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

Special Publication 800-117
(Draft)

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

Recommendations of the National Institute of Standards and Technology

Matthew Barrett
Chris Johnson
Peter Mell
Stephen Quinn
Karen Scarfone

NIST Special Publication 800-117 (Draft)

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

Recommendations of the National Institute of Standards and Technology

Matthew Barrett
Chris Johnson
Peter Mell
Stephen Quinn
Karen Scarfone

C O M P U T E R S E C U R I T Y

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

May 2009

U.S. Department of Commerce

Gary Locke, Secretary

National Institute of Standards and Technology

Dr. Patrick D. Gallagher, Deputy 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 (Draft)
Natl. Inst. Stand. Technol. Spec. Publ. 800-117, 25 pages (May 2009)

Acknowledgments

The authors, Matthew Barrett, Chris Johnson, Peter Mell, Stephen Quinn, and Karen Scarfone of the National Institute of Standards and Technology (NIST), wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors would like to acknowledge Tim Grance of NIST, Kevin Sitto and Brad Wood of G2, Lieutenant Colonel Joe Wolfkiel and Dan Schmidt of Department of Defense, Scott Armstrong and Andy Bove of Secure Elements, and Rob Hollis of Threat Guard for their keen and insightful assistance throughout the development of the document. Additional acknowledgments will be added to the final version of the publication.

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. This is extremely time-consuming and error-prone because there has been no standardized, automated way of securing software. Organizations also need to periodically verify the security of each system, which is also much more difficult to do without standardized, automated checking tools. Further complicating system security management is the need to respond appropriately to new vulnerabilities and threats, prioritizing them so the most significant ones can be addressed sooner. Another problem is the lack of interoperability across system security tools; for example, the use of proprietary names for vulnerabilities or platforms creates inconsistencies in reports from multiple tools, which can cause delays in security assessment, decision-making, and vulnerability remediation. Organizations also need to demonstrate compliance with security requirements in mandates such as the Federal Information Security Management Act (FISMA).

Organizations need a comprehensive, standardized approach to overcoming these challenges, and the Security Content Automation Protocol (SCAP) has been developed to help provide such an approach. SCAP comprises a suite of specifications for organizing and expressing security-related information in standardized ways, as well as related reference data, such as identifiers for software flaws and security configuration issues. SCAP can be used for maintaining 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. This document defines SCAP and the component specifications that comprise it. It 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, 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 checklist, documents desired security configuration settings, installed patches, and other system security elements in 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. Customization is generally easy to do. After fully testing the checklists, organizations should implement their recommendations. (The current version of SCAP does not provide a capability to automatically implement checklists. However, SCAP-expressed checklists can be applied today using proprietary methods, and NIST plans on enhancing SCAP to provide standardized implementation methods.) Organizations should use SCAP-expressed checklists on an ongoing basis to confirm that systems are configured properly. Federal agencies should use SCAP-expressed checklists to ensure conformance to NIST and OMB security configuration guidance.

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, NIST has created mappings between Windows XP security configuration settings and the high-level security controls in NIST Special Publication (SP) 800-53, which supports FISMA. These mappings can help demonstrate that the

implemented settings adhere to FISMA requirements. The mappings are embedded in SCAP-expressed checklists, which allows SCAP-enabled tools to automatically generate assessment and compliance evidence. This can provide a substantial savings in effort and cost. To produce FISMA compliance evidence for many NIST SP 800-53 controls, Federal agencies should use SCAP-enabled tools along with SCAP-expressed checklists.

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 issue 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 vulnerability and patch advisories.

Organizations should use SCAP for vulnerability measurement and scoring.

SCAP enables quantitative and repeatable measurement and scoring of software flaw vulnerabilities across systems through the combination of the Common Vulnerability Scoring System (CVSS), CVE, and CPE. The ability to accurately and consistently convey the characteristics of a vulnerability 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 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 should acquire and use SCAP-validated products.

NIST has established a SCAP product validation program to ensure that SCAP products are thoroughly tested and validated to conform to SCAP requirements. Acquisition officials have already embedded requirements for SCAP-validated products in their procurements. For example, OMB requires Federal agencies and agency IT providers to use SCAP-validated Federal Desktop Core Configuration (FDCC) Scanners for testing and assessing FDCC compliance.[1]

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

Whenever feasible, 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

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

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). 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's capabilities within their offerings.

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. The security of software code is outside the scope of this document.

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 mid-level management, chief information security officers, 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 several appendices with supporting material:

- Appendix A provides details on how SCAP can be used in support of FISMA compliance efforts. This appendix is technical in nature.
- Appendix B defines acronyms and abbreviations for the document.
- Appendix C 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 a standardized approach to maintaining the security of enterprise systems, such as automatically verifying the presence of patches, checking system security configuration settings, and examining systems for signs of compromise. 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. 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 newly being targeted by attackers. In 2008, more than 5,600 software flaw vulnerabilities were added to the National Vulnerability Database (NVD).[2] 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).[3] 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 17 security control families and 171 controls.[4] Many of these controls deal with how systems are configured, patched, and

---

2 The data was taken from NVD's CVE and CCE Statistics Query page (http://nvd.nist.gov/statistics.cfm).

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

4 SP 800-53 Revision 2, *Recommended Security Controls for Federal Information Systems*, and a draft of Revision 3 are both available at http://csrc.nist.gov/publications/PubsSPs.html.

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.

Organizations need a comprehensive, standardized 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,[5] 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 issues. 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

SCAP has two major elements. First, it is a protocol—a suite of six open specifications that standardize the format and nomenclature by which security software communicates information about software flaws and security configurations.[6] Each specification is also known as an *SCAP component*. Second, SCAP includes software flaw and security configuration standardized reference data, also known as *SCAP content*. SCAP has several uses, including automating checks for known vulnerabilities, automating the verification of security configuration settings, and generating reports that link low-level settings to high-level requirements.

Table 2-1 lists the six components of the SCAP protocol. SCAP version 1.0 is comprised of particular versions of these components, as listed in draft NIST Interagency Report (NISTIR) 7511, *Security Content Automation Protocol (SCAP) Version 1.0 Validation Program Test Requirements*.[7] The components are grouped by type: enumerations, vulnerability measurement and scoring, and expression and checking languages. The *enumerations* group has nomenclatures and dictionaries for security and product-related information. The *vulnerability measurement and scoring* group has specifications for measuring the characteristics of vulnerabilities and generating scores based on those characteristics. The *expression and checking languages* group has Extensible Markup Language (XML) schemas for specifying checklists, generating checklist reports, and specifying the low-level testing procedures used by the checklists.

---

5 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 1, *National Checklist Program for IT Products*, http://csrc.nist.gov/publications/PubsSPs.html.

6 The term "SCAP" technically refers to the entire SCAP protocol; however, many people also use it to refer to just one SCAP specification or a combination of specifications. The context surrounding the use of the term should indicate which meaning is intended.

7 Section 2.2 of NISTIR 7511 provides additional information on the six specifications. http://csrc.nist.gov/publications/PubsNISTIRs.html

Table 2-1. SCAP Version 1.0 Components

SCAP Component	Description	Maintaining Organization
Enumerations		
Common Configuration Enumeration (CCE)	Nomenclature and dictionary of system security issues	MITRE Corporation
Common Platform Enumeration (CPE)	Nomenclature and dictionary of product names and versions	MITRE Corporation
Common Vulnerabilities and Exposures (CVE)	Nomenclature and dictionary of security-related software flaws	MITRE Corporation
Vulnerability Measurement and Scoring		
Common Vulnerability Scoring System (CVSS)	Specification for measuring the relative severity of software flaw vulnerabilities	Forum of Incident Response and Security Teams (FIRST)
Expression and Checking Languages		
Extensible Configuration Checklist Description Format (XCCDF)	Language for specifying checklists and reporting checklist results	National Security Agency (NSA) and NIST
Open Vulnerability and Assessment Language (OVAL)	Language for specifying low-level testing procedures used by checklists	MITRE Corporation

SCAP content—reference data—is available from multiple sources. For example, NVD[8] hosts a dictionary of CPE entries and information on CVE entries, while the MITRE Corporation hosts an OVAL database and maintains a list of CCE entries.[9] Each of the six SCAP components offers a unique function and is used independently, but greater benefits can be achieved by using the components together. For example, the ability to express CCEs according to CPEs in an XCCDF format comprises the building blocks for *SCAP-expressed* checklists.[10] In other words, SCAP-expressed checklists use a standardized language (XCCDF) to express what platform is being discussed (CPE) and what security settings (CCE) 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 and remediate known security configuration issues, