National Institute of
Standards and Technology
U.S. Department of Commerce

# Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2 (Draft)

## Recommendations of the National Institute of Standards and Technology

Stephen Quinn
Karen Scarfone
David Waltermire

**NIST Special Publication 800-117**
**Revision 1 (Draft)**

# Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2 (Draft)

*Recommendations of the National Institute of Standards and Technology*

**Stephen Quinn**
**Karen Scarfone**
**David Waltermire**

## COMPUTER SECURITY

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

January 2012

**U.S. Department of Commerce**

John Bryson, Secretary

**National Institute of Standards and Technology**

Patrick D. Gallagher,
Under Secretary for Standards and Technology
and Director

## Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

**National Institute of Standards and Technology Special Publication 800-117 Revision 1 (Draft)**
**Natl. Inst. Stand. Technol. Spec. Publ. 800-117 rev. 1, 26 pages (Jan. 2012)**

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

## Acknowledgments

The authors, Stephen Quinn of the National Institute of Standards and Technology (NIST), Karen Scarfone of Scarfone Cybersecurity, and David Waltermire of NIST wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content, especially John Banghart of NIST and Lon Kight of G2, Inc.

The authors would like to acknowledge the contributors to the original version of this publication, particularly Matthew Barrett of G2, Inc. and Chris Johnson of HP Enterprise Services, co-authors of the original version; Tim Grance of NIST; Kimberly Watson, Lieutenant Colonel Joe Wolfkiel, Paul Bartock, and Dan Schmidt of Department of Defense; Kevin Sitto, Brad Wood, and Greg Witte of G2; Andy Bove of Secure Elements; Scott Armstrong of Symantec; and Rob Hollis of ThreatGuard.

## Trademark Information

## Table of Contents

## List of Tables

# Executive Summary

Managing the security of systems throughout an enterprise is challenging for several reasons. Most organizations have many systems to patch and configure securely, with numerous pieces of software (operating systems and applications) to be secured on each system. Organizations need to conduct continuous monitoring of the security configuration of each system and be able to determine the security posture of systems and the organization at any given time. Organizations also need to demonstrate compliance with various sets of security requirements, such as the Federal Information Security Management Act (FISMA), which is mandated by the U.S. Government for its agencies. All of these tasks are extremely time-consuming and error-prone because there has been no standardized, automated way of performing them. Another problem for organizations is the lack of interoperability across security tools; for example, the use of proprietary names for vulnerabilities or software products creates inconsistencies in reports from multiple tools, which can cause delays in security assessment, decision-making, and vulnerability remediation.

Organizations need standardized, automated approaches to overcoming these challenges, and the Security Content Automation Protocol (SCAP) was developed to help address this. The definition for SCAP (pronounced ess-cap), as expressed in NIST Special Publication (SP) 800-126, is "a suite of specifications that standardize the format and nomenclature by which software flaw and security configuration information is communicated, both to machines and humans." SCAP is designed to organize, express, and measure security-related information in standardized ways, as well as related reference data, such as identifiers for post-compilation software flaws and security configuration issues. SCAP can be used to maintain the security of enterprise systems, such as automatically verifying the installation of patches, checking system security configuration settings, and examining systems for signs of compromise. Individual specifications that comprise SCAP can also be used for other purposes.

This document describes common uses of SCAP and makes recommendations for SCAP users. The document also provides insights to IT product and service vendors about adopting SCAP in their offerings. SCAP does not replace existing security software; rather, support for it can be embedded into existing software.

To take advantage of SCAP's capabilities, Federal agencies and other organizations should follow these recommendations:

**Organizations should use security configuration checklists that are expressed using SCAP to improve and monitor their systems' security.**

A security configuration checklist that is expressed using SCAP, otherwise known as an SCAP-expressed[1] checklist, documents desired security configuration settings, installed patches, and other system security elements using a standardized format. Organizations should identify and obtain SCAP-expressed checklists relevant for their systems' software, then customize the checklists as appropriate to meet specific organizational requirements. After fully testing the checklists, organizations should implement their recommendations. (SCAP does not provide a capability to automatically remediate system security issues found because of using checklists. However, remedies can be applied today using proprietary methods, and NIST plans on providing standardized remediation methods in the future.) Organizations should use SCAP-expressed checklists on an ongoing basis to confirm that systems are configured properly. Federal agencies should use SCAP-expressed checklists such as the United States Government Configuration Baseline (USGCB) checklists to ensure conformance to NIST and OMB

[1] SCAP-expressed content conforms to the requirements specified in NIST SP 800-126 and can be tested for compliance to SP 800-126 using the SCAP Content Validation Tool located at http://scap.nist.gov/revision/1.2/index.html.

security configuration guidance. Also, if available, organizations should use signed content (such as signed SCAP-expressed checklists) to ensure that the content is unaltered.

**Organizations should take advantage of SCAP to demonstrate compliance with high-level security requirements that originate from mandates, standards, and guidelines.**

SCAP-expressed checklists can map individual system security configuration settings to their corresponding high-level security requirements. For example, there are mappings between Windows 7 security configuration settings and the high-level security controls in NIST SP 800-53. These mappings can help demonstrate that the implemented settings adhere to FISMA requirements.[2] The mappings are embedded in SCAP-expressed checklists, which allows SCAP-enabled tools to automatically generate assessment and compliance evidence. This increased automation can significantly reduce the effort needed to achieve assessment results, providing substantial cost savings. To produce FISMA compliance evidence for many NIST SP 800-53 controls, Federal agencies should use SCAP-enabled tools along with SCAP-expressed checklists.

Another area where organizations can benefit from SCAP is continuous monitoring, as discussed in NIST SP 800-137 and NIST Interagency Report (IR) 7756. Continuous monitoring can be performed using SCAP-enabled tools and SCAP-expressed checklists. This allows changes that negatively affect system security to be identified and remediated rapidly, thus minimizing their potential impact. Additionally, deviations from SCAP-expressed checklists may be documented in the form of exceptions, permitting future review of accepted risk by auditors and management.

**Organizations should use standardized SCAP enumerations—identifiers and product names.**

An organization typically uses a collection of tools for security management, such as vulnerability scanners, patch management utilities, and intrusion detection systems. SCAP allows organizations to use standardized enumerations when referring to security-related software flaws, security configuration issues, and platforms. The common understanding achieved through the use of standardized enumerations makes it easier to use security tools, share information, and provide guidance to address security issues. Organizations should encourage security software vendors to incorporate support for Common Vulnerabilities and Exposures (CVE), Common Configuration Enumeration (CCE), and Common Platform Enumeration (CPE) into their products, as well as encourage all software vendors to include CVE and CCE identifiers and CPE product names in their configuration guides, and vulnerability and patch advisories. NIST SP 800-51 Revision 1 contains additional guidelines related to SCAP enumerations.

**Organizations should use SCAP for security measurement and scoring.**

SCAP enables quantitative and repeatable measurement and scoring of software flaw vulnerabilities and software security configuration issues across systems through the combination of the Common Vulnerability Scoring System (CVSS), the Common Configuration Scoring System (CCSS), CVE, CCE, and CPE. The ability to accurately and consistently convey the characteristics of these security problems allows organizations to institute consistent and repeatable mitigation policies throughout the enterprise.

Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws. CVSS scores can be used more easily when organizations use CVE to reference specific vulnerabilities whenever possible. When a new vulnerability is publicly announced, a new CVE identifier is created for it, the affected products are

[2] SCAP can be used to demonstrate compliance with many sets of requirements other than FISMA, such as ISO 27001, DOD 8500, and the Federal Information System Controls Audit Manual (FISCAM).

identified using CPE, and the CVSS base measures and score are computed and added to the National Vulnerability Database (NVD). Organizations can review the CVSS base measures and scores for each new CVE as part of their vulnerability mitigation prioritization processes. SCAP content can be used to check their systems for the presence of the new vulnerability.

Organizations may use CCSS base scores as available, along with information on the organization’s specific policies and environment, to assist in prioritizing the remediation of software security configuration issues based on the relative severity of the issues. CCSS scores can be used more easily when organizations use CCE identifiers to reference specific software security configuration issues whenever possible.

**Organizations should acquire and use SCAP-validated products.**

NIST has established an SCAP product validation program to ensure that SCAP products are thoroughly tested and validated to conform to SCAP requirements. The validation program emphasizes a modular component architecture such that SCAP-validated products are interoperable and interchangeable. The validation program also focuses on correctness testing where appropriate, such as for vulnerability and configuration scanning. Many acquisition officials have embedded requirements for SCAP-validated products in their procurements. For example, OMB requires Federal agencies and agency IT providers to use SCAP-validated products for testing and assessing compliance with the USGCB settings, which are the official configuration settings under the Federal Desktop Core Configuration (FDCC) initiative.

**Software developers and checklist producers should adopt SCAP and use its capabilities.**

Software developers should ensure that their software provides the ability to assess underlying software configuration settings using SCAP, rather than relying on manual checks or proprietary checking mechanisms. Also, product vendors and other checklist developers should create their checklists using SCAP. NIST encourages IT product vendors to participate in SCAP content development because of their depth of knowledge and their ability to speak authoritatively about the most effective and accurate means of assessing their products’ security configurations. Checklist developers are urged to contribute their applicable security configuration checklists to NIST’s National Checklist Program to ensure that the checklists are available to the broadest possible audience.

# 1. Introduction

## 1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), “Securing Agency Information Systems,” as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

## 1.2 Purpose and Scope

The purpose of this document is to provide an overview of the Security Content Automation Protocol (SCAP) version 1.2. This document discusses SCAP at a conceptual level, focusing on how organizations can use SCAP-enabled tools to enhance their security posture. It also explains to IT product and service vendors how they can adopt SCAP version 1.2 capabilities within their offerings.

As new versions of SCAP are released, this document will be updated as needed to reflect any resulting differences in SCAP use and adoption.

Configuration management technologies for non-security purposes, such as functionality and performance, are out of the scope of this document, but not necessarily out of scope for SCAP applicability. This document only addresses security for operational environments for deployed software and does not attempt to address problems with the secure development of software.

## 1.3 Audience

The intended audience for this document is individuals who have responsibilities for maintaining or verifying the security of systems in operational environments. This includes chief information security officers, mid-level management, and technical directors within Federal and state governments and other large organizations; software and hardware vendor product managers; and auditors.

## 1.4 Document Structure

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

- Section 2 explains the motivation behind creating SCAP, defines SCAP, and gives a brief overview of the NIST SCAP product validation and laboratory accreditation programs. This section is intended for all readers.
- Section 3 describes common ways in which SCAP can be used, such as to verify that technical security controls comply with requirements and to communicate information regarding vulnerabilities in a standardized manner. The section also makes recommendations for SCAP users. This section is most relevant to organizations that are interested in adopting and using SCAP.
- Section 4 makes recommendations for how IT product and service vendors can adopt SCAP within their product and service offerings.

The document also contains appendices with supporting material:

- Appendix A defines acronyms and abbreviations for the document.
- Appendix B lists SCAP-related resources.

# 2. SCAP Overview

This section provides an overview of SCAP. First, it explains the initial motivation for creating SCAP. Next, it defines SCAP and provides a high-level overview of its main elements. Finally, it describes the programs that NIST has established for validating SCAP-enabled products and accrediting SCAP product testing laboratories.

## 2.1 The Motivation for Creating SCAP

SCAP was created to provide an automated, standardized approach to maintaining the security of enterprise systems, such as implementing security configuration baselines, verifying the presence of patches and known vulnerabilities, performing continuous monitoring of vulnerabilities and system security configuration settings, examining systems for signs of compromise, and having situational awareness—being able to determine the security posture of systems and an organization at any given time. This is challenging because of the following:

- **The number and variety of systems to secure.** Most organizations have many systems to secure, with numerous applications to be secured for each system. Dozens of operating systems and thousands of applications may be in use across an enterprise, each with its own mechanisms for patching and security configuration management. The same software often needs to be secured somewhat differently on multiple hosts (for example, more stringently on a high-impact system). Also, a single host may have thousands of security configuration settings for its operating system and applications. All of these factors make it more complicated to determine what security changes are needed on each system; to implement those changes quickly, correctly, and consistently; and to verify the security configuration of each system.

- **The need to respond quickly to new threats.** Organizations often need to reconfigure software or install patches to mitigate vulnerabilities that are newly discovered or that are being targeted by attackers. Each year, several thousand new software flaw vulnerabilities are added to the National Vulnerability Database (NVD).[3] Given the number of vulnerabilities and the resources needed to mitigate each one, organizations often have to prioritize the mitigation of the vulnerabilities to ensure that the most important vulnerabilities are addressed more quickly than others.

- **The lack of interoperability.** Many tools for system security, such as patch management and vulnerability management software, use proprietary formats, nomenclatures, measurements, terminology, and content. For example, when vulnerability scanners do not use standardized names for vulnerabilities, it might not be clear to security staff whether multiple scanners are referencing the same vulnerabilities in their reports. This lack of interoperability can cause delays and inconsistencies in security assessment, decision-making, and remediation.

Organizations also need to be able to demonstrate that they have complied with mandates such as the Federal Information Security Management Act (FISMA).[4] To accomplish this, organizations can map the low-level technical details of their system security, such as individual security configuration settings, to high-level security requirements from the mandates. Determining the mappings is time-consuming and is highly susceptible to errors and differences in interpretation. To address this, some common high-level requirements have already been decomposed into lower levels of items. For example, NIST Special Publication (SP) 800-53 decomposes required security controls for FISMA into 18 control families and

[3] The data was taken from NVD's CVE and CCE Statistics Query page (http://web.nvd.nist.gov/view/vuln/statistics).

[4] Examples of other mandates are the Health Information Portability and Accountability Act (HIPAA) and Sarbanes-Oxley (SOX).

over 200 controls.[5] Many of these controls deal with how systems are configured, patched, and securely operated. However, these controls are not at the lowest technical level, so additional mappings are needed to complete the linkage from high-level requirements to individual low-level settings. NIST has created Extensible Configuration Checklist Description Format (XCCDF) checklists containing Common Configuration Enumeration (CCE) identifiers. Data feeds for these identifiers, such as those provided by NVD, can be used to map the CCE identifiers to the high-level security controls provided in NIST SP 800-53.

Organizations need a comprehensive, standardized, automated approach to assessing the security configuration of their operational systems and producing evidence of compliance to high-level requirements. A first step toward establishing this approach was NIST's National Checklist Program,[6] which provides a centralized repository of system security checklists—security recommendations and guidelines that organizations can implement in their operational environments. The initial checklists in the repository were in English prose format, describing actions such as navigating an operating system (OS) or application's menus to view a particular configuration setting value. A prose checklist might be accompanied by a configuration file for implementing the settings or scripts for checking settings. The expectation was that system or security administrators would use the prose documentation, with supporting configuration files or scripts if available, to implement or verify settings and manually document any conflicts or other problems. The concept of a checklist has since expanded to include more fully automated means of implementing security configuration settings, checking patch levels, and installing patches. SCAP was created to support these automation efforts by providing a standardized format for documenting system security settings and configuration mechanisms.

## 2.2 The Definition of SCAP

This document builds conceptually on the technical definition of SCAP, which is maintained in NIST SP 800-126: "The Security Content Automation Protocol (SCAP) is a suite of specifications that standardize the format and nomenclature by which software flaw and security configuration information is communicated, both to machines and to humans."[7] The individual specifications used by SCAP are known as the *SCAP component specifications.* The *SCAP protocol*[8] is an additional, higher-level specification that defines how the component specifications are to be used together in support of SCAP. The security information (both input and output) used by the SCAP protocol is known as *SCAP content*; this includes standardized software flaw, security configuration, and platform identification reference data.

Table 2-1 lists the component specifications for the SCAP version 1.2 protocol. The components are grouped by type:

- **Languages.** The SCAP languages provide standard vocabularies and conventions for expressing security policy, technical check mechanisms, and assessment results.
- **Reporting formats.** The SCAP reporting formats provide the necessary constructs to express collected information in standardized formats.

[5] SP 800-53 Revision 3, *Recommended Security Controls for Federal Information Systems and Organizations* is available at http://csrc.nist.gov/publications/PubsSPs.html#800-53.

[6] This program was originally known as the Security Configuration Checklists Program for IT Products. For more information on the checklists program, see NIST SP 800-70 Revision 2, *National Checklist Program for IT Products,* http://csrc.nist.gov/publications/PubsSPs.html#800-70.

[7] SP 800-126 Revision 2, *The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2,* is available at http://csrc.nist.gov/publications/PubsSPs.html#800-126.

[8] The term "protocol" has different meanings in different contexts. It is meant here as a suite of related specifications for data format and nomenclature, and not meant as in the Internet Engineering Task Force (IETF) definition of the term.

- **Enumerations**. Each SCAP enumeration defines a standard nomenclature (naming format) and an official dictionary or list of items expressed using that nomenclature.
- **Measurement and scoring systems.** In SCAP this refers to evaluating specific characteristics of a security weakness (for example, software vulnerabilities and security configuration issues) and, based on those characteristics, generating a score that reflects their relative severity.
- **Integrity protection.** An SCAP integrity protection specification helps to preserve the integrity of SCAP content and results.

**Table 2-1. SCAP Version 1.2 Component Specifications**

| SCAP Component | Description |
|---|---|
| **Languages** | |
| Extensible Configuration Checklist Description Format (XCCDF) 1.2 | A language for authoring security checklists/benchmarks and for reporting results of evaluating them |
| Open Vulnerability and Assessment Language (OVAL) 5.10 | A language for representing system configuration information, assessing machine state, and reporting assessment results |
| Open Checklist Interactive Language (OCIL) 2.0 | A language for representing assessment content that collects information from people or from existing data stores made by other data collection efforts |
| **Reporting Formats** | |
| Asset Reporting Format (ARF) 1.1 | A format for expressing the exchange of information about assets and the relationships between assets and reports |
| Asset Identification 1.1 | A format for uniquely identifying assets based on known identifiers and/or known information about the assets |
| **Enumerations** | |
| Common Platform Enumeration (CPE) 2.3 | A nomenclature and dictionary of hardware, operating systems, and applications, plus an applicability language for constructing complex logical groupings of CPE names |
| Common Configuration Enumeration (CCE) 5 | A nomenclature and dictionary of software security configurations |
| Common Vulnerabilities and Exposures (CVE) | A nomenclature and dictionary of security-related software flaws |
| **Measurement and Scoring Systems** | |
| Common Vulnerability Scoring System (CVSS) 2.0 | A system for measuring the relative severity of software flaw vulnerabilities |
| Common Configuration Scoring System (CCSS) 1.0 | A system for measuring the relative severity of system security configuration issues |
| **Integrity Protection** | |
| Trust Model for Security Automation Data (TMSAD) 1.0 | A specification for using digital signatures in a common trust model applied to other security automation specifications |

SCAP reference data and other content based on these specifications is available from multiple sources. For example, NVD[9] hosts the official dictionary of CPE entries. Each of the SCAP components offers unique functions and can be used independently, but greater benefits can be achieved by using the components together. For example, the ability to have XCCDF documents that use CCE, CPE, and CVE identifiers with OVAL definitions to express rules and relationships for technical checks and that use OCIL questionnaires to express management and operational checks comprises the building blocks for

[9] NVD is the U.S. government repository of standards-based vulnerability management data (http://nvd.nist.gov/).

*SCAP-expressed* checklists.[10] In other words, SCAP-expressed checklists use a standardized language (XCCDF) to express what checks should be performed (OVAL, OCIL), which platforms are being discussed (CPE), and which security settings (CCE) and software flaw vulnerabilities (CVE) should be addressed.

Use of SCAP-expressed checklists makes it easier for organizations to implement technical security controls on systems, perform ongoing security monitoring, and automate reporting of compliance with high-level security requirements. SCAP-expressed checklists help organizations to quickly and effectively find known security configuration issues and plan their remediation, so as to prevent attackers from compromising systems through known avenues. SCAP-expressed checklists are also used to check systems for signs of compromise, such as the presence of a particular instance of malware. The National Checklist Program (NCP) Repository, located at http://checklists.nist.gov/, hosts a directory of SCAP-expressed checklists and their metadata.

## 2.3 NIST SCAP Product Validation and Laboratory Accreditation Programs

NIST has established both an SCAP product validation program and an SCAP laboratory accreditation program. These programs work together to ensure that SCAP products are thoroughly tested and validated to conform to SCAP requirements. Given SCAP's complexity, this formal testing is needed to ensure that products properly implement SCAP. Organizations should acquire and use SCAP-validated products.[11]

SCAP laboratory accreditation is operated by NIST's National Voluntary Laboratory Accreditation Program (NVLAP). NVLAP accredits independent testing laboratories to perform SCAP product validation testing.[12] Once accredited, laboratories test products using derived test requirements (DTR) as outlined in NISTIR 7511.[13] This report contains a list of specific product requirements, vendor documentation requirements, and testing procedures that the laboratory must perform. After a product is tested, the laboratory sends a detailed test report to NIST's SCAP product validation program with corresponding evidence. Product validation staff reviews the test report, and the program issues a product validation upon successful completion.[14]

A product may be validated as conforming to one or more SCAP capabilities. SCAP capabilities are not product types, but rather ways in which a product may use SCAP. The list of capabilities continues to evolve over time as SCAP evolves. A current list of available SCAP capabilities and their definitions is available within NISTIR 7511 and on the SCAP product validation program website.[15] The term "SCAP product validation program" is used in a general way to include any validation received under the program, even if the validation does not include all of the SCAP components.

The SCAP product validation program will ensure that a product conforms to one or more SCAP capabilities. NIST recommends that organizations acquire the most recent version of SCAP-validated products to receive the greatest SCAP functionality and the most capable version of the vendor's product. Each new version of SCAP automates more functionality and provides other benefits; for example, SCAP

[10] SCAP-expressed checklists are further defined in Table 4-1 of NIST SP 800-70 Revision 1.

[11] See OMB Memorandum 08-22, "Guidance on the Federal Desktop Core Configuration (FDCC)" (http://www.whitehouse.gov/omb/memoranda/fy2008/m08-22.pdf).

[12] A list of laboratories is at http://nvd.nist.gov/scapproducts.cfm. General accreditation requirements for laboratories are defined in NIST Handbook 150 (http://ts.nist.gov/Standards/Accreditation/upload/nist-handbook-150.pdf), and SCAP-specific requirements are found in NIST Handbook 150-17 (http://ts.nist.gov/Standards/Accreditation/handbook.cfm).

[13] NIST Interagency Report (NISTIR) 7511, *Security Content Automation Protocol (SCAP) Version 1.2 Validation Program Test Requirements*, http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7511

[14] A list of currently validated products is available at http://nvd.nist.gov/scapproducts.cfm.

[15] http://scap.nist.gov/validation/

version 1.2 added more checks and supported enhanced reporting. Also, the rigor of product testing increases with each new version of the product validation program.

# 3. Recommendations for Common Uses of SCAP

This section describes several common uses of SCAP version 1.2 and makes related recommendations to SCAP users. In addition to following these specific recommendations, organizations should also talk with their relevant IT product and service vendors about their support of SCAP, the security configurations of their products, and the need for standardized, automation-supporting security content.

## 3.1 Security Checklist Verification

Many organizations produce security guidance for a wide range of platforms. It is important that the guidance be both human and machine-readable to allow automated verification of security guidance. SCAP enables this via the creation of SCAP-expressed security checklists that can be processed by SCAP-validated content consumers.[16] The settings in such a checklist can be compared to a system's actual configuration to confirm compliance with the checklist and identify any deviations from the organization's requirements. The core of each checklist is expressed using XCCDF, with OVAL and/or OCIL used to check security configuration settings, patch levels, and other security characteristics. OVAL is used for automated checks; OCIL is used for checks that cannot be performed satisfactorily using OVAL. XCCDF, OVAL, and OCIL use CVE identifiers, CCE identifiers, and CPE product names, as described in Section 3.4. A checklist can contain multiple profiles for securing a product deployed in multiple environments, each with unique security requirements.[17]

Both comprehensive SCAP checklists, such as a checklist to secure an operating system, and more specialized SCAP checklists are valuable. A specialized checklist can be used to check particular characteristics of systems to identify potential security problems. A common example is using SCAP content to confirm the installation of patches and identify which patches are missing. SCAP-formatted data for patch checking can be made publicly available by software vendors for their products; organizations can download this data and use it through their SCAP-capable tools.[18]SCAP-expressed security checklists are helpful in other ways. They can be used to improve the testing of new software. For example, an organization may be planning on installing a new application on systems that have been secured using a set of SCAP-expressed checklists. As part of testing the new application, the organization can use the checklists to ensure that the new application does not alter existing checklist settings and that the new application functions properly with the checklist settings in place. SCAP-expressed checklists are also helpful for security assessments because they provide an unambiguous way of communicating what and how individual security settings and software flaws will be checked. It also allows SCAP content and corresponding SCAP results to be readily used as evidence that certain security configurations and software flaws exist or do not exist in an enterprise. Through the SCAP content, assessors can understand the rationale for security configuration. SCAP-expressed checklists also give security operation teams a way to communicate security configurations to other teams so that they can integrate the configuration into standardized builds and images.

To help automate security checklist verification, organizations should identify and obtain SCAP-expressed security checklists relevant for their systems' operating systems and applications. In some cases, a security configuration is mandated in policy (for example, the United States Government Configuration Baseline [USGCB] configuration settings mandated for Federal agency Windows hosts), which supersedes the authority of all other configurations. In all other cases, selecting a checklist from the National Checklist Program (NCP) is highly recommended. Due to February 2008 modifications to

[16] Human-readable guidance can be generated from XCCDF using automated tools.

[17] See NIST SP 800-70, *National Checklist Program for IT Products—Guidelines for Checklist Users and Developers* for additional information on environments. http://csrc.nist.gov/publications/PubsSPs.html#800-70

[18] Patch information can be downloaded from the MITRE OVAL Repository at http://oval.mitre.org/repository/.

Federal Acquisition Regulation (FAR) Part 39, Federal agencies must procure IT products with relevant NCP checklists applied.[19] NCP checklists are publicly vetted, and many offer manufacturer-endorsed methods of configuring and evaluating products. Federal agencies should use SCAP-expressed checklists in conjunction with SCAP-validated tools to ensure conformance to NIST and OMB security configuration guidance.[20] Also, if available, organizations should use digitally signed SCAP-expressed checklists[21] to ensure that the content is unaltered.

After acquiring checklists, organizations should customize them as appropriate to tailor them to specific organizational and operational requirements. For example, an organization might choose to omit a check for a particular security setting because the organization uses a compensating control instead of that setting. SCAP-expressed checklists are documented in a standardized XML format, so that they can be customized easily, allowing organizations to add, modify, and delete checks. SCAP-validated tools can process the customized checklists without modification to the tools. After performing any necessary customization and fully testing the checklists, organizations should implement the checklists' recommendations across all possible systems; see Section 3.6 for additional information. This can be done before systems are deployed to ensure that they have been secured as intended. Once systems have been securely configured, organizations should continuously monitor their security, as described in Section 3.5, to ensure that the checklist settings are maintained.

## 3.2 Artifact Identification

Software and hardware present on a system leave behind artifacts that can be detected by security automation technologies. Examples of artifacts are possible executables, system libraries, drivers, and configuration settings. Even if the software or hardware has been uninstalled from the system, artifacts might still remain that allow its previous presence to be established. Short SCAP-expressed checklists can specify logical combinations of artifacts to be identified by OVAL definitions. OVAL definitions might check for artifacts by calculating file checksums, checking for the existence of a particular service, verifying the value of a particular configuration setting, etc. OCIL questionnaires could also be used to manually verify artifacts that cannot be detected using OVAL.

The ability to find artifacts and link them to software or hardware can be valuable for several reasons. One is for inventory purposes, such as identifying which software is currently installed on a system and which software was previously installed on the system. Artifact identification can also be performed as part of whitelist or blacklist verification—for example, ensuring that prohibited software is not and has not been installed on a system. Another possible use is malware detection—not taking the place of antivirus software, but rather looking for instances of particular malware known or suspected to be in the environment. If the method for finding evidence of a particular attack can be determined, then the check can be expressed in an SCAP format. Software vendors, incident response teams, and other organizations can rapidly make the check information publicly available. As soon as this information is available, an organization can use it with all of their SCAP-validated tools, instead of having to wait for each tool vendor to perform the necessary research and develop, test, and distribute the check information. The ability to use the same SCAP content with many tools permits an organization to conduct the checks and identify problems much more quickly, thus reducing the window of opportunity for successful attacks.

[19] Paragraph (d) of section 39.101 states, "In acquiring information technology, agencies shall include the appropriate IT security policies and requirements, including use of common security configurations available from the NIST's website at http://checklists.nist.gov/. Agency contracting officers should consult with the requiring official to ensure the appropriate standards are incorporated." http://www.acquisition.gov/far/current/html/FARTOCP39.html

[20] The DoD also publishes SCAP content, and DoD profiles are often available within NIST SCAP content.

[21] For more information on digitally signed SCAP-expressed checklists, see NIST SP 800-126, http://csrc.nist.gov/publications/PubsSPs.html#800-126 and NIST IR 7802, http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7802.

## 3.3 Requirements Evidence

There are many high-level sets of requirements for security, ranging from Congressional and executive mandates to standards and guidelines from industry, federal agencies, integrators, academia, and vendors. Examples of such requirements include FISMA, COBIT, Sarbanes-Oxley (SOX), Health Insurance Portability and Accountability Act (HIPAA), ISO 27001, DOD 8500, and the Federal Information System Controls Audit Manual (FISCAM). Organizations often find it challenging and costly to demonstrate that they have implemented their security controls in accordance with these requirements: to link evidence measuring the implementation and effectiveness of controls to the controls themselves to help support risk measurement. To assist with this, SCAP content can characterize these links for one or more high-level sets of requirements.

For example, NIST has created SCAP links between low-level Windows 7 settings (which SCAP expresses using CCE identifiers) and the high-level security controls in NIST SP 800-53, which are the security requirements for FISMA. An SCAP-expressed checklist can link a requirement for authentication management in NIST SP 800-53 to a specified need to check that the system’s minimum password length is at least eight characters, as well as defining how that check should be conducted on a particular platform. The XCCDF checklists have embedded links that can be used to automatically generate NIST SP 800-53 assessment and compliance evidence and potentially evidence for other high-level policies as well. NIST hosts several data streams of links between NIST SP 800-53 security controls and SCAP CCE identifiers at http://scap.nist.gov/content. Product vendors and agencies are encouraged to use these mappings when demonstrating relationships between NIST SP 800-53 security controls and configuration settings expressed using CCE.

Whenever feasible, Federal agencies should automate their FISMA technical security control compliance activities using SCAP, because SCAP enables security operations staff to run low-level configuration and vulnerability scans with output that can be used for evidence of compliance with FISMA. Thus, the agency’s security operations team can generate this evidence while performing their normal job of scanning and securing the agency’s systems.

## 3.4 Standardized Enumerations

As discussed in Section 2.2, SCAP supports a variety of enumerations. An SCAP enumeration includes a set of identifiers. Each identifier is a unique reference to a logical entity such as a system software flaw, security configuration issue, or product. Within SCAP, an enumerated value is often expressed as a pairing of an identifier and a value, the value being specific to the type of enumeration.

An organization typically uses a collection of tools for security management, such as vulnerability scanners, patch management utilities, and intrusion detection systems. Historically, these tools have used proprietary data formats, nomenclature, and interfaces, which prevents interoperability and creates disjointed data stores that require manual intervention or customized application development to facilitate data exchange. The SCAP protocol and reference data allow organizations to use standardized enumerations—specifically, CVE identifiers, CCE identifiers, and CPE product names—when referring to security-related software flaws and configuration issues and to platforms. The common understanding achieved through the use of standardized enumerations makes it easier to use security tools, share information, and provide guidance to address security problems. For example, it simplifies reporting on the results of internal security scans and correlating between scans of different tools—an organization using multiple SCAP-validated vulnerability scanners can readily consolidate their outputs into a single database or report. Use of CVE and CCE identifiers and CPE product names also helps minimize confusion regarding which problem is being referenced, and enables organizations to quickly identify additional information about the problem (e.g., remediation advice).

Organizations should encourage security software vendors to incorporate support for CVE, CCE, and CPE into their products, as well as encourage all software vendors to include CVE and CCE identifiers and CPE product names in their product security advisories and other security-related documentation and communications. This is particularly helpful for improving communications between vendors and users. The use of standardized identifiers and product names is also helpful for incident response, enabling faster decision making and ensuring consistency for incident reporting throughout an organization and between an organization and external entities such as US-CERT and law enforcement agencies. Organizations should report incident details using these standardized enumerations where possible. This ensures that all vulnerability communications precisely identify relevant vulnerabilities and affected products, enable correlation and integration of reports, and enable correlation with supplemental information residing in other data repositories.

## 3.5 Continuous Monitoring

Continuous monitoring can be performed using SCAP-validated authenticated configuration scanners and SCAP-expressed checklists. This allows changes that negatively affect system security to be identified and remediated rapidly, thus minimizing their potential impact. Additionally, deviations from SCAP-expressed checklists may be documented in the form of exceptions, permitting future review of accepted risk by auditors and management. It is rarely sufficient to configure a computer once and assume that the settings continue to provide appropriate security—they may change as software is installed, upgraded, and patched, or as computers are connected and disconnected from domains, for example. Local administrators and users who maintain the computers may also alter security, such as a user who feels that a certain security feature—such as a locking screen saver—is inconvenient and turns it off.

Draft NIST Interagency Report 7756, *CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Architecture*[22], provides more information on continuous monitoring and how it relates to SCAP. For continuous monitoring to be accomplished, security sensors and controllers will need to be modestly instrumented to support the continuous monitoring architecture and related security automation standards. Much of this has already been accomplished through SCAP and the NIST SCAP Validation Program. In addition, new functionality in data aggregation, analysis, and event management products is needed. Fortunately, many existing products contain much of the needed functionality and should require only modest instrumentation to support the continuous monitoring architecture.

Leveraging the automation and standardization of SCAP enables organizations to move beyond compliance verification and vulnerability assessment into the realm of security data analytics—the use of analysis methods to identify, express, measure, detect, and report on patterns and elements of interest in the data collected. The technical standards, framework, and capabilities required to support the four categories of common uses discussed earlier in this section can be used as the foundational elements to securely operate and manage networked systems. Use of SCAP enables security analysts to detect artifacts that indicate the presence of, or potential for, complex combinations of vulnerabilities that might be used for exploitation. For example, automated tools can currently use OVAL checks to identify both the existence of a given vulnerability (e.g., CVE-2008-4250) and signs of infection (e.g., existence of a known-malicious Windows registry key, HKLM\SYSTEM\CurrentControlSet\Services\vcdrlxeu, installed by variants of the Conficker worm) to locate potentially compromised information systems.

The ability to perform security configuration verification and requirements traceability implies that an organization is able to identify the devices on its networks, has defined permissible states for these devices, and can compare the operational state to the permissible condition. Using consistent security

[22] http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7756

enumerations enables this information to be consistently associated and integrated with asset management, network management, provenance, and current state information. The improved awareness of the system conditions supports explicit and accurate statements of information system health, trust, and risk. Consistent vulnerability measurement enables reliable descriptions of the characteristics and attributes of specific vulnerabilities. It supports consistent association of vulnerabilities with threat and impact information, enabling the prioritization of network management activities based on risk rather than simply the existence of vulnerabilities. Using SCAP-derived and enumerated data also helps to identify root-cause and systemic issues associated with current and near-term risks, enabling organizations to measure and address entire classes of risk from their systems in a more cost-effective manner.

The use of SCAP capabilities and content enables security analytics to be developed and implemented at various organizational levels, enabling consistent data collection, aggregation, and summarization in support of security-related risk measurement and decision-making within the organization.[23]

## 3.6 Remediation

SCAP version 1.2 does not provide a capability to automatically implement remediations, such as applying patches or altering noncompliant security configuration settings. However, this does not mean that SCAP does not provide value for remediation. In fact, the results of other uses of SCAP identified in this section, such as Requirements Verification, Artifact Identification, Continuous Monitoring, and Security Measurement, all provide strong information regarding what needs to be remediated. Because this information is reported using standardized formats and enumerations, it can easily be used by non-SCAP remediation tools as input to identify what the security problems are that need to be remediated.

## 3.7 Security Measurement

SCAP enables quantitative and repeatable measurement and scoring of software flaw vulnerabilities and software security configuration issues across systems through the combination of CVSS, CCSS, CVE, CCE, and CPE. The ability to accurately and consistently convey the characteristics of these security problems allows organizations to institute consistent and repeatable mitigation policies throughout the enterprise. For example, an organization could establish a policy that specifies how quickly software flaw vulnerabilities must be mitigated based in part on their measures or scores, such as patching the most severe within a certain amount of time after patches become available.[24] Organizations could also have separate requirements for different types of software to ensure that a problem in a critical application is remediated more quickly than a similar problem in a non-critical application. Another helpful feature is that the major properties of each security problem are documented as part of generating each CVSS or CCSS score. This allows users to understand the basis for each score and to take these properties into account when planning mitigation strategies, as part of overall risk assessment and decision making. Security measurement is an increasingly important part of risk management and the Risk Management Framework.[25]

Organizations should use CVSS base scores to assist in prioritizing the remediation of known security-related software flaws based on the relative severity of the flaws. Organizations may also find it beneficial

[23] Section 3.5 is derived primarily from material by Kimberly Watson of the National Security Agency that is available at http://scap.nist.gov/events/2009/itsac/presentations/day2/Day2_SCAP_Watson_Text.pdf.

[24] For example, the Payment Card Industry has mandated the use of CVSS scores when evaluating which software flaw vulnerabilities on a payment card server must be remediated. For more information, see https://www.pcisecuritystandards.org/pdfs/pci_dss_technical_and_operational_requirements_for_approved_scanning_vendors_ASVs_v1-1.pdf.

[25] For more information on the Risk Management Framework, see NIST SP 800-37 Revision 1, *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* at http://csrc.nist.gov/publications/PubsSPs.html#800-37.

to customize CVSS scores for their specific environments as resources and tools permit. CVSS scores can be used more easily when organizations use CVE to reference specific vulnerabilities whenever possible. When a new vulnerability is publicly announced, a new CVE identifier is created for it, the affected products are identified using CPE, and the CVSS base measures and score are computed and added to NVD. Organizations can review the CVSS base measures and scores for each new CVE as part of their vulnerability mitigation prioritization processes. SCAP content can be used to check their systems for the presence of the new vulnerability. This entire process helps an organization to achieve better situational awareness of its overall security posture.

Organizations may use CCSS base scores as available, along with information on the organization's specific policies and environment, to assist in prioritizing the remediation of software security configuration issues based on the relative severity of the issues. CCSS scores can be used more easily when organizations use CCE identifiers to reference specific software security configuration issues whenever possible. CCSS scores can be a valuable component of quantitative risk measurement.

# 4. Recommendations for Vendor and Service Adoption of SCAP

This section makes recommendations for how various groups—software developers and SCAP content producers—may adopt SCAP and take advantage of its capabilities. Enhancing a single product, service, or process so that it supports SCAP is valuable, but greater benefits are achieved by using SCAP across different products, services, and processes to improve interoperability. This increases the efficiency of security management and improves the security of systems.

In addition to the recommendations presented below for specific groups, NIST also encourages community involvement in how SCAP and individual SCAP components evolve and are applied. NIST invites interested parties to participate in the SCAP and SCAP components' mailing lists to be aware of ongoing development and voice opinions.

## 4.1 Software Developers

The following recommendations are for organizations and individuals who develop software, particularly operating systems and applications:

- **Register and use standardized identifiers.** Software developers are encouraged to request unique CPE identifiers for their products. A CPE identifier has much broader usage than just security—it can also be used as a unique identifier for compliance, configuration, change, and asset management purposes. Once identifiers have been established, software developers should incorporate them in their security advisories and other security-related documentation and communications. The same is true for CCE identifiers (for software security configuration issues) and CVE identifiers (for software security flaws); see NIST SP 800-51 Revision 1 for additional information.

- **Make security settings available through automation.** Software developers should ensure that their software's underlying configuration settings can be checked automatically through APIs, rather than relying primarily on GUI-based instructions for people to manually check configuration settings. In some cases, checks in SCAP-expressed checklists must be left as manual checks or expressed in OCIL questionnaires because there is not a reliable automated checking method available.

- **Develop security software with SCAP validation requirements in mind.** Before beginning development of SCAP-enabled security software, tool developers are encouraged to familiarize themselves with the SCAP product validation program test requirements.

See Section 4 of NIST SP 800-51 Revision 1, *Guide to Using Vulnerability Naming Schemes* for additional recommendations.[26]

## 4.2 SCAP Content Producers

The following recommendations are applicable to all organizations and individuals who develop SCAP content, including software developers. NIST particularly encourages product vendors to create SCAP content for their own products.

- **Develop security checklists in SCAP format.** Security checklists usually involve verification of a product's security configuration settings, checks for a product's known software flaws, and other product-specific elements to be evaluated. Checklist developers should create SCAP-expressed checklists to support automated configuration management, requirements traceability, and interoperability. NIST particularly encourages IT product vendors to participate in SCAP content

[26] http://csrc.nist.gov/publications/PubsSPs.html#800-51

development because of their depth of knowledge and their ability to speak authoritatively about the most effective and accurate means of assessing their products' security configurations. While SCAP content can comprise a subset of the SCAP components, NIST encourages vendors to use all applicable SCAP components for improved effectiveness and interoperability.

- **Use the SCAP Content Validation Tool.** NIST offers a free SCAP Content Validation Tool[27] that can validate SCAP content to ensure that it satisfies the requirements of NIST SP 800-126. This includes ensuring that content is well-formed, cross-references are valid, and required values are appropriately set. Checklist developers are strongly encouraged to use the SCAP Content Validation Tool on their checklists.

- **Contribute checklists to the National Checklist Program.** Checklist developers are strongly urged to contribute their applicable security configuration checklists to the National Checklist Program. This ensures that the SCAP content is available to the broadest possible audience. The NCP accepts submissions of SCAP-expressed checklist content and makes them available via the NVD web site.

- **Participate in developing OVAL.** The OVAL specification cannot provide additional types of checks without subject matter experts providing input regarding APIs for specific products. For example, if an application or operating system exposes configuration data via an API function call, then the subject matter expert can inform the custodian of OVAL[28] to help expand OVAL's applicability. Subject matter experts from software vendors are particularly encouraged to provide suggestions related to OVAL checks and to contribute OVAL code associated with their products to the NCP. This will assure that content consumers have access to vendors' specific guidance for assessing the security of their installed software.

[27] http://scap.nist.gov/revision/1.2/index.html

[28] As of this writing, the custodian for OVAL is the MITRE Corporation.

## Appendix A—Acronyms and Abbreviations

Selected acronyms and abbreviations used in the publication are defined below.

| | |
|---|---|
| **API** | Application Programming Interface |
| **ARF** | Asset Reporting Format |
| **CCE** | Common Configuration Enumeration |
| **CCSS** | Common Configuration Scoring System |
| **CPE** | Common Platform Enumeration |
| **CVE** | Common Vulnerabilities and Exposures |
| **CVSS** | Common Vulnerability Scoring System |
| **DoD** | Department of Defense |
| **FAR** | Federal Acquisition Regulation |
| **FDCC** | Federal Desktop Core Configuration |
| **FIPS** | Federal Information Processing Standard |
| **FIRST** | Forum of Incident Response and Security Teams |
| **FISCAM** | Federal Information System Controls Audit Manual |
| **FISMA** | Federal Information Security Management Act |
| **GUI** | Graphical User Interface |
| **HIPAA** | Health Information Portability and Accountability Act |
| **ISO** | International Organization for Standardization |
| **IT** | Information Technology |
| **ITL** | Information Technology Laboratory |
| **NCP** | National Checklist Program |
| **NIST** | National Institute of Standards and Technology |
| **NISTIR** | National Institute of Standards and Technology Interagency Report |
| **NSA** | National Security Agency |
| **NVD** | National Vulnerability Database |
| **NVLAP** | National Voluntary Laboratory Accreditation Program |
| **OCIL** | Open Checklist Interactive Language |
| **OMB** | Office of Management and Budget |
| **OS** | Operating System |
| **OVAL** | Open Vulnerability and Assessment Language |
| **SCAP** | Security Content Automation Protocol |
| **SOX** | Sarbanes-Oxley |
| **SP** | Special Publication |
| **TMSAD** | Trust Model for Security Automation Data |
| **URL** | Uniform Resource Locator |
| **US-CERT** | United States Computer Emergency Readiness Team |
| **USGCB** | United States Government Configuration Baseline |
| **XCCDF** | Extensible Configuration Checklist Description Format |
| **XML** | Extensible Markup Language |

# Appendix B—SCAP Resources

This appendix lists selected SCAP-related resources.

**Table B-1. Websites for SCAP Component Specifications**

| Resource | URL |
|---|---|
| Asset Identification | http://scap.nist.gov/revision/1.2/#ai |
| Asset Reporting Format (ARF) | http://scap.nist.gov/revision/1.2/#arf |
| Common Configuration Enumeration (CCE) | http://scap.nist.gov/revision/1.2/#cce<br>http://cce.mitre.org/ |
| Common Configuration Scoring System (CCSS) | http://scap.nist.gov/revision/1.2/#ccss |
| Common Platform Enumeration (CPE) | http://scap.nist.gov/revision/1.2/#cpe<br>http://cpe.mitre.org/ |
| Common Vulnerabilities and Exposures (CVE) | http://scap.nist.gov/revision/1.2/#cve<br>http://cve.mitre.org/ |
| Common Vulnerability Scoring System (CVSS) | http://scap.nist.gov/revision/1.2/#cvss<br>http://www.first.org/cvss/ |
| Extensible Configuration Checklist Description Format (XCCDF) | http://scap.nist.gov/revision/1.2/#xccdf |
| Open Checklist Interactive Language (OCIL) | http://scap.nist.gov/revision/1.2/#ocil |
| Open Vulnerability and Assessment Language (OVAL) | http://scap.nist.gov/revision/1.2/#oval<br>http://oval.mitre.org/ |
| Trust Model for Security Automation Data (TMSAD) | http://scap.nist.gov/revision/1.2/#tmsad |

**Table B-2. NIST SCAP-Related Publications**

| Publication | URL |
|---|---|
| SP 800-51 Revision 1, *Guide to Using Vulnerability Naming Schemes* | http://csrc.nist.gov/publications/PubsSPs.html#800-51 |
| SP 800-70 Revision 2, *National Checklist Program for IT Products—Guidelines for Checklist Users and Developers* | http://csrc.nist.gov/publications/PubsSPs.html#800-70 |
| SP 800-126 Revision 2, *The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2* | http://csrc.nist.gov/publications/PubsSPs.html#800-126 |
| IR 7275 Revision 4, *Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7275-r4 |
| IR 7435, *The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agencies* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7435 |
| IR 7502, *The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7502 |
| IR 7511 Revision 2, *Security Content Automation Protocol (SCAP) Version 1.2 Validation Program Test Requirements (Draft)* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7511 |

| Publication | URL |
|---|---|
| IR 7693, *Specification for Asset Identification 1.1* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7693 |
| IR 7694, *Specification for the Asset Reporting Format 1.1* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7694 |
| IR 7695, *Common Platform Enumeration: Naming Specification Version 2.3* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7695 |
| IR 7696, *Common Platform Enumeration: Name Matching Specification Version 2.3* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7696 |
| IR 7697, *Common Platform Enumeration: Dictionary Specification Version 2.3* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7697 |
| IR 7698, *Common Platform Enumeration: Applicability Language Specification Version 2.3* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7698 |
| IR 7756, *CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Architecture (Draft)* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7756 |
| IR 7802, *Trust Model for Security Automation Data (TMSAD) Version 1.0* | http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7802 |

**Table B-3. Other SCAP Resources**

| Resource | URL |
|---|---|
| Federal Desktop Core Configuration (FCCC) | http://fdcc.nist.gov/ |
| List of SCAP Validated Products | http://nvd.nist.gov/scapproducts.cfm |
| National Checklist Program (contains U.S. government SCAP checklists) | http://web.nvd.nist.gov/view/ncp/repository |
| National Voluntary Laboratory Accreditation Program (NVLAP) | http://www.nist.gov/pml/nvlap/ |
| National Vulnerability Database (NVD) | http://nvd.nist.gov/ |
| NIST Handbook 150 (general laboratory accreditation requirements) | http://ts.nist.gov/Standards/Accreditation/upload/nist-handbook-150.pdf |
| NIST Handbook 150-17 (includes specific SCAP laboratory accreditation requirements) | http://ts.nist.gov/Standards/Accreditation/handbook.cfm |
| Official Common Platform Enumeration (CPE) Dictionary | http://nvd.nist.gov/cpe.cfm |
| SCAP Validation Program | http://scap.nist.gov/validation/ |
| Security Content Automation Protocol (SCAP) | http://scap.nist.gov/ |
| United States Government Configuration Baseline (USGCB) | http://usgcb.nist.gov/ |