     |
+---------------------+
```

The summary-report element is the root element of the ASR data model; it contains identification and paging information for an ASR document. One ASR document may be paged into multiple XML documents as desired, but each record set MUST be completely represented on one page of a summary report (i.e., a record set may not span multiple pages). Summary-report has three attributes: report-id, page-number, and last-page; it also has three children: metadata, record-set, and data-source.

The metadata element is a child of the summary-report element. Metadata contains information related to the generation of an ASR document. Metadata has two attributes: generator-name and timestamp. Metadata does not have any defined children, but there is an extension point where any arbitrary XML is allowed.

The record-set element is a child of the summary-report construct. A record-set contains summary data, and may appear multiple times in the same summary-report. Record-set has four attributes: id, data-source-ref, record-set-type, and comment. Data-source-ref is a reference to the data-source element that represents the source of the information used in the record-set, record-set-type indicates the type of report being represented, and the comment field allows a text comment, if desired. Record-set has one child: record.

The record element is a child of the record-set construct, and contains any number of attributes; record may appear multiple times in the same record-set. Record has two children: asset-references and identifier-list; only one of the two may be present for a single report. Both children provide methods to identify and list the assets that each record describes.

The identifier-list element is used to enumerate assets that relate to data in the record. In the definition for the record set type, an attribute that contains a count may be associated with this list. When that occurs, the identification information captured in this element is known to enumerate the list of assets that make up the associated count. Assets are enumerated using this element through a unique identification scheme defined by capturing the URI for the schema. Subsequently, each id provided as a child to this element is understood within the context of the identifier scheme specified. This solution is optimal when assets are easily identified using a single string identifier.

The asset-references element is used to enumerate assets that compose a count in the record. This element is used, instead of identifier-list, when assets are identified using [Asset Identification]. This element has a single attribute that accepts a list of identifiers. Those identifiers must match the identifiers of assets captured in the data-source element. The assets in the data-source element are identifiable using either [Asset Identification], or some other identification scheme. In either case, the identification may be more robust than is permitted with the identifier-list element.

The data-source element is a child of the summary-report element. It has four attributes: id, resource, population-size, and comment. Id is the identifier of the data-source from which a record set is generated. Resource is a URI indicating the actual resource that the data-source element represents. Population-size indicates the number of assets that the resource has knowledge of. Comment allows a comment about the data-source. Data-source has four children: scan-info, extended-info, ai:assets, and asset-list.

The scan-info element is a child of the data-source element. Scan-info is intended to be used when the data-source is a network, vulnerability, compliance, or other type of scan. Scan-info has seven attributes: scan-id, authenticated, execution-location, scan-start, scan-end, population-applies-to, and population-assessed. Scan-id is the ID of the scan. Authenticated represents whether the scan was authenticated or not. Execution-location represents where the scan took place. Scan-start and scanend represent the start and end date and time of the scan. Population-applies-to represents the number of assets that the scan applied to; if a scanner has knowledge of 100 assets, but a particular scan only applies to 50 of them (e.g., a patch scan for a specific OS), the population-applies-to would be set to 50. Population-assessed represents the number of assets that were assessed in the scan. Scan-info has one child, scanner.

The scanner element is a child of scan-info. Scanner has three attributes: product-name, productversion, and scanner-type. The product-name contains the name of the scanner that produced this scan. Product name SHOULD be a CPE name or a SWID, but MAY be any string. Productversion indicates the version of the scanner. Scanner-type indicates the type of scanner. Scanner does not have any children.

The extended-info element is a child of data-source. It is an extension point of the ASR schema, intended to allow data-source information that cannot be captured in other data-source constructs. Extended-info does not have any attributes. Extended-info may have any XML children. The ai:assets element is a child of data-source. It is defined in the [Asset Identification] specification, and is used to list assets. Please see the [Asset Identification] specification for details of this element.

The asset-list element is a child of data-source. It is used to list assets when using ai:assets is not feasible or desired. Asset-list does not have any attributes. Asset-list has one child, asset, which is used to list individual assets.

The asset element is a child of asset-list. Asset has one attribute, id. Id is used to uniquely identify a single asset within the scope of a data-source. Asset may have any XML children.

6.2 XML Data Model Requirements In order to comply with the ASR data model,

The user MUST produce an XML asr:summary-report element consistent with the data model described below.

