Security Content Automation Protocol (SCAP)

SCAP Vendor Assertions Document v3.4

April 2, 2014

by

Red Hat

314 Littleton Rd.

Westford, MA 01886

http://www.redhat.com

For

OpenSCAP

Version 1.0.8

Assertion:

Red Hat, Inc. asserts that OpenSCAP version 1.0.8 meets or exceeds the Derived Test Requirements (DTR) for SCAP 1.2 as described in NIST IR 7511 Revision 3 for the following SCAP capabilities and supported platform family:

Capabilities:
- ✚ Authenticated Configuration Scanner
- ✚ CVE
- ❑ OCIL

Platform Family:
- ❑ Microsoft Windows (XP Pro, Vista, Win 7-32 bit, Win 7-64 bit) Family
- ✚ Red Hat Enterprise Linux 5 Desktop (32 bit and 64 bit edition) Family

SCAP Component Technologies:

The following table provides a brief summary of the individual SCAP Component Standards supported by OpenSCAP:

Supported	Component	Version	Description
✚	AI	1.1	Asset Identification (AI) is a specification for identifying assets
✚	ARF	1.1	The Asset Reporting Format (ARF) is a specification describing a data model for asset reporting
✚	CCE	5	The Common Configuration Enumeration ™ (CCE) is a nomenclature and dictionary of software security configurations
✚	CCSS	1.0	The Common Configuration Scoring System (CCSS) is a specification for measuring the relative severity of system security configuration issues
✚	CPE	2.3	The Common Platform Enumeration (CPE) is a specification measuring the relative severity of system security configuration issues
✚	CVE	n/a	The Common Vulnerability Enumeration® (CVE) is a specification describing a nomenclature and dictionary of security-related software

			flaws
✚	CVSS	2.0	The Common Vulnerability Scoring System is a language for representing system configuration information, assessing machine state, and reporting assessment results
❑	OCIL	2.0	The Open Checklist Interactive Language (OCIL) is a language for representing checks that collect information from people or from existing data stores made by other data collection efforts
✚	OVAL	5.10.1	The Open Vulnerability and Assessment Language is a language for representing system configuration information, assessing machine state, and reporting assessment results
✚	SCAP	1.2	SCAP is a specification for expressing and manipulating security data in standardized ways. SCAP uses several individual specifications in concert to automate continuous monitoring, vulnerability management, and security policy compliance evaluation reporting
✚	TMSAD	1.0	The trust Model for Security Automation Data (TMSAD) describes a common trust model that can be applied to specifications within the security automation domain
✚	XCCDF	1.2	Extensible Configuration Checklist Description Format (XCCDF) is a specification language for writing security checklists, benchmarks, and related kinds of documents

SCAP Implementation Statement(s):

OpenSCAP provides a library to create SCAP related tools as well as a multipurpose tool, oscap. The oscap program is a command line tool that allows users to load, scan, validate, transform, and export SCAP documents. The oscap tool can take as input eXtensible Checklist Configuration Document Format (XCCDF) and Open Vulnerability and Assessment Language (OVAL) XML files or Data Streams.

There are a number of command line options that allow the user to refine the way in which the provided input file should be evaluated. For example, the selection of a specific profile from the input XCCDF file, a Data Stream ID from input provided in Data Stream format, or alternate CPE dictionaries and tailoring files to use during evaluation just to mention a few. The user may direct the program to output XCCDF results in well formed XML, HTML, or text. Additional output options include Asset Reporting Format (ARF), OVAL, and low level check engine results. Besides performing a scan, oscap can create a security checklist guide from the XCCDF content.

The HTML scan report created by oscap has several parts. The beginning of the report gives some basic information about the content used, then basic information about the target of the scan, and followed by an overall scoring of the system. An overview of the scan with the title of the check and the result of it is given. The title is a hyperlink that takes you to detailed information about that check.

The OpenSCAP library is composed of multiple API's one each for the SCAP standard being implemented and can be programmed from C or Python. The OVAL implementation includes standalone helper applications (probes) for each OVAL test supported. The probes are called by the library and perform one task. This makes the resulting scanner SE Linux policy friendly. The data collected by the probe is marshalled and made available to the calling library by means of an “S-expressions” API. The library uses POSIX compliant system calls and the GNU autoconf/automake system so that it is portable to a wide range of platforms and operating systems. The library will build as much of itself as the target platform supports. For Red Hat Enterprise Linux, this means the library, and by extension the oscap utility, fully

supports i386, x86_64, PPC, IA64, as well as S390 based systems.

SCAP Backwards Compatibility:

OpenSCAP natively supports the older SCAP 1.1 and 1.0 specifications. It does this by detecting the version of OVAL or XCCDF specified in the content and then processing it based on the selected OVAL probes. The user does not need to do anything special, support is automatic. The 1.0 support includes using:

- XCCDF version 1.1.4
- OVAL version 5.3
- CCE version 5
- CPE version 2.2
- CVSS version 2
- CVE

The SCAP 1.1 support includes:

- XCCDF version 1.1.4
- OVAL version 5.8
- CCE version 5
- CPE version 2.2
- CVSS version 2
- CVE

For both SCAP 1.0 and 1.1, the support in OVAL is for the following schemas: Linux, Unix, Core, Common, and Independent schemas. Within the Linux schema, there is no support for the dpkginfo or slackwarepkginfo tests.

Disclaimer: