Change Proposals for SCAP 1.2

May 31, 2011

Table of Contents

Note: All proposed namespaces in this document are notional and subject to change.

Specification Revisions for SCAP 1.2

The table below lists the component specifications proposed for inclusion in SCAP 1.2, along with a list of SCAP 1.1 versions for comparison.

Spec. Abbr.	Component Specification Name	SCAP 1.1 Version	Proposed SCAP 1.2 Version	Proposal Numbers	Specification Document(s) for Proposed SCAP 1.2 Version
AI	Asset Identification	----	1.1	#1, #3	Draft NIST IR 7693
ARF	Asset Reporting Format	----	1.1	#1, #3	Draft NIST IR 7694
CCE	Common Configuration Enumeration	5	5		http://cce.mitre.org/lists/cce_list.html
CCSS	Common Configuration Scoring System	----	1.0	#9	NIST IR 7502
CPE	Common Platform Enumeration	2.2	2.3	#7	Draft NIST IRs 7695, 7696, 7697, and 7698
CVE	Common Vulnerabilities and Exposures	N/A	N/A		http://cve.mitre.org/cve/index.html
CVSS	Common Vulnerability Scoring System	2.0	2.0		NIST IR 7435
OCIL	Open Checklist Interactive Language	2.0	2.0		NIST IR 7692
OVAL	Open Vulnerability and Assessment Language	5.8	5.10	#8	http://oval.mitre.org/ (Draft for 5.10)
XCCDF	Extensible Configuration Checklist Description Format	1.1.4	1.1.5		Draft NIST IR 7275 Revision 4

In addition, SCAP 1.2 may leverage the W3C's XML Signature Syntax and Processing specification (second edition) (see proposals #4 and 5).
http://www.w3.org/TR/xmldsig-core/

1. Reporting Asset Identification information within an SCAP Datastream

Motivation

SCAP results often include OVAL system characteristics that help identify the target computer that the OVAL results are about. Traditionally, OVAL defines several fields to identify the target, including operating system name, host name, interfaces, etc. When used in SCAP, though, that same asset is also identified using the Asset Identification (AI) standard in the Asset Reporting Format (ARF). ARF and AI offer multiple higher-level capabilities that, when used with OVAL, will reduce the volume of information in SCAP reporting without sacrificing the level of detail within the results. The same capabilities can be applied to OCIL and XCCDF as well.

Proposed Solution

We propose enabling the linking of OVAL results to the ARF asset. Figure 1 provides a high-level diagram of how OVAL System Characteristics can link to an ARF asset.

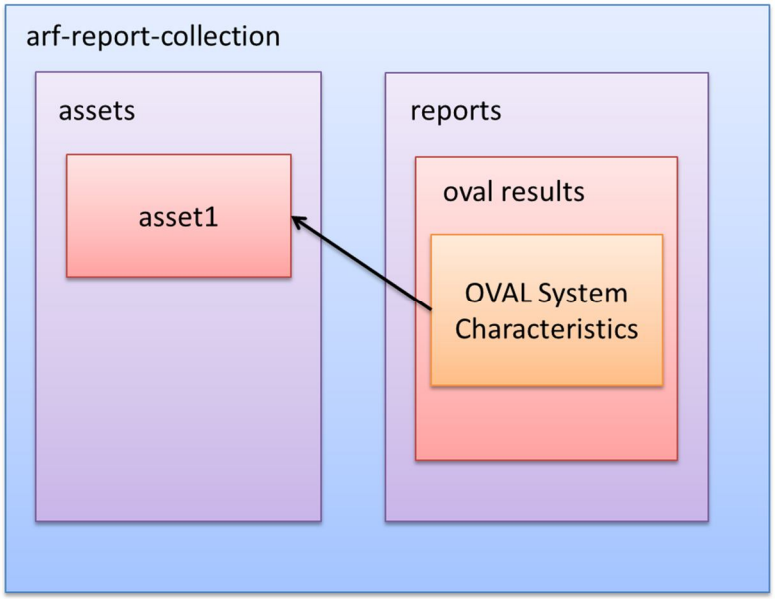

Figure 1 – OVAL Results Linking to an ARF Asset

In the diagram, the OVAL results are captured as an ARF report in an arf-report-collection. ARF provides an arf:object-ref element that allows reports in an ARF collection to point out to ARF objects (assets, other reports, etc.) By leveraging the arf:object-ref in OVAL System Characteristics, we can allow the system characteristics to point out to an asset in the ARF collection.

To accomplish this goal, we suggest leveraging the xs:any field in the OVAL System Characteristics. We will define a construct in SCAP that allows a choice between arf:object-ref and ai:asset-related. See the XML Schema below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://scap.nist.gov/schema/scap/1.2/constructs"
    elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.2-draft"
    xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:ai="http://scap.nist.gov/schema/asset-identification/1.1">
    <xs:import namespace="http://scap.nist.gov/schema/asset-reporting-format/1.1"
        schemaLocation="http://scap.nist.gov/schema/asset-reporting-format/1.1/asset-reporting-format_1.1.0.xsd"/>
    <xs:import namespace="http://scap.nist.gov/schema/asset-identification/1.1"
        schemaLocation="http://scap.nist.gov/schema/asset-identification/1.1/asset-identification_1.1.0.xsd"/>
    <xs:element name="target-asset">
        <xs:complexType>
            <xs:choice>
                <xs:element ref="arf:object-ref"/>
                <xs:element ref="ai:asset-related"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

The target-asset element can be embedded in the xs:any tag of an OVAL System Characteristics element. When the OVAL results are embedded in an ARF collection, the arf:object-ref can be used to point to an ARF asset. When the results are separated from an ARF collection, the ARF asset identification element referenced by the arf:object-ref can be embedded directly into the target-asset element.

This same methodology can be applied to XCCDF and OCIL results, assuming those XML schemas allow xs:any in the appropriate locations. We will work to incorporate this solution into the XCCDF and OCIL results payloads as well.

Examples

The first example below shows how OVAL results in an ARF collection can point out to an ARF asset.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<asset-report-collection xmlns="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:ai="http://scap.nist.gov/schema/asset-identification/1.1"
    xmlns:rc="http://scap.nist.gov/schema/reporting-core/1.1"
    xmlns:scapc="http://scap.nist.gov/schema/scap/1.2/constructs">
    <rc:relationships>
        ...
    </rc:relationships>
    <ai:assets>
        <ai:asset id="asset1">
            <ai:computing-device>
                <ai:connections>
                    <ai:connection>
                        <ai:ip-address>
                            <ai:ip-v4>192.168.222.1</ai:ip-v4>
                        </ai:ip-address>
                        <ai:mac-address>00-50-56-C0-00-01</ai:mac-address>
                    </ai:connection>
                </ai:connections>
                <ai:hostname>host.domain.tld</ai:hostname>
            </ai:computing-device>
        </ai:asset>
    </ai:assets>
    <reports>
        ...
        <report id="minimal-oval-res.xml">
            <content>
                <oval_results xmlns="http://oval.mitre.org/XMLSchema/oval-results-5">
                    ...
                    <results>
                        <system>
                            ...
                            <oval_system_characteristics
                                xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">
                                ...
                                <system_info>
                                    <os_name>unknown Professional Service Pack 2</os_name>
                                    <os_version>6.0.6002</os_version>
                                    <architecture>INTEL32</architecture>
                                    <primary_host_name>host.domain.tld</primary_host_name>
```

```
                                    <interfaces>
                                        <interface>
                                            <interface_name>VMware Virtual Ethernet Adapter for VMnet1</interface_name>
                                            <ip_address>192.168.222.1</ip_address>
                                            <mac_address>00-50-56-C0-00-01</mac_address>
                                        </interface>
                                    </interfaces>
                                    <scapc:target-asset>
                                        <arf:object-ref ref-id="asset1"/>
                                    </scapc:target-asset>
                                </system_info>
                                ...
                            </oval_system_characteristics>
                        </system>
                    </results>
                </oval_results>
            </content>
        </report>
        ...
    </reports>
</asset-report-collection>
```

The arf:object-ref points to asset1. This second example shows how the OVAL results would look in an SCAP context when they are not embedded in an ARF collection.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<oval_results xmlns="http://oval.mitre.org/XMLSchema/oval-results-5"
    xmlns:scapc="http://scap.nist.gov/schema/scap/1.2/constructs">
    ...
    <results>
        <system>
            ...
            <oval_system_characteristics
                xmlns="http://oval.mitre.org/XMLSchema/oval-system-characteristics-5">
                ...
                <system_info>
                    <os_name>unknown Professional Service Pack 2</os_name>
                    <os_version>6.0.6002</os_version>
                    <architecture>INTEL32</architecture>
                    <primary_host_name>host.domain.tld</primary_host_name>
                    <interfaces>
                        <interface>
                            <interface_name>VMware Virtual Ethernet Adapter for VMnet1</interface_name>
```

```
                        <ip_address>192.168.222.1</ip_address>
                        <mac_address>00-50-56-C0-00-01</mac_address>
                    </interface>
                </interfaces>
                <scapc:target-asset>
                    <asset-related xmlns="http://scap.nist.gov/schema/asset-identification/1.1" asset-ref="asset1">
                        <asset id="asset1">
                            <computing-device>
                                <connections>
                                    <connection>
                                        <ip-address>
                                            <ip-v4>192.168.222.1</ip-v4>
                                        </ip-address>
                                        <mac-address>00:50:56:C0:00:01</mac-address>
                                    </connection>
                                </connections>
                                <hostname>host.domain.tld</hostname>
                            </computing-device>
                        </asset>
                    </asset-related>
                </scapc:target-asset>
            </system_info>
            ...
        </oval_system_characteristics>
    </system>
  </results>
</oval_results>
```

In this example, asset1 is embedded directly into the OVAL results.

2. SCAP Source Datastream

Motivation

Previous versions of SCAP have defined an "SCAP bundle" that includes various XML files, often including XCCDF, OVAL, and CPE dictionary files. The physical representation of the bundle was not strictly defined in SCAP, but it often took the form of a ZIP file. SCAP used a file naming scheme to define the relationships between the files within a bundle using a defined convention.

Defining the SCAP bundles this way limits the options available to SCAP content creators. Specifically, a bundle could include at most a single XCCDF benchmark file, two OVAL files, a single OCIL file, and a single dictionary file for specific use cases. In addition, the relationships between the XCCDF, OVAL, OCIL and CPE dictionary files were hard-coded into the content; for example, XCCDF had to refer to an OVAL definition explicitly by file name and definition name, limiting reuse of content across related datastreams. Finally, because the physical nature of a bundle was not defined, nothing precluded different formats (e.g., ZIP, CAB) from being used. Naturally, this can cause some loss of content interoperability. Furthermore, use of the bundle methodology created technical challenges for supporting signing of content using XML Digital Signatures and using SCAP datastreams in various service technologies (e.g., SOAP Web Services, REST).

Proposed Solution

To overcome some of the shortcomings related to the "bundle" concept in previous versions of SCAP, we propose that a new SCAP source datastream collection format be created that defines how to represent SCAP version 1.2 source content. The use of the "bundle" concept will be deprecated to previous SCAP revisions. This is necessary to enable support for content signing.

The datastream-collection

The proposed datastream collection will be a single XML file made up of several different components, tied together by datastream packages. Figure 2 provides a high-level visual representation of the proposal.

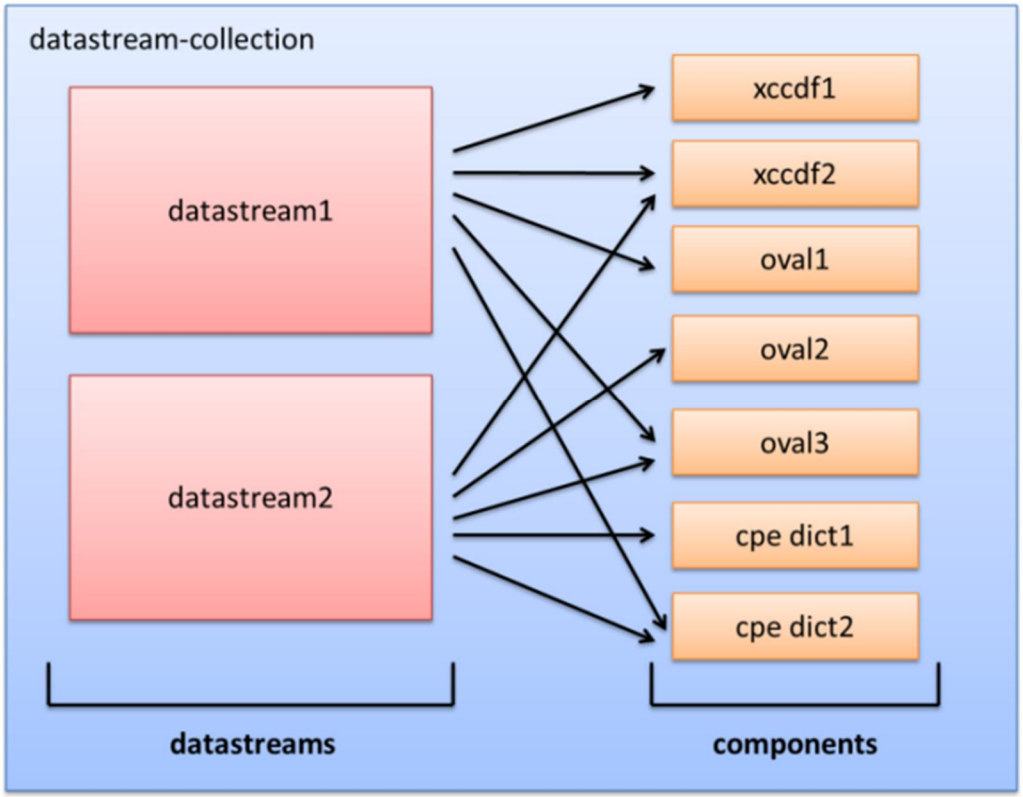

Figure 2 – datastream-collection

As shown in Figure 2, a datastream collection is composed of one or more datastreams and an arbitrary number of components. Each datastream points (i.e. links) to the components that compose that datastream. This approach provides multiple advantages:

- Multiple SCAP source datastreams can be represented in a single package.
- Components are easily reusable across datastreams.
- Components and datastreams can be added and removed from the collection in a modular fashion.
- A well-defined format for the SCAP source content eliminates ambiguity of the physical representation of the content.
- A single XML file approach enables content validation options such as XML Schema and Schematron validation.
- It offers the ability to handle any arbitrary number of checklist, check, and dictionary components.

- It allows use of multiple OVAL versions simultaneously.
- A single XML file approach reduces complexity in creating, validating and managing XML Digital Signatures.
- It reduces the complexity of using SCAP datastreams in service implementations (e.g., SOAP Web Services, REST).
- In an environment using dynamic tasking commands, SCAP content can be built on-the-fly based on the needs of a task.

The datastream

In order to increase interoperability and reuse of content, the datastream links will contain mappings between the hrefs that content components specify, and the components in the datastream collection that those hrefs correspond to. Figure 3 shows this graphically.

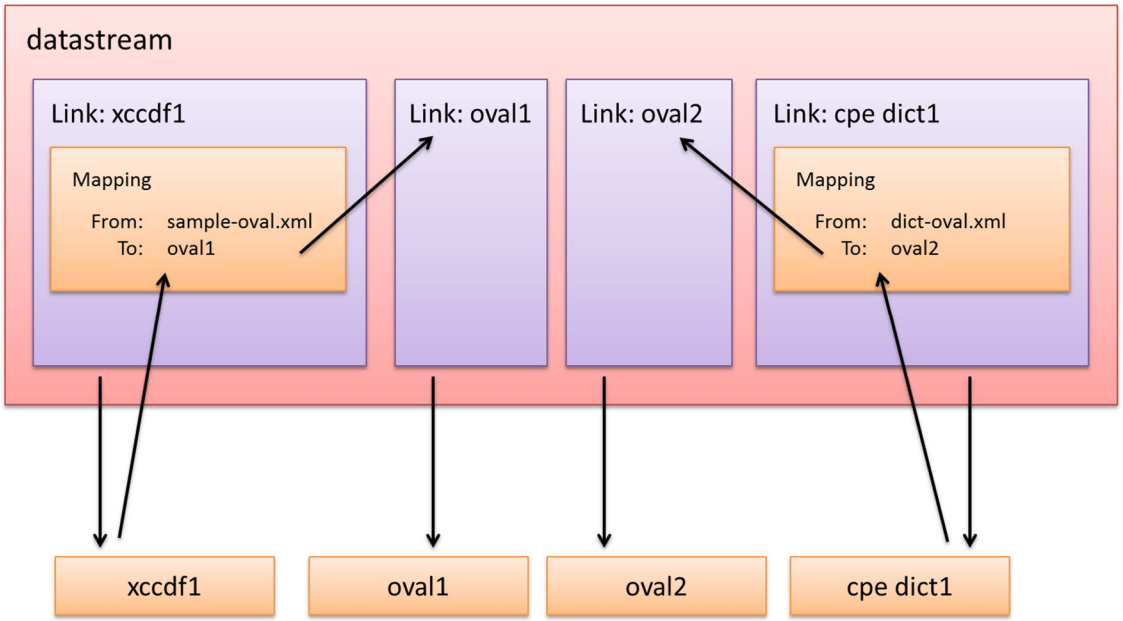

Figure 3 – datastream details

The datastream shown in Figure 3 links to four components, labeled xccdf1, oval1, oval2, and cpe dict1.

The mapping in the xccdf1 link indicates that when the XCCDF has an href to sample-oval.xml (e.g., `<xccdf:check-content-ref href="sample-oval.xml" name="oval:gov.nist.test.compliance:def:1"/>`), then dereference that URI to the component linked to by the oval1 link, using the mapping, if any, specified in the oval1 link.

The mapping specified in the cpe dict1 link functions the same way. When cpe dict1 references dict-oval.xml (e.g., `<check system="http://oval.mitre.org/XMLSchema/oval-definitions-5" href="dict-oval.xml">oval:gov.nist.test.inventory:def:1</check>`), then dereference that href to the component linked to by the oval2 link with the mappings, if any, specified by the oval2 link.

An advantage of this mapping facility is that the component content is completely agnostic of the higher SCAP content. The SCAP specification will now specify the dereferencing rules, which means that components are completely reusable. Since the hrefs in components are not lexically tied to the IDs in the datastream, the hrefs may be defined once and then mapped in the context of the SCAP datastreams where that component is used. In addition, mapping gives the SCAP content creator the flexibility to assign IDs to the components as he or she sees fit, without being bound to the hrefs specified in other content. This mapping solution isolates the concerns of interlinking components at the SCAP logical level.

The link

There are two options for linking from a datastream. The first type is a local link. This is the most common type and was used in all of the previous examples. A local link points to a component in the current datastream-collection. The second type of link is a remote link. Remote links reference content outside of the current datastream-collection. This is helpful in situations where content can be commonly accessed and need not travel with the datastream collection. The most obvious example is remote patches-up-to-date content from previous versions of SCAP. The improvement for this iteration of SCAP is that any content may now be remote.

There are a few important notes regarding this functionality: 1) remote links must be resolvable by the tool evaluating the content, and 2) the ability to trust the remote content can be managed via the digital signing mechanisms presented in a separate proposal for SCAP 1.2. Also note that remote references SHALL NOT failover to a local reference in cases where the remote content is not accessible. In cases where the remote content is not resolvable, an error SHALL be generated.

An Example

Presented below is a sample datastream-collection.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<datastream-collection xmlns="http://scap.nist.gov/schema/scap/source/1.2"
    id="datastream-collection-1">
    <datastream id="datastream1" scap-version="1.2" timestamp="2011-05-09T09:48:55" use-case="CONFIGURATION">
        <checklists>
            <component-ref id="xccdf1" href="sample-xccdf">
                <deref-map key="oval_check_source" href="oval1"/>
```

```
            </component-ref>
        </checklists>
        <checks>
            <component-ref id="oval1" href="sample-oval"/>
            <component-ref id="cpe-oval1" href="sample-cpe-oval"/>
        </checks>
        <dictionaries>
            <component-ref id="cpe-dict1" href="sample-cpe-dictionary">
                <deref-map key="cpe_dict_checks" href="cpe-oval1"/>
            </component-ref>
        </dictionaries>
    </datastream>
    <component id="sample-xccdf">
        <Benchmark xmlns="http://checklists.nist.gov/xccdf/1.1">
           ...
           <check-content-ref href="oval_check_source" name="oval:gov.nist.test.compliance:def:1"/>
           ...
        </Benchmark>
    </component>
    <component id="sample-oval">
        <oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5">
           ...
        </oval_definitions>
    </component>
    <component id="sample-cpe-oval">
        <oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5">
            ...
        </oval_definitions>
    </component>
    <component id="sample-cpe-dictionary">
        <cpe-list xmlns="http://cpe.mitre.org/dictionary/2.0">
            ...
            <check system="http://oval.mitre.org/XMLSchema/oval-definitions-5"
href="cpe_dict_checks">oval:gov.nist.test.inventory:def:1</check>
            ...
        </cpe-list>
    </component>
</datastream-collection>
```

3. SCAP Result Datastream

Motivation

Previous versions of SCAP have not been sufficiently detailed regarding the results format of an SCAP scan. They have specified the relationship between the result XCCDF, OVAL, and OCIL files, but they never defined a physical format for representing all of the results in a single package. Defining a structured results format allows better validation and interoperability of results data, enables tools and operators to digitally sign SCAP results, and allows web services to be defined around the SCAP specification. In support of Continuous Monitoring and other initiatives, conformance to NIST guidelines regarding asset reporting is also desirable.

Proposed Solution

We propose an SCAP results format that leverages the existing Asset Reporting Format (ARF). Figure 4 shows the high-level structure of the proposed report.

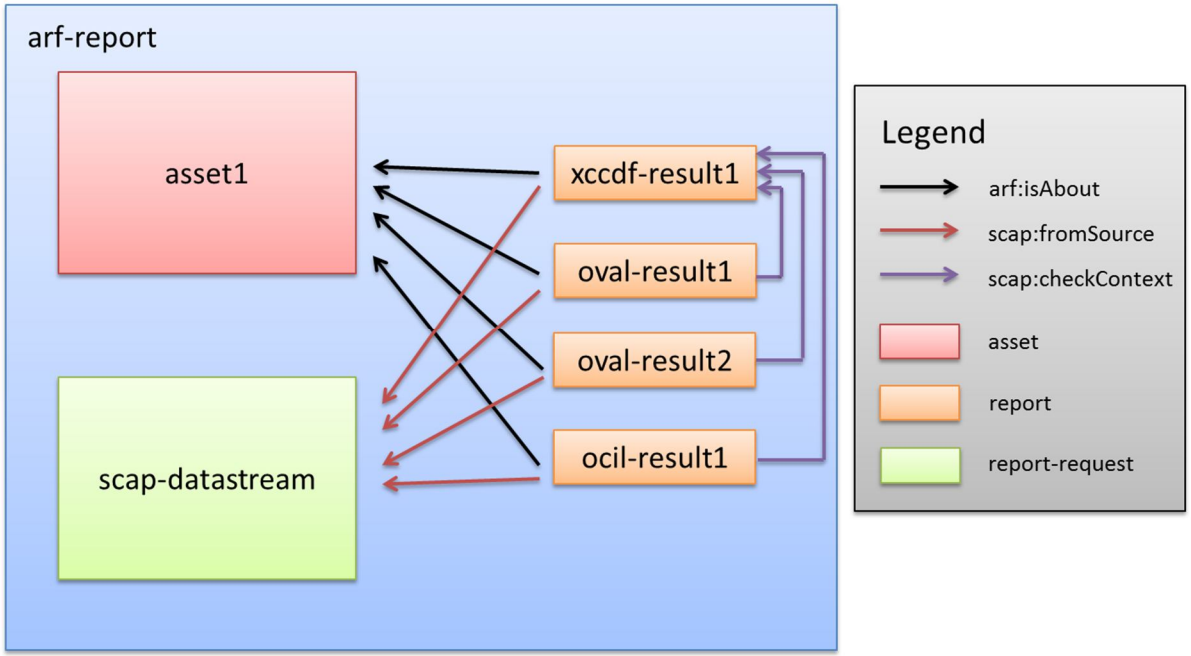

Figure 4 – ARF-based SCAP Results

In Figure 4, the various outputs from an SCAP scan (e.g., XCCDF, OVAL, and OCIL results) comprise separate report payloads in an arf-report-collection. The source SCAP datastream can optionally be captured via the ARF report-request mechanism, and the asset that the scan was run against is identified using Asset Identification (AI). Then, using the relationship capability built into ARF, relationships will be defined between the individual report payloads, the asset, and the source datastream. The payloads can be related to the asset using the arf:isAbout relationship defined in the ARF specification, the payloads can be related to the source datastream using an SCAP defined relationship (e.g., scap:fromSource), and the check results can be associated to the XCCDF results using an SCAP defined relationship (e.g., scap:checkContext). Inter-component relationship (i.e. references from within the components to other components) will continue to be handled similarly to how they were handled in previous versions of SCAP. The only difference is that the references will resolve to the report IDs in the arf-report-collection instead of to a filename.

This solution allows SCAP to begin reporting results in a standard way that is flexible and robust. The advantage of using ARF is that many SCAP result sets can be combined into a single ARF report without loss or duplication of data. The results from a single scap-datastream run against any number of hosts may be captured in a single or multiple ARF reports, as needed for the specific case. In addition, the entire report can be digitally signed, as well as used as a response in a web service infrastructure.

An Example

Presented below is a sample arf-report-collection holding an scap-datastream result package.

```
<?xml version="1.0" encoding="UTF-8"?>
<asset-report-collection xmlns="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:ai="http://scap.nist.gov/schema/asset-identification/1.1"
    xmlns:rc="http://scap.nist.gov/schema/reporting-core/1.1"
    xmlns:rel="http://scap.nist.gov/vocabulary/arf/relationships/1.0#"
    xmlns:rel-scap="http://scap.nist.gov/vocabulary/scap/relationships/1.0#">
    <rc:relationships>
        <rc:relationship subject="minimal-xccdf-res.xml" type="rel:isAbout">
            <rc:ref>target1</rc:ref>
        </rc:relationship>
        <rc:relationship subject="minimal-oval-res.xml" type="rel:isAbout">
            <rc:ref>target1</rc:ref>
        </rc:relationship>
        <rc:relationship subject="minimal-cpe-oval-res.xml" type="rel:isAbout">
            <rc:ref>target1</rc:ref>
        </rc:relationship>
        <rc:relationship subject="minimal-xccdf-res.xml" type="rel-scap:fromSource">
            <rc:ref>target1</rc:ref>
        </rc:relationship>
        <rc:relationship subject="minimal-oval-res.xml" type="rel-scap:fromSource">
```

```
            <rc:ref>target1</rc:ref>
        </rc:relationship>
        <rc:relationship subject="minimal-cpe-oval-res.xml" type="rel-scap:fromSource">
            <rc:ref>target1</rc:ref>
        </rc:relationship>
    </rc:relationships>
    <report-requests>
        <report-request id="scap-datastream1">
            <datastream-collection xmlns="http://scap.nist.gov/schema/scap/source/1.2">
                ...
            </datastream-collection>
        </report-request>
    </report-requests>
    <assets>
        <asset id="target1">
            <ai:computing-device>
                <ai:connections>
                    <ai:connection>
                        <ai:ip-address>
                            <ai:ip-v4>127.0.0.1</ai:ip-v4>
                            <ai:ip-v6>0:0:0:0:0:0:0:1</ai:ip-v6>
                        </ai:ip-address>
                    </ai:connection>
                </ai:connections>
                <ai:hostname>host.domain.tld</ai:hostname>
            </ai:computing-device>
        </asset>
    </assets>
    <reports>
        <report id="minimal-xccdf-res.xml">
            <content>
                <Benchmark xmlns="http://checklists.nist.gov/xccdf/1.1">
                    ...
                    <TestResult>
                        ...
                    </TestResult>
                </Benchmark>
            </content>
        </report>
        <report id="minimal-oval-res.xml">
            <content>
                <oval_results xmlns="http://oval.mitre.org/XMLSchema/oval-results-5">
                    ...
                    <results>
                        ...
```

```
                    </results>
                </oval_results>
            </content>
        </report>
        <report id="minimal-cpe-oval-res.xml">
            <content>
                <oval_results xmlns="http://oval.mitre.org/XMLSchema/oval-results-5">
                    ...
                    <results>
                        ...
                    </results>
                </oval_results>
            </content>
        </report>
    </reports>
</asset-report-collection>
```

4. Signing SCAP Source Datastream

Motivation

SCAP content is intended to express the formal policies and checks for an organization. Those policies and checks are often prescribed by an authoritative and trusted group for the organization, and are mandated for use by the compliance group. Therefore, content fidelity and trust are of utmost importance to ensure that the authoritative and mandated content is being run. It is essential that security professionals, auditors, and others can verify the source of content and determine if the content is authorized.

Proposed Solution

Figure 5 provides a high-level diagram of a proposed SCAP datastream collection.

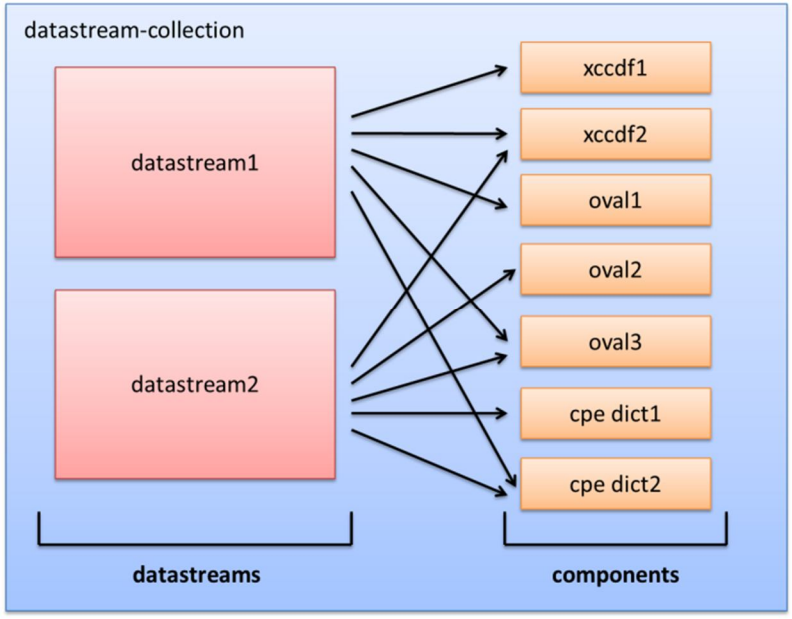

Figure 5 – datastream-collection

Given the proposed SCAP datastream model (see Proposal 2), we propose that zero or more digital signatures may be appended as the last children in a datastream-collection. Each signature will sign the content of a datastream element, as well as sign a manifest of all of the

components that the datastream element refers to, in accordance with the digital trust specification. Figure 6 shows the composition of the signature.

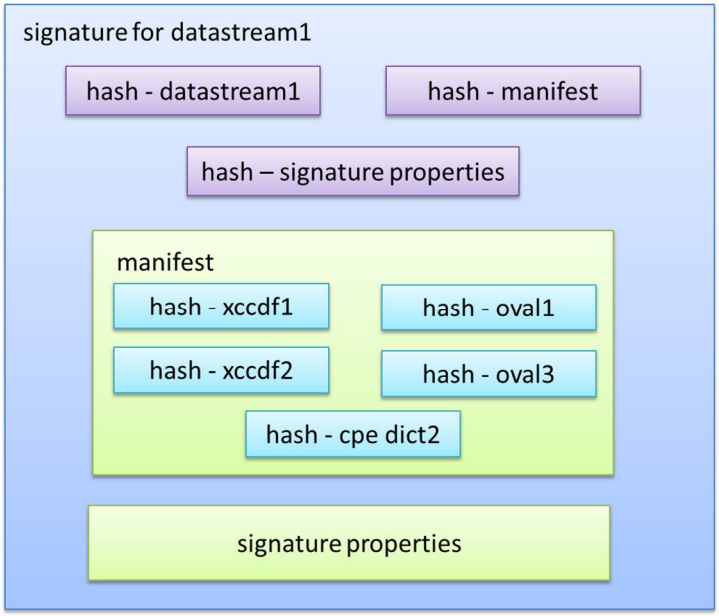

Figure 6 – XML Digital Signature for SCAP Source Datastream

The proposed XML signature will follow the recommendations set by W3C regarding XML digital signatures, as well as the NIST recommendations in the digital trust specification. Three "references" (hashes), shown in purple, will be defined.

- The first reference is to the datastream element being signed. The datastream element contains all of the references and mappings to components in the datastream-collection.
- The second reference is to a manifest that contains "references" (hashes) of all of the components linked to by the datastream element being defined.
- The third reference is to a signature properties element that contains metadata related to the signature, as defined by the NIST digital trust specification.

Each reference contains a hash and metadata about how the hash was computed, in accordance with the W3C XML digital signatures specification. The signature also optionally contains the certificate information and public key used to verify the references. Finally, the signature contains the actual signed value.

In order to verify the signature, each reference in the signature (datastream1, manifest, signature properties) is verified using the hash in the reference and the metadata about how the hash was computed. When verifying the manifest reference, it only confirms that the text of the manifest is valid; verifying the references in the manifest is a separate step that may or may not be done at the same time as the signature validation. Once the public key is confirmed to be from a trusted source, the signature is validated and the manifest references are validated; then the datastream source content can be trusted.

An Example

Presented below is an example of signing an SCAP source datastream.

```
<?xml version="1.0" encoding="UTF-8"?>
<ds:datastream-collection id="datastream-collection-1">
   <ds:datastream id="sample.zip" scap-version="1.2" timestamp="2011-04-13T09:48:55" use-case="CONFIGURATION">
      <ds:checklists>
         <ds:component-ref href="sample-xccdf.xml" id="xccdf1">
            <ds:deref-map href="oval1" key="oval_check_source"/>
         </ds:component-ref>
      </ds:checklists>
      <ds:checks>
         <ds:component-ref href="sample-oval.xml" id="oval1"/>
         <ds:component-ref href="sample-cpe-oval.xml" id="cpe-oval1"/>
      </ds:checks>
      <ds:dictionaries>
         <ds:component-ref href="sample-cpe-dictionary.xml" id="cpe-dict1">
            <ds:deref-map href="cpe-oval1" key="cpe_dict_checks"/>
         </ds:component-ref>
      </ds:dictionaries>
   </ds:datastream>
   <ds:component id="sample-xccdf.xml">
      <Benchmark>
         ..
      </Benchmark>
   </ds:component>
   <ds:component id="sample-oval.xml">
      <oval_definitions>
         ...
      </oval_definitions>
```

```
    </ds:component>
    <ds:component id="sample-cpe-oval.xml">
       <oval_definitions>
          ...
       </oval_definitions>
    </ds:component>
    <ds:component id="sample-cpe-dictionary.xml">
       <cpe-list>
          ...
       </cpe-list>
    </ds:component>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" Id="dsig-20110513123455287-70">
       <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
          <Reference URI="">
             <Transforms>
                <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                   <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect"
                      >//ds:datastream-collection[@id = 'datastream-collection-1']/ds:datastream[@id =
                      'sample.zip']</XPath>
                </Transform>
             </Transforms>
             <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
             <DigestValue>syeWAJqJIAPoXNDBIayJqu4jiDDjJrwJvN1XDBkbMxA=</DigestValue>
          </Reference>
          <Reference Type="http://www.w3.org/2000/09/xmldsig#Manifest" URI="#manifest-20110513123455278-63">
             <Transforms>
                <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
             </Transforms>
             <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
             <DigestValue>rD+Jnh0cfl3dkFnkApe5cZlk3KvPl7COG7jYT3heIP8=</DigestValue>
          </Reference>
          <Reference Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties" URI="#sig-prop-20110513123455287-74">
             <Transforms>
                <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
             </Transforms>
             <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
             <DigestValue>WNLM8fje++tCGksdiSWQYuT4jRivr7xRp9AWkeaOUgQ=</DigestValue>
          </Reference>
       </SignedInfo>
       <SignatureValue>nD2NO3H4oSEDlJO4jZd/uvPVBCqHkDtGeKE36ZzJzVM0dYJQy+RUFVzWwwywTd4xiiwQtPUyRiTk
             lWKuMIowXqQz5+7e+fNsQFFsnHkbsJialujbgb/L8g9CtSmsY4mzoNYkiQ5jomYd+phoSRdRcY2G
             h+tVFPfPlN+1k4Z8aj3Wq6LSNAHB8j6mb02ASsT1yvv9WwmPYglDDJA6XyN9mWCT9gNLNB4DXfF4
```

```
            6oohrkGMP25WiqYO+yiemZT2Yf99AGWNkj5+DFZEiDH0x0Cqz4//lHA+c5+RZJfNo8w/moTas+OM
            iQterVodG/Rbx8g9UqyGTFdftClLJ8mTCySjVw==</SignatureValue>
         <KeyInfo>
            <KeyValue>
               <RSAKeyValue>
                  <Modulus>z8adrX9m0S8OxIxN+fui33wiz4ZYgb4xPbR9MS5pOp1A8kVpH5Ew3N6O3/dMs2a4diIxyGLVh0r8
                           6QXWH/W6T2IC2ny+hi+jWRwXrvgTY3ZAFgePvz2OdRhVN/cUbOto4Pa4I2mVZWW+/Q0Fn7YpqPBD
                           DxlGq/xyFPuYq/4y7Y+Ah+vHO2ZSaiQjbj8F38XrGhwlcbFVyK8AmxK3z0zWwX86uMEqVCjW6s6j
                           2KAWdbAjEpgZHlJY87i/DqnFgxfmdg3oru+YeiEPVRY8hyQpYbtgryveZOHTgnCHmS/53U9jSS0c
                           yb/ADuj1upfyNoOiMMgQr7Olhc5pTvuWAl4Fnw==</Modulus>
                  <Exponent>AQAB</Exponent>
               </RSAKeyValue>
            </KeyValue>
         </KeyInfo>
         <Object>
            <Manifest Id="manifest-20110513123455278-63">
               <Reference URI="">
                  <Transforms>
                     <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                     <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect"
                           >//ds:datastream-collection[@id = 'datastream-collection-1']/ds:component[@id =
                           'sample-xccdf.xml']</XPath>
                     </Transform>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <DigestValue>dFAFnjs0eivUlu+0bPnqVKqeiDs1zAhzEUNURm783jI=</DigestValue>
               </Reference>
               <Reference URI="">
                  <Transforms>
                     <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                     <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect"
                           >//ds:datastream-collection[@id = 'datastream-collection-1']/ds:component[@id =
                           'sample-oval.xml']</XPath>
                     </Transform>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <DigestValue>CTsxOrw4acE+S09z/8E018GuuaYWPsBPBtaL3i0/Cxg=</DigestValue>
               </Reference>
               <Reference URI="">
                  <Transforms>
                     <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                     <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect"
```

```
                        >//ds:datastream-collection[@id = 'datastream-collection-1']/ds:component[@id =
                        'sample-cpe-oval.xml']</XPath>
                     </Transform>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <DigestValue>g37CcIk4DvN3BiPmFnXKzGnHl6GcE8+MMqVONVmV6sM=</DigestValue>
               </Reference>
               <Reference URI="">
                  <Transforms>
                     <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                     <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2" Filter="intersect"
                           >//ds:datastream-collection[@id = 'datastream-collection-1']/ds:component[@id =
                           'sample-cpe-dictionary.xml']</XPath>
                     </Transform>
                  </Transforms>
                  <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                  <DigestValue>90MZMPkPkHtNBp6Oy1iPjQ3yGmsVxeZO3ykvIT4selw=</DigestValue>
               </Reference>
            </Manifest>
         </Object>
         <Object>
            <SignatureProperty Id="sig-prop-20110513123455287-74" Target="#dsig-20110513123455287-70">
               <dsig:signature-info xmlns:dsig="http://scap.nist.gov/xml-dsig/1.0">
                  <dsig:timestamp>2011-05-13T12:34:55-0700</dsig:timestamp>
                  <dsig:nonce>04EED3035045C9E7</dsig:nonce>
               </dsig:signature-info>
            </SignatureProperty>
         </Object>
      </Signature>
</ds:datastream-collection>
```

5. Signing SCAP Result Datastream

Motivation

When reporting on the results of an SCAP scan, it is important to establish a trusted relationship between the person and/or tool receiving and relying on the results, and the person and/or tool that generated the results. In addition, it is important to be able to verify that the results reported are for the expected source SCAP content being run against the identified target(s). All of these problems can be solved in the content by providing a mechanism to digitally sign the results of an SCAP datastream scan.

Proposed Solution

Figure 7 provides a high-level diagram of the structure of an ARF report.

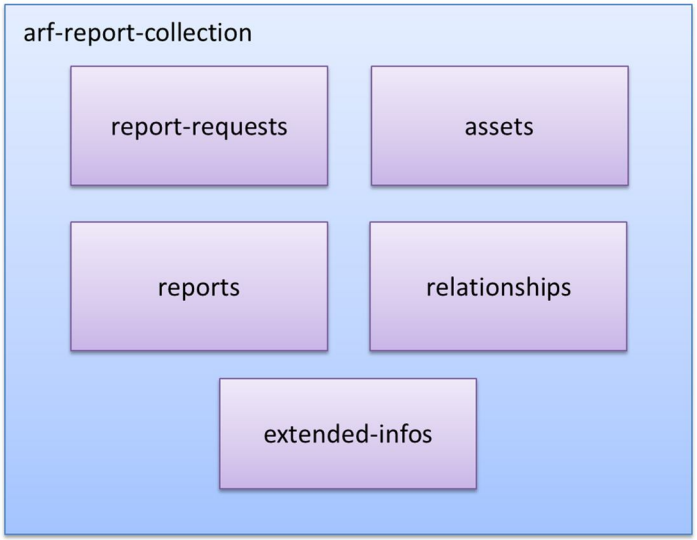

Figure 7 – ARF report structure

As discussed in a different proposal (Proposal 3), the ARF structure can house the results of an SCAP scan, as well as the source SCAP content that was used during the scan. The extended-infos section of ARF allows for the housing of any XML content. In this proposal, we suggest using the extended-infos section to house an optional XML digital signature. The signature would sign the root node of the arf-report collection, excluding the extended-info element that contains the signature itself. In cases where countersigning the content is required (e.g., a tool signs

the content produced, then a person countersigns the content before sending it back to the requestor), an enveloping signature would be created to sign the original signature. The new signature would be placed into the ARF document in a new extended-info element, and the original signature and extended-info element would be removed (i.e. the old signature is now contained inside the new signature, so it no longer needs to live alone in the ARF report). Once the public key(s) provided are confirmed to be from a trusted source and the signature (or chain of nested signatures in the case of counter-signing) is verified, then the result content can be trusted, especially when the source content is included in the ARF report.

An Example

Presented below is an example of signing an SCAP result datastream.

```
<?xml version="1.0" encoding="UTF-8"?>
<asset-report-collection xmlns="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:ai="http://scap.nist.gov/schema/asset-identification/1.1" id="report-collection-1">
    <rc:relationships>
        ...
    </rc:relationships>
    <assets>
        ...
    </assets>
    <reports>
        ...
    </reports>
    <extended-infos>
        <extended-info id="dsig-20110513123500733-1">
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" Id="dsig-20110513123500737-55">
                <SignedInfo>
                    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                    <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference URI="">
                        <Transforms>
                            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                            <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                                <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2"
                                    xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
                                    xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Filter="subtract"
                                    >//arf:asset-report-collection[@id = 'report-collection-1']/arf:extended-
infos[count(arf:extended-info[dsig:Signature]) = count(*)]</XPath>
                                <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2"
                                    xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
                                    xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Filter="subtract"
                                    >//arf:asset-report-collection[@id = 'report-collection-1']/arf:extended-
```

```
infos/arf:extended-info[dsig:Signature]</XPath>
                            </Transform>
                        </Transforms>
                        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <DigestValue>08xYNFJvmv3c7rWJn0eNmdG6nky1VSA3m35LEnJuLsU=</DigestValue>
                    </Reference>
                    <Reference Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties"
                        URI="#sig-prop-20110513123500738-29">
                        <Transforms>
                            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                        </Transforms>
                        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <DigestValue>ZZjPgYF/jqSctFVIJhXQ2nxS0pKk1jyH9QzlWCjDxR8=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>HGTl5kIg5585NH6J9xNyZ+HI6A+2aW0WDao1/bRzjFossuveYimr5EVbFOhZbZG2l1F9mxzCYU6Q
                    evOzDcEn5S7n2SeTMXaLQ58rj5iDZS/BtvObM7K3CdxLcTi348/RyQZ0ddaRlKYfW89GQq6eckWa
                    92SP9NmG3MGY+TgfrwaDrwHkiW6slsgsChccqtP/qT5drgGYDbIC0SC5gvlVsOQOWknzKBHueDab
                    AYd8deV1W5zLPmMRm94TrBCcY5tOsGIZUcKOGTjfhYNOYFCBWmeObxqQfT/HCkawkTgpDik9GQV5
                    gApZDi+5FhyXYp3fosT/xFD7csUbVInKPdvL8A==</SignatureValue>
                <KeyInfo>
                    <KeyValue>
                        <RSAKeyValue>
                            <Modulus>z8adrX9m0S8OxIxN+fui33wiz4ZYgb4xPbR9MS5pOp1A8kVpH5Ew3N6O3/dMs2a4diIxyGLVh0r8
                                6QXWH/W6T2IC2ny+hi+jWRwXrvgTY3ZAFgePvz2OdRhVN/cUbOto4Pa4I2mVZWW+/Q0Fn7YpqPBD
                                DxlGq/xyFPuYq/4y7Y+Ah+vHO2ZSaiQjbj8F38XrGhwlcbFVyK8AmxK3z0zWwX86uMEqVCjW6s6j
                                2KAWdbAjEpgZHlJY87i/DqnFgxfmdg3oru+YeiEPVRY8hyQpYbtgryveZOHTgnCHmS/53U9jSS0c
                                yb/ADuj1upfyNoOiMMgQr7Olhc5pTvuWAl4Fnw==</Modulus>
                            <Exponent>AQAB</Exponent>
                        </RSAKeyValue>
                    </KeyValue>
                </KeyInfo>
                <Object>
                    <SignatureProperty Id="sig-prop-20110513123500738-29" Target="#dsig-20110513123500737-55">
                        <dsig:signature-info xmlns:dsig="http://scap.nist.gov/xml-dsig/1.0">
                            <dsig:timestamp>2011-05-13T12:35:00-0700</dsig:timestamp>
                            <dsig:nonce>1852648DD522EF0D</dsig:nonce>
                        </dsig:signature-info>
                    </SignatureProperty>
                </Object>
            </Signature>
        </extended-info>
    </extended-infos>
</asset-report-collection>
```

The following example shows how to countersign content that has already been signed.

```
<?xml version="1.0" encoding="UTF-8"?>
<asset-report-collection xmlns="http://scap.nist.gov/schema/asset-reporting-format/1.1"
    xmlns:ai="http://scap.nist.gov/schema/asset-identification/1.1"
    xmlns:rc="http://scap.nist.gov/schema/reporting-core/1.1"
    id="report-collection-1">
    <rc:relationships>
        ...
    </rc:relationships>
    <ai:assets>
        ...
    </ai:assets>
    <reports>
        ...
    </reports>
    <extended-infos>
        <extended-info id="dsig-20110513123500819-63">
            <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" Id="dsig-20110513123500824-98">
                <SignedInfo>
                    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                    <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                    <Reference Type="http://www.w3.org/2000/09/xmldsig#Object" URI="#obj-cs-20110513123500817-90">
                        <Transforms>
                            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                        </Transforms>
                        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <DigestValue>34JrTS8pUcHmpVhFPzJZFfMXwAO07sojWEBj/L6qdRs=</DigestValue>
                    </Reference>
                    <Reference Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties"
                        URI="#sig-prop-20110513123500824-67">
                        <Transforms>
                            <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                        </Transforms>
                        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                        <DigestValue>AeUhn/lah1GLXlzbqCAloJHIeuSrVVgkbpSBc4tF1kg=</DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>ZEYPFVb/nnW6ev4NnotEYMjcHkOjhIoayaK+qf4E2u2tk5ph0F6nkfuPDoT+4lpjIwRDzAow+Zin
                    LBkGHNYcFixEtZGo6shLxj5Io3z0vmNgOe7HQBknfHqJp/WyTtxfjFsU4SGxllXCpXJAxY7lv3Ni
                    uz7Welu4Awl7/xTfUjQFUUd9nKZzDmwcAWcFy7okiYJH6yBNJMOJSzDXDrYNFX9XmZGGjNMh0Dl8
                    DaMFQVKrAiaWXR5O88xrHfThKCLebEm3v3XOUffhOLIPFxR9kvIc8ENbwSsjiWzUFFEjmjM+lIf4
                    KLvHd3bC0p1Q6FD4WwhnyE/FlrHmt1ffrXAjGQ==</SignatureValue>
                <KeyInfo>
                    <KeyValue>
```

```
                        <RSAKeyValue>
                            <Modulus>z8adrX9m0S8OxIxN+fui33wiz4ZYgb4xPbR9MS5pOp1A8kVpH5Ew3N6O3/dMs2a4diIxyGLVh0r8
                                6QXWH/W6T2IC2ny+hi+jWRwXrvgTY3ZAFgePvz2OdRhVN/cUbOto4Pa4I2mVZWW+/Q0Fn7YpqPBD
                                DxlGq/xyFPuYq/4y7Y+Ah+vHO2ZSaiQjbj8F38XrGhwlcbFVyK8AmxK3z0zWwX86uMEqVCjW6s6j
                                2KAWdbAjEpgZHlJY87i/DqnFgxfmdg3oru+YeiEPVRY8hyQpYbtgryveZOHTgnCHmS/53U9jSS0c
                                yb/ADuj1upfyNoOiMMgQr7Olhc5pTvuWAl4Fnw==</Modulus>
                            <Exponent>AQAB</Exponent>
                        </RSAKeyValue>
                    </KeyValue>
                </KeyInfo>
                <Object Id="obj-cs-20110513123500817-90">
                    <Signature Id="dsig-20110513123500737-55">
                        <SignedInfo>
                            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                            <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                            <Reference URI="">
                                <Transforms>
                                    <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                                    <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2"
                                            xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
                                            xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Filter="subtract"
                                            >//arf:asset-report-collection[@id = 'report-collection-1']/arf:extended-
infos[count(arf:extended-info[dsig:Signature]) = count(*)]</XPath>
                                        <XPath xmlns="http://www.w3.org/2002/06/xmldsig-filter2"
                                            xmlns:arf="http://scap.nist.gov/schema/asset-reporting-format/1.1"
                                            xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Filter="subtract"
                                            >//arf:asset-report-collection[@id = 'report-collection-1']/arf:extended-
infos/arf:extended-info[dsig:Signature]</XPath>
                                    </Transform>
                                </Transforms>
                                <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                                <DigestValue>08xYNFJvmv3c7rWJn0eNmdG6nky1VSA3m35LEnJuLsU=</DigestValue>
                            </Reference>
                            <Reference Type="http://www.w3.org/2000/09/xmldsig#SignatureProperties"
                                URI="#sig-prop-20110513123500738-29">
                                <Transforms>
                                    <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
                                </Transforms>
                                <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                                <DigestValue>ZZjPgYF/jqSctFVIJhXQ2nxS0pKk1jyH9QzlWCjDxR8=</DigestValue>
                            </Reference>
                        </SignedInfo>
                        <SignatureValue>HGTl5kIg5585NH6J9xNyZ+HI6A+2aW0WDao1/bRzjFossuveYimr5EVbFOhZbZG2l1F9mxzCYU6Q
                            evOzDcEn5S7n2SeTMXaLQ58rj5iDZS/BtvObM7K3CdxLcTi348/RyQZ0ddaRlKYfW89GQq6eckWa
```

```
                            92SP9NmG3MGY+TgfrwaDrwHkiW6slsgsChccqtP/qT5drgGYDbIC0SC5gvlVsOQOWknzKBHueDab
                            AYd8deV1W5zLPmMRm94TrBCcY5tOsGIZUcKOGTjfhYNOYFCBWmeObxqQfT/HCkawkTgpDik9GQV5
                            gApZDi+5FhyXYp3fosT/xFD7csUbVInKPdvL8A==</SignatureValue>
                        <KeyInfo>
                            <KeyValue>
                                <RSAKeyValue>