The XML element produced MUST validate against the XSD for Asset Summary Format 1.0 listed at http://scap.nist.gov/specifications/asr/#resource-1.0. In situations where the XSD does not match the documented model in this specification, the XSD SHALL take precedence. The following tables formalize the data model. The data contained in the tables are requirements and MUST be interpreted as follows:

The "Element Name" field indicates the name for the XML element being described. Each element name has a namespace prefix indicating the namespace to which the element belongs. See Table 1-1 for a mapping of namespace prefixes to namespaces.

The "Definition" field indicates the prose description of the element. The definition field MAY contain requirement words as indicated in [RFC 2119].

The "Properties" field is broken into four subfields:

o The "Name" column indicates the name of a property that MAY or MUST be included in the described element, in accordance with the cardinality indicated in the "Count" field

o The "Type" column indicates the REQUIRED data type for the value of the property. There are three categories of types: literal, element, and special. A literal type will indicate the type of literal as defined in [XSD]. An element type will reference the name of another element that ultimately defines that property. A special type is listed when the type is neither literal nor element. The special type will indicate the nature of permitted content, such as allowing any XML to be used.

o The "Count" column indicates the cardinality of the property within the element. The property MUST be included in the element in accordance with the cardinality. If a range is given, and "n" is the upper-bound of the range, then the upper limit is unbounded.

o The "Definition" column defines the property in the context of the element. The definition MAY contain requirement words as indicated in [RFC 2119].

Element name: asr:summary-report	
Definition	An ASR report may need to be expressed through multiple XML instances. This element is the root of an ASR XML instance. Each ASR XML instance SHALL consist of a single root asr:summary-report element, which encloses a single page of a summary report. An ASR report MAY span one or more pages, indicated by the report-id and page-number properties.

Properties			
Name	Type	Count	Definition
report-id	NCName	1	The identifier for the report. This SHOULD be unique on a per-report basis. In the case where one report consists of multiple XML documents, this ID MUST match the ID of the related report documents.
page-number	positive integer	1	Which page of the report this XML document represents. If the entire summary report is represented in a single XML document, this attribute MUST be set to "1". If the summary report spans multiple pages, this attribute MUST be set to the positive integer that indicates the page of the current XML document. Each page MUST be numbered sequentially, and

			the sequence MUST start with "1".
last-page	boolean	1	If this XML document represents the last page of a summary report. last-page MUST be set to "true" when this page is the last page, and MUST be set to "false" otherwise. When a summary report is fully represented on one page, last-page MUST be set to "true".
metadata	asr:metadata	0-1	Information about the generation of this asr:summary-report. See Table 6-2.
record-set	asr:record-set	1-n	Information about the record set. See Table 6-3.
data-source	asr:data-source	0-n	A source of data for an asr:summary-report. See Table 6-7.

Note: I will create Tables 6-2 through 6-12 in similar fashion to 6-1 if table 6-1 looks good. Otherwise, it didn’t seem worth the time to do all the tables before knowing if the format was appropriate for an RFC.

TODO: Create Table 6-2

TODO: Create Table 6-3

TODO: Create Table 6-4

TODO: Create Table 6-5

TODO: Create Table 6-6

TODO: Create Table 6-7

TODO: Create Table 6-8

TODO: Create Table 6-9

TODO: Create Table 6-10

TODO: Create Table 6-11

TODO: Create Table 6-12

7 Defining a Record Set Type

A record set type is the definition of a record set; it gives context to a record set. A record set type is defined separate from a summary report. It defines the purpose of a record set, the required and optional attributes for each record, whether an asset list should be provided (and if so, which attribute it associates with), and any additional information related to the record set type. This section documents the requirements for defining a record set type.

A record set type may be defined in any form (e.g., verbal, human readable, XML). In whatever form the record set type definition takes, it MUST specify the following information:

A namespace-qualified name, using [XSD Qual] Section 3.2 Qualified Locals, identifying the record set type. This QName is referenced from each record set that implements the record set type. The record set type name is specified in the record-set-type attribute of the record-set element.

The intended use of the record set type.

The namespace-qualified attributes for each record within the record set type. Attributes MUST be namespace qualified as defined using [XSD Qual] Section 3.2 Qualified Locals.

o For each attribute, it MUST indicate if the attribute MUST, MUST NOT, SHOULD, SHOULD NOT, or MAY be present. For each attribute, it MUST also specify a description of the attribute.

o If an asset list is permitted or required, a single attribute MUST be designated as the "countfor-asset-list" attribute. The count-for-asset-list attribute MUST have a type that only allows non-negative integers. A count-for-asset-list attribute SHALL contain an integer that is equal to the number of assets in the asset list for the record. The criteria for an asset to be included in the asset list MUST be specified in the description of the count-for-asset-list attribute. An attribute SHALL NOT be identified as the ’count-for-asset-list’ attribute when an asset list is prohibited. At most one attribute on a record set SHALL be specified as the count-for-assetlist attribute.

Whether an asset list is required, permitted, or prohibited. If an asset list is required or permitted, it MUST indicate which attribute it is on the record that the list is associated with.

Whether attributes not explicitly declared in the record set type are permitted.

In the resources section of the ASR website, located at http://scap.nist.gov/specifications/asr/#resource- 1.0, an XSD provides a common XML format for representing a record set type. Record set types SHOULD be documented in the format defined by the XSD, though they are not required to be. See the XSD for details on documenting the record set type in that format.

Additionally, two Extensible Stylesheet Language Transformations (XSLTs) are provided on the resources section of the ASR website to support record set types documented in the above referenced XSD format. The first XSLT document converts record set type XML documents into a human readable HTML file. The second XSLT accepts an ASR document as input, along with a record set type as a parameter. The XSLT analyzes the ASR document against the provided record set type, and reports any errors that are discovered. Both of the XSLTs, along with the record set type XSD, are informative and are intended to assist with adopting ASR.

8 IANA Considerations

<IANA considerations text>

9 Security Considerations

As a data format, Asset Summary Reporting does not have security concerns that are known at this time. However, as a data format designed to be stored and transmitted between entities within an enterprise, the fact of the matter is that it SHOULD be used within a properly secured environment. Over time, a significant amount of information valuable to attackers can be gleaned from Asset Summary Reporting information. Therefore, it is recommended that use of

Asset Summary Reporting be performed in environments providing communication security mechanisms supplying the properties of confidentiality, data integrity, and non- repudiation.

10 References

10.1 Normative References

[ARF] NIST Interagency Report (IR) 7694 - Asset Reporting Format 1.1, June 2011. See: http://scap.nist.gov/specifications/arf/index.html

[Asset Identification] NIST Interagency Report (IR) 7693 - Asset Identification 1.1, May 2011. See: http://scap.nist.gov/specifications/ai/index.html

[Continuous Monitoring] NIST Special Publication (SP) 800-137 - Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations, January 2012. See: http://csrc.nist.gov/publications/PubsSPs.html#800-137

[CPE-N] NIST Interagency Report (IR) 7695 - Common Platform Enumeration: Naming Specification 2.3, August 2011. See: http://scap.nist.gov/specifications/cpe/naming.html#resource-2.3

[RFC 2119] Internet Engineering Task Force (IETF) Request for Comment (RFC) 2119: Key words for use in RFCs to Indicate Requirement Levels, March 1997. See: http://www.ietf.org/rfc/rfc2119.txt

[XCCDF] NIST Interagency Report (IR) 7275 - Specification for the Extensible Configuration Checklist Description Format 1.2, September 2011. See: http://scap.nist.gov/specifications/xccdf/#resource-1.2

[XML] W3C Recommendation Extensible Markup Language (XML) 1.0 (Fifth Edition), 26 November 2008. See: http://www.w3.org/TR/REC-xml/

[XSD] W3C Recommendation XML Schema, 28 October 2004. See: http://www.w3.org/XML/Schema.html

[XSD Qual] W3C Recommendation XML Schema Part 0: Primer Second Edition, 28 October 2004. See: http://www.w3.org/TR/xmlschema-0/

10.2 Informative References

TODO: Decide which references belong in the normative/informative sections

Appendix A Acknowledgements

Special thanks go to Mark Davidson, Adam Halbardier, and David Waltermire, and those others who participated in the definition of the Asset Summary Reporting version 1.0 [IR7848]

Appendix B Use Cases

This appendix documents some common use cases that were considered when developing ASR.

A.1 Continuous Monitoring

Information security continuous monitoring (ISCM) is defined as maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions. Organizational security status is determined using metrics established by the organization to best convey the security posture of an organization’s information and information systems, along with organizational resilience given known threat information. Among other requirements, ISCM tools must provide reporting with the ability to tailor output and drill down from high-level, aggregate metrics to systemlevel metrics, and allow for data consolidation into Security Information and Event Management (SIEM) tools and dashboard products.

ASR intends to meet the tool needs above by providing a vendor and technology neutral, and flexible reporting data model. ASR places few constraints on the type and nature of data that can be transported using its model, so many types of continuous monitoring data may be communicated using the model.

A.2 Security Automation

Security automation efforts (e.g. the Security Content Automation Protocol (SCAP)), are another use case for ASR. An example of a security automation need is a system administrator who needs to assess the 15,000 desktops on her network for compliance with the United States Government Configuration Baseline (USGCB), a government baseline for security settings. The system administrator runs the USGCB content against all desktops on her network, and receives one result file per desktop. Each result file contains, for each security check, a pass/fail indication, required setting, and actual setting. There are over 200 security settings checked per desktop. In order to avoid manually reviewing each result, the system administrator decides to use summary reporting. The system administrator creates three reports, detailed below. A sample record is provided for each report. Note that attributes with the 'example' prefix are defined only in the scope of this use case.

A report that shows the overall compliance percentage of desktops on the network. This summary report is used in a monthly dashboard that is presented to upper management. <asr:record example:compliance_pct="73"/>

Percentage compliance scores for each desktop on the network. This report is used to track perdesktop compliance trends and identify high-risk desktops. <asr:record example:hostname="desktop1" example:compliance_score="100"/>

Percentage compliance scores for the compliance of each security setting across the network. This report is used as a component in taking a risk-based approach to remediation. A bulletin of the top five non-compliant settings is generated monthly and sent to all employees as part of a security awareness campaign. <asr:record asr-attr:xccdf-benchmark="USGCB: Guidance for Securing Microsoft Windows 7 Systems for IT Professional" asr-attr:xccdf-profile= "united_states_government_configuration_baseline_version_1.2.0.0" asr-attr:xccdf-rule="minimum_password_length" example:compliance_score="75"/> Since the summary reports are in a standard format, they can be consumed by an application and presented in a meaningful manner. One meaningful presentation of this data is a list of security checks sorted from least compliant to most compliant. The system administrator has used summary reporting to increase the visibility and transparency of her security operations to management and end users, improved the accuracy and completeness of her data, and prioritized her highest value work.

Appendix C Integration with ARF

The Asset Reporting Format (ARF) is a data model for expressing the exchange format of information about assets and the relationships between assets and reports. The data model facilitates the reporting, correlating, and fusing of asset information throughout and between organizations. The intent of ARF is to provide a uniform foundation for the expression of reporting results, fostering more widespread application of sound IT management practices.

ARF has four primary components: assets, reports, report requests, and relationships. Assets, reports, and report requests exist in isolated buckets. The relationships component allows for explicit relationships between components (assets, report-requests, and reports) and uses a controlled vocabulary to do so. ASR reports may be captured as report objects in ARF. When an ASR is captured as an ARF report payload, a relationship MAY be established between the ASR report and the report request that caused the ASR to be generated. ARF 1.1 defines a relationship type arf-rel:createdFor in [ARF] Table 6-1. The arf-rel:createdFor relationship, established with the ASR report object as the subject and the report request object as the object, provides a well-defined connection between the request and response. Additional relationships may be defined between ASR reports and other reports as needed. Those relationships should be determined by the report creators and collaborators as needed.

Appendix D Record Set Example

This section shows an example scenario where an ASR report is created. To illustrate the record set concept, consider a scenario where the Chief Information Security Officer (CISO) of Example Corp wants to know the organization’s security posture relative to a recently published CVE (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2013). The following ASR document accurately reports an answer to the CISO’s question:

```
<asr:summary-report xmlns:ex="com.example"
xmlns:asr="http://scap.nist.gov/schema/asset-summary-reporting/1.0"
xmlns:asr-attr="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/attr"   page-number="1" last-page="true" report-
id="d1e1">  <asr:metadata timestamp="2011-11-08T14:27:44.97Z"/>
<asr:record-set id="asr:com.example:rset:1"    data-source-
ref="asr:com.example:dsrc:1"    record-set-type="ex:cve-report-
small">    <asr:record asr-attr:cve-id="CVE-2011-2013"      asr-
attr:inventory-finding="EXISTS" asr-attr:count="50"/>    <asr:record
asr-attr:cve-id="CVE-2011-2013"      asr-attr:inventory-
finding="NOT_EXISTS" asr-attr:count="170"/>    <asr:record asr-
attr:cve-id="CVE-2011-2013"      asr-attr:inventory-
finding="NOT_APPLICABLE" asr-attr:count="30"/>  </asr:record-set>
<asr:data-source id="asr:com.example:dsrc:1"
resource="VulnDb.abc.com" population-size="250"/> </asr:summary-
report>
```

Notice that the summary report uses attributes defined in the ASR Common Attributes schema. While record set types may use any XML attribute, it is preferable to leverage existing attributes defined in the ASR Common Attributes schema. This is one such example where existing attributes are useful. Since this vulnerability has a CVSS score of 10.0 and a rating of ’Critical’ has been given to the patch that fixes this vulnerability, the CISO forwards this information to the Windows Support Team. The CISO requests that the Windows Support Team apply the appropriate patch quickly. In order to honor the request, the Windows Support Team needs the report broken up by operating system and physical location, with a list of assets. With this information, the Windows Support Team will be able to efficiently deploy their resources for patching. The Windows Support Team requests the report with those additional details (only for systems that need to be patched). The following ASR document accurately reports the needed information. Asset listings have been truncated for simplicity.

```
<asr:summary-report  xmlns:ex="com.example" xmlns:ex-
attr="com.example.attr"
xmlns:asr="http://scap.nist.gov/schema/asset-summary-reporting/1.0"
xmlns:asr-attr="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/attr"  page-number="1" last-page="true" report-
id="d1e2">  <asr:metadata timestamp="2011-11-08T16:33:04.73Z"/>
<asr:record-set id="asr:com.example:rset:2" data-
sourceref="asr:com.example:dsrc:1" record-set-type="ex:cve-report-
informative"> <asr:record asr-attr:cve-id="CVE-2011-2013"
    asr-attr:inventory-finding="EXISTS"                 asr-
attr:count="30"                 asr-
attr:cpe="cpe:2.3:o:microsoft:windows_7:-:*:*:*:*:*:*:*"
   ex-attr:location="Miami">      <asr:identifier-list identifier-
system="com.example.asset-tag">       <asr:id>111111</asr:id>
<asr:id>222222</asr:id>        ...      </asr:identifier-list>
</asr:record> <asr:record asr-attr:cve-id="CVE-2011-2013"
    asr-attr:inventory-finding="EXISTS"                 asr-
attr:count="15"                 asr-
attr:cpe="cpe:2.3:o:microsoft:windows_7:-:*:*:*:*:*:*:*"
   ex-attr:location="Boston">      <asr:identifier-list identifier-
system="com.example.asset-tag">        <asr:id>333333</asr:id>
...      </asr:identifier-list>    </asr:record> <asr:record asr-
attr:cve-id="CVE-2011-2013"                 asr-attr:inventory-
finding="EXISTS"                 asr-attr:count="5"
asr-attr:cpe="cpe:2.3:o:microsoft:windows_2003_server:-
:*:*:*:*:*:*:* "                ex-attr:location="Miami">
<asr:identifier-list identifier-system="com.example.asset-tag>
... </asr:identifier-list>    </asr:record> <asr:record asr-attr:cve-
id="CVE-2011-2013"                 asr-attr:inventory-
finding="EXISTS"                 asr-attr:count="0"
asr-attr:cpe="cpe:2.3:o:microsoft:windows_2003_server:-
:*:*:*:*:*:*:* "                 asr-attr:location="Boston"/>
</asr:record-set>  <asr:data-source id="asr:com.example:dsrc:1"
resource="VulnDb.example.com" population-size="250"/> </asr:summary-
report>
```

This information can also be represented as a data table: TODO:

Recreate the table

With the above summary report, the Windows Support Team can deploy the necessary resources to the appropriate locations. With the asset list present in the identifier-list element, the resources deployed to Miami and Boston will be able to accurately and completely remediate the CVE. The report in this section is described in Appendix D.

In this report, Example Corp used attributes defined in the ASR Common Attributes Schema (cve-id, cpe, inventory-finding, and count) as well as an attribute that Example Corp defined, location.

The record sets used in this example have a record-set-type of ’cve-report-small’ and ’cve-reportinformative’, respectively. While this example uses two record-set-types, it is possible to use a single, more flexible record-set-type. Record-set-types may be flexible or restrictive, depending on the requirements of the entity that defines the record-set-type. Both examples are given in Appendix D

Appendix E Sample Record Set Type Definitions

This section describes an example of creating record set type definitions for the ASR reports described in Appendix C. Those reports leverage two record set types: ex:cve-report-small and ex:cve-reportinformative. The fictitious record set type cve-report-small is illustrated below. To demonstrate the flexibility of record set type definitions, a more permissive record set type is included in this section below the ex:cve-report-small example.

Record Set Type Name: {com.example}cve-report-small Description: To report on the number of computers affected by a CVE. Attributes

asr-attr:cve-id - MUST include. This is the CVE ID being reported on. Type: XML schema "string".

asr-attr:inventory-finding - MUST include. This is a status of the CVE for each asset in the count. Value must be one of "EXISTS", "NOT_EXISTS", "NOT_APPLICABLE", "NOT_REPORTED", "ERROR", or "UNKNOWN". Type: XML schema "string".

asr-attr:count - MUST include. Asset list is associated with this attribute. This count is the number of assets with the CVE related to the asset via the inventory-finding. Type: XML schema nonNegativeInteger.

Permit attributes not explicitly described here: no

Require asset list: not permitted

Require identifier list: not permitted

The record set type documented above may be more formally represented using the ASR record set type XML definition format:

```
<record-set-types xmlns="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/record-set-type"  xmlns:ex="com.example"
xmlns:asr="http://scap.nist.gov/schema/asset-summary-reporting/1.0"
xmlns:asr-attr="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/attr">  <record-set-type record-set-type-
qname="ex:cve-report-small" permitadditional-attributes="false">
<description> To report on the number of computers affected by a CVE.
    </description> <attributes>      <attribute attribute-qname="asr-
attr:cve-id" use="MUST"        description="This is the CVE ID being
reported on"/>      <attribute attribute-qname="asr-attr:inventory-
finding" use="MUST" description="This is a status of the CVE for each
asset in the count. Value must be one of ’EXISTS’, ’NOT_EXISTS’,
’NOT_APPLICABLE’, ’ NOT_REPORTED’, ’ERROR’, or ’UNKNOWN’." />
<attribute attribute-qname="asr-attr:count" use="MUST"
description="Asset list is associated with this attribute. This count
is the number of assets with the CVE related to the asset via the
inventoryfinding."/>    </attributes> <asset-listing use-identifier-
list="MUST NOT" use-asset-references="MUST NOT"/>  </record-set-type>
</record-set-types>
```

Given the definition above, the first report in Appendix C is able to be produced. The description above documents the following:

The record set type name "ex:cve-report-small".

The required attributes for the report. For each attribute, it specifies that the attribute must be

included. The attribute type is defined outside the record set type definition.

There are not any attributes permitted in addition to those specified here.

No form of asset listing is allowed.

The XML representation of the record set type definition (as described in Section 7) for ex:cve-reportinformative is illustrated below. <record-set-types xmlns="http://scap.nist.gov/schema/asset-summaryreporting/1.0/record-set-type" xmlns:ex="com.example" xmlns:asr="http://scap.nist.gov/schema/asset-summary-reporting/1.0"

```
xmlns:asr-attr="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/attr"  xmlns:ex-attr="com.example.attr">
<record-set-type record-set-type-qname="ex:cve-report-informative"
permitadditional-attributes="false">    <description> To report on
the number of computers affected by a CVE.     </description>
<attributes>      <attribute attribute-qname="asr-attr:cve-id"
use="MUST"        description="This is the CVE ID being reported
on"/>      <attribute attribute-qname="asr-attr:inventory-finding"
use="MUST" description="This is a status of the CVE for each asset in
the count. Value must be one of ’EXISTS’, ’NOT_EXISTS’,
’NOT_APPLICABLE’, ’ NOT_REPORTED’, ’ERROR’, or ’UNKNOWN’." />
<attribute attribute-qname="asr-attr:cpe" use="MUST" description="The
Common Platform Enumeration for the operating system of the
applicable assests." />      <attribute attribute-qname="ex:location"
use="MUST" description="The physical location of the applicable
assests." />      <attribute attribute-qname="asr-attr:count"
use="MUST" description="Asset list is associated with this attribute.
This count is the number of assets with the CVE related to the asset
via the inventoryfinding." count-for-asset-list="true"/>
</attributes>    <asset-listing use-identifier-list="MUST" use-asset-
references="MUST NOT"/>  </record-set-type> </record-set-types>
```

The description above documents the following:

The record set type name "ex:cve-report-informative".

The required attributes for the report. For each attribute, it specifies that the attribute must be included. The attribute type is defined outside the record set type definition.

There are not any attributes permitted in addition to those specified here.

Assets must be listed using the identifier-list method. It is possible for a record set type definition to be flexible enough that both reports in Appendix C could be produced in compliance with it. That record set type definition is illustrated below:

```
<record-set-types xmlns="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/record-set-type"  xmlns:ex="com.example"
xmlns:asr="http://scap.nist.gov/schema/asset-summary-reporting/1.0"
xmlns:asr-attr="http://scap.nist.gov/schema/asset-
summaryreporting/1.0/attr">  <record-set-type record-set-type-
qname="ex:cve-report-informative" permitadditional-attributes="true">
   <description> To report on the number of computers affected by a
CVE.     </description> <attributes>      <attribute attribute-
qname="asr-attr:cve-id" use="MUST"        description="This is the
CVE ID being reported on"/>      <attribute attribute-qname="asr-
```

attr:inventory-finding" use="MUST" description="This is a status of the CVE for each asset in the count. Value must be one of ’EXISTS’, ’NOT_EXISTS’, ’NOT_APPLICABLE’, ’ NOT_REPORTED’, ’ERROR’, or ’UNKNOWN’." />   <attribute attribute-qname="asr-attr:count" use="MUST" description="Asset list is associated with this attribute. This count is the number of assets with the CVE related to the asset via the inventoryfinding." count-for-asset-list="true"/> </attributes> <asset-listing use-identifier-list="OPTIONAL" use-asset-references="MUST NOT"/> </record-set-type> </record-set-types>

The description above documents the following:

The record set type name "ex:cve-report-informative".

That attributes beyond those listed in the record set type are permitted in record sets. Therefore, in the second example in Appendix C, "asr-attr:cpe" and "ex-attr:location" are permitted. This is known because of the value of "permit-additional-attributes".

The required attributes for the report. For each cve-id, inventory-finding, and count, it specifies that the attribute must be included.

The attribute associated with the asset list. In this case, "asr-attr:count" is the count associated with the asset listing.

The use of asset listings. In this case, the report may optionally use the identifier list (which the second report in Appendix C does), but it may not use the asset references element.

All of the information needed to properly format the aforementioned record sets is included in the record set type definitions given above.

This section contains a list of XML attributes defined in the ASR Common Attributes XSD located at http://scap.nist.gov/specifications/asr/#resource-1.0. These attributes are defined to provide a core of usable attributes with a common meaning across multiple areas. Each attribute is accompanied with a short description that describes what it means. Usage of this XSD and its associated attributes is RECOMMENDED, but not required.

Some attributes may require additional context in order to make sense. For example, the statistical ’count’ attribute has little meaning by itself. The context can be provided in the record-set-type definition through the report’s description of the attribute. For example, if a record-set-type definition had the following text in the attribute definition, the count would make sense: "Contains the count of employees that have completed the annual security awareness training."

All attributes in this section are defined in the following namespace: http://scap.nist.gov/schema/assetsummary-reporting/1.0/attr. Types specified with the prefix "xs:" are XML schema datatypes as defined in [XSD] Part 2. Unless otherwise specified, values for all attributes are restricted to xs:string. All pattern restrictions are XML schema regular expressions as defined in [XSD] Part 2, Section G

F.1 SCAP Attributes

This section contains SCAP attributes that are intended for use in reporting scenarios that involve SCAP data.

TODO: Create this table

F.3 Statistical Attributes

This section contains common statistical attributes with well-defined mathematical meanings.

TODO: Create this table

F.4 Other Attributes

This is a list of attributes that do not fit into another category. Some of these attributes exist to provide a common name in attempt to provide interoperability and may be further restricted as reporting scenarios dictate.

TODO: Create this table

E.2 Finding Attributes

This section contains attributes related to findings. Each finding attribute has values that allow the state of the finding to be indicated as well as attributes that allow for the indication of various non-finding conditions (error, not applicable, etc.)

TODO: Create this table

Authors’ Addresses

Mark Davidson

EMail: mdavidson@mitre.org

David Oliva
EMail: david.oliva@verizon.net