SCAP Content Validation Tool

Harold Booth

NIST

Agenda

- Why do we need it?
- What does it do?
- How can it help?
- How do you use it?
- Where is the tool going?

Why do we need it?

- SCAP Content creators need to know if the content they are writing can be processed by SCAP products
- Content consumers need a way to know if content will work in their tools
- Product vendors need to know if they should be able to process a data stream (i.e. valid according to NIST SP 800-126)
- Validation must be automated

What does it do?

- Validates SCAP 1.0 data streams
- Checks that the requirements defined in NIST SP 800-126 are satisfied by the content
- Validates that:
  - Content is well-formed
  - Cross-references are valid
  - Required values are appropriately set

Validation Process

1. Verifies that provided files are appropriate for the use case
2. Schema validation
3. OVAL Schematron validation
   - Minor changes to default OVAL Schematron
4. If necessary, combines all files in the data stream into a single XML file
5. SCAP requirements Schematron validation

How is it already helping?

- Identified ambiguous requirements in the NIST SP 800-126 document
- Improved FDCC and USGCB content
- Improved confidence that content written will run in an SCAP product
- Encouraged more rigor in the content creation workflow in order to avoid “the wrath of the validation tool”
- Used by the National Checklist Program as an automated way to determine if content may be classified Tier III

How can it help content creators?

- Use to verify that content conforms to NIST SP 800-126 to increase confidence it will run in an SCAP validated product
- Help to increase rigor in content development processes
- Informative list of requirements in one place
- Encourages best practices

How can it help content consumers?

- Verify that provided SCAP content is acceptable prior to running in a product
  - Help diagnose content errors when content does not run correctly within a tool
- Improve confidence in content and products

How can it help tool vendors?

- If a data stream passes validation then a validated product should be able to process the data stream
  - Exception to this would be OVAL tests for platforms the tool does not run on
- An informative list of requirements
- Code is available upon request

How do you use it?

- Requires JRE 1.6 or later
- Command-line tool
- Download from http://scap.nist.gov site
- Current version for SCAP 1.0 may be found at http://scap.nist.gov/revision/1.0/index.html#tools
- Read scapval.html contained within the zip bundle to get started
- Download NIST SP 800-126 at http://csrc.nist.gov/publications/PubsSPs.html#SP-800-126

Command-line options

- Required
  - file or dir – input data stream to process
  - usecase – the use case of the data stream
    - CONFIGURATION
    - VULNERABILITY_XCCDF_OVAL
    - VULNERABILITY_OVAL
    - SYSTEM_INVENTORY
- Optional
  - online – allows the tool to access the internet
  - debug, quiet, version, and batch

Requirements matrix

- Located in the scap-val-requirements-1.0.html file
- Contains the requirements from NIST SP 800-126 extracted into a tabular format
- Each requirement is given an identifier
- Grouped by use case with requirements applying to all use cases grouped into “General”

Requirements Matrix Example

Requirement ID	800-126 Section	800-126 Statement	800-126 Derived Requirement	Checked?	Requirement Type	Error Level	Requirement Category	Notes
1	4.1	An SCAP Benchmark document validates against the XCCDF schema (http://nvd.nist.gov/scap/xccdf/docs/xccdf-1.1.4.xsd) and conforms to all relevant content requirements as outlined in the XCCDF Specification [QUI08].	For all SCAP XCCDF documents a validating parse must be run with no errors prior to performing any other processing.	true	SCHEMA	ERROR	SOURCE_CONTENT	

Requirements Matrix Explained

- Requirement ID – this is the requirement identifier; output by the tool in the results file as a cross-reference into the matrix
- 800-126 Section – the section number where the requirement could be found
- 800-126 Statement – the statement in NIST SP 800-126 containing the requirement
- 800-126 Derived Requirement – a restatement of the requirement as the item or items which should be checked

Requirements Matrix Explained (cont'd)

- Checked?
  - true – the tool is checking for this requirement
  - false – the tool is not checking or is unable to check for the requirement
- Requirement Type
  - APPLICATION – the tool either checks or imposes the requirement
  - SCHEMA – requirement is checked through schema validation
  - SCHEMATRON – requirement is checked through Schematron validation
  - NOT_CHECKED – requirement is not checked

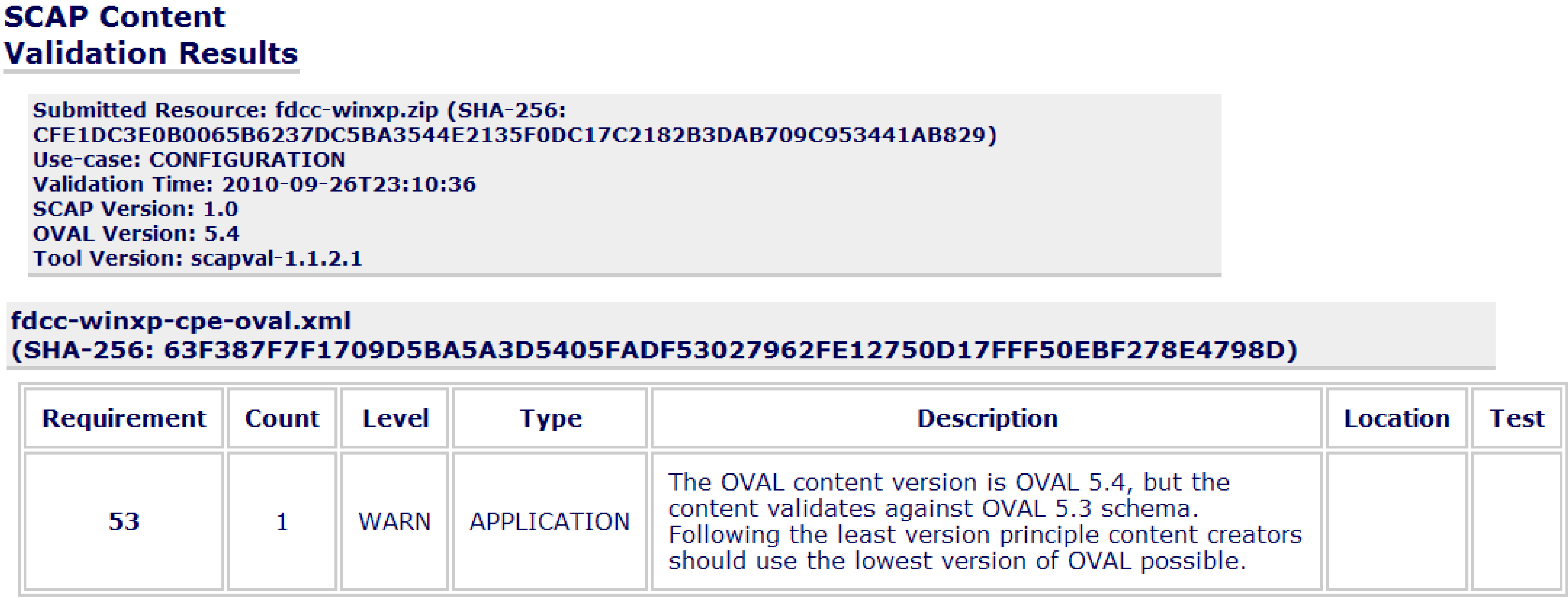
Requirements Matrix Explained (cont’d)

- Error Level
  - ERROR – the data stream must be fixed in order to pass validation
  - WARNING – the data stream passed validation but a best practice or a suggestion has not been followed
- Requirement Category – whether the requirement applies specifically to the input data stream, the results or an SCAP tool
- Notes – any additional comments

Results Files

- By default two results files are created
  - scap-validation-result.xml
  - scap-validation-result.html
- A log file is also created
  - scap-validation.log

Example Result



SCAP Content Validation Results

Submitted Resource: fdcc-winxp.zip (SHA-256: CFE1DC3E0B0065B6237DC5BA3544E2135F0DC17C2182B3DAB709C953441AB829)
Use-case: CONFIGURATION
Validation Time: 2010-09-26T23:10:36
SCAP Version: 1.0
OVAL Version: 5.4
Tool Version: scapval-1.1.2.1

fdcc-winxp-cpe-oval.xml
(SHA-256: 63F387F7F1709D5BA5A3D5405FADF53027962FE12750D17FFF50EBF278E4798D)

Requirement	Count	Level	Type	Description	Location	Test
53	1	WARN	APPLICATION	The OVAL content version is OVAL 5.4, but the content validates against OVAL 5.3 schema. Following the least version principle content creators should use the lowest version of OVAL possible.		

Anatomy of the result files

- Requirement – the requirement identifier; this is a cross-reference into the requirements matrix
- Count – the number of times this item occurred
- Level – whether it was a WARNING or an ERROR
- Type – one of SCHEMATRON, SCHEMA, or APPLICATION

Anatomy of the result files (cont’d)

- Description – the “800-126 Derived Requirement” from the requirements matrix
- Location – the XPATH location where the error was triggered
- Test – the Schematron test (if applicable) for the requirement

Requirement	Count	Level	Type	Description	Location	Test
A17	2	ERROR	SCHEMATRON	CCE-3867-0 - CCE number is in an invalid format or the check-digit does not match. It should be of format CCE-XXXX-X or CCE-XXXXX-X where each X is a digit, and the final X is a check-digit.	/*:Benchmark[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][4]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Rule[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:ident[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1] /*:Benchmark[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][4]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Rule[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:ident[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][2]	if((@system eq 'http://cce.mitre.org' or @system eq 'CCE') and matches(., '^CCE-\d4-\d$')) then (sum(for $j in (for $i in reverse(string-to-codepoints(concat(substring(.,5, 4),substring(.,10,1))))[position() mod 2 = 0] return ($i - 48) * 2, for $i in reverse(string-to-codepoints(concat(substring(.,5, 4),substring(.,10,1))))[position() mod 2 = 1] return ($i - 48)) return ($j mod 10, $j idiv 10)) mod 10) eq 0 else true()

Requirement	Count	Level	Type	Description	Location	Test
74	2	WARN	SCHEMATRON	CCE-3867-0 - Generate a warning for all CCE references that are not in the Official CCE dictionary.	/*:Benchmark[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][4]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Rule[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:ident[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1] /*:Benchmark[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][4]/*:Group[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][1]/*:Rule[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][6]/*:ident[namespace-uri()='http://checklists.nist.gov/xccdf/1.1'][2]	if(@system eq 'http://cce.mitre.org') then exists(document(concat($datafiles_directory,'/nvdcce-0.1-feed.xml'))/nvd-config:nvd/nvd-config:entry[@id eq current()]) else true()

Where is the tool going?

- Support for SCAP 1.1
- Add the ability to check results files for correctness
- Lower the learning curve for the tool

Acknowledgments

- Development Team
  - Adam Halbardier
  - Harold Owen
- Early Users
  - Kurt Dillard
  - Tim Harrison
  - Matt Kerr
  - Jim Ronayne
  - Shane Shaffer

Questions & Answers / Feedback

Harold Booth

Computer Scientist

National Institute of Standards and Technology (NIST)

harold.booth@nist.gov