<Modulus>z8adrX9m0S8OxIxN+fui33wiz4ZYgb4xPbR9MS5pOp1A8kVpH5Ew3N6O3/dMs2a4diIxyGLVh0r8
                                        6QXWH/W6T2IC2ny+hi+jWRwXrvgTY3ZAFgePvz2OdRhVN/cUbOto4Pa4I2mVZWW+/Q0Fn7YpqPBD
                                        DxlGq/xyFPuYq/4y7Y+Ah+vHO2ZSaiQjbj8F38XrGhwlcbFVyK8AmxK3z0zWwX86uMEqVCjW6s6j
                                        2KAWdbAjEpgZHlJY87i/DqnFgxfmdg3oru+YeiEPVRY8hyQpYbtgryveZOHTgnCHmS/53U9jSS0c
                                        yb/ADuj1upfyNoOiMMgQr7Olhc5pTvuWAl4Fnw==</Modulus>
                                    <Exponent>AQAB</Exponent>
                                </RSAKeyValue>
                            </KeyValue>
                        </KeyInfo>
                        <Object>
                            <SignatureProperty Id="sig-prop-20110513123500738-29" Target="#dsig-20110513123500737-55">
                                <dsig:signature-info xmlns:dsig="http://scap.nist.gov/xml-dsig/1.0">
                                    <dsig:timestamp>2011-05-13T12:35:00-0700</dsig:timestamp>
                                    <dsig:nonce>1852648DD522EF0D</dsig:nonce>
                                </dsig:signature-info>
                            </SignatureProperty>
                        </Object>
                    </Signature>
                </Object>
                <Object>
                    <SignatureProperty Id="sig-prop-20110513123500824-67" Target="#dsig-20110513123500824-98">
                        <dsig:signature-info xmlns:dsig="http://scap.nist.gov/xml-dsig/1.0">
                            <dsig:timestamp>2011-05-13T12:35:00-0700</dsig:timestamp>
                            <dsig:nonce>022D398BEDE9A0F4</dsig:nonce>
                        </dsig:signature-info>
                    </SignatureProperty>
                </Object>
            </Signature>
        </extended-info>
    </extended-infos>
</asset-report-collection>
```

6. Legacy Support for SCAP Content Processing

Motivation

The SCAP 1.1 specification contains the following requirement for legacy SCAP content support: "Products supporting SCAP 1.1 SHALL process SCAP 1.0 content as described under the SCAP 1.0 version of NIST SP 800-126." [SP 800-126r1, Section 4.1]

As the number of SCAP versions increases, it becomes a larger burden for tools (especially new tools) to support all of the previous content and processing requirements. It is beneficial for tools to provide support for legacy SCAP content, but it should not be mandatory.

Proposed Solution

We propose that SCAP 1.2 alter the above requirement to wording similar to the following:

"Products SHALL process SCAP 1.2 content, MAY process SCAP 1.1 content, and MAY process SCAP 1.0 content.

Products that process legacy SCAP content (1.1, 1.0) SHALL process it as defined under the corresponding version of NIST SP 800-126 (for SCAP 1.1, SP 800-126 revision 1; for SCAP 1.0, the original SP 800-126).

Products that process legacy SCAP content MUST be capable of outputting results in the same SCAP version as the source content, and MAY convert the legacy SCAP results into a newer SCAP results version."

7. Use of CPE 2.3

Motivation

SCAP 1.1 and 1.0 use CPE version 2.2. The latest CPE version, 2.3, offers new features that would be very useful for SCAP to leverage.

For example, the CPE 2.3 Language specification adds support for referencing checks, allowing evaluation of system state as part of determining applicability. This will allow SCAP-validated products to determine what software or services are installed at runtime. This will help to eliminate, in some cases, false positive results that are caused by using static lists of configuration settings that must be assessed regardless of configuration posture.

Another example is the formatted string binding defined in the CPE Naming 2.3 specification. It looks similar to the URI binding (CPE 2.2 style), but because it's defined as a formatted string instead of a URI, the requirements that typically apply to URIs (as specified in RFC 3986) are relaxed. Also, the formatted string binding supports the inclusion of several attributes that were not supported in CPE version 2.2.

Proposed Solution

We propose using the CPE 2.3 specifications (Naming, Matching, Dictionary, and Language) for SCAP 1.2. The CPE 2.3 dictionary specification will be leveraged to define the CPE dictionary components in SCAP 1.2. This will directly replace the CPE 2.2 dictionary used in previous versions of SCAP. Also, formatted string bindings (as defined in the CPE Naming 2.3 specification) will be used to represent CPE names in SCAP 1.2.

8. OVAL Version Update

Motivation

Updates to OVAL have added functionality and fixed problems. SCAP should take advantage of this.

Proposed Solution

We propose that SCAP 1.2 support OVAL 5.3 - OVAL 5.10.

The following language proposed for the SCAP 1.2 specification is identical to the SCAP 1.1 specification, only changing the OVAL version numbers (highlighted):

1. While the default version of OVAL used in SCAP 1.2 SHALL be OVAL version 5.10, content authors SHOULD utilize the earliest SCAP-supported version of OVAL (5.3 at minimum) that includes all required tests and is necessary to properly address the content's purpose or use case. This approach, often referred to as the "least-version-principle", allows for SCAP content to remain viable over a longer period of time by allowing for the broadest support within products, while reducing the content maintenance burden that would be required to maintain revisions of content for multiple specification versions.
2. Products supporting OVAL SHALL support OVAL Definition documents written against OVAL versions 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, and 5.10.
3. Each OVAL result data stream component SHALL validate against version 5.10 of the OVAL Results schema regardless of the version of the OVAL Definitions document that was evaluated.

9. Common Configuration Scoring System (CCSS) Addition

Motivation

SCAP 1.1 references the Common Vulnerability Scoring System (CVSS), which provides measures and scores of relative severity for software flaw vulnerabilities (CVEs). In the past year, the Common Configuration Scoring System (CCSS) specification was finalized, which provides a measurement and scoring capability for system security configuration issues (CCEs).

See NIST IR 7502 for the CCSS specification.
http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7502

Proposed Solution

We propose that CCSS support be added to SCAP 1.2, as a parallel to the CVSS support.

1. Products MAY use CCSS base, temporal, and environmental scores and vectors.
2. Products SHOULD use CCSS scores acquired dynamically in place of static CCSS scores in the `@weight` attribute within XCCDF configuration setting-related rules.

10. SCAP Schematron Validation

Motivation

Previous versions of SCAP have used a Schematron rule set as part of the SCAP Content Validation Tool to validate SCAP content. However, these rules have not been published.

Proposed Solution

We propose that the Schematron rules written for SCAP 1.2 be published along with the XML Schemas. These Schematron rules should be used to confirm well-formattedness of SCAP content. These Schematron rules will be documented. In addition, while referencing the official XML schema and Schematron files in their official locations, we propose that we post the XCCDF, OVAL, and OCIL XML schemas and Schematrons as well. In some cases these may need to be modified to support the SCAP specification and to fix errors; we will document those modifications.

11. Additional XML Examples

Motivation

There have been requests to add more XML examples to the SCAP specification. This would help to illustrate certain requirements or conventions that may be difficult to understand otherwise. However, XML examples can be quite long, so adding several could make the specification longer and harder to navigate; in addition, long examples may be hard to understand.

Proposed Solution

We propose that SCAP 1.2 add a small number of XML examples based on community recommendations for which areas of the specification would be most improved through examples.

Also, we would alter or drop any existing XML examples that are not effective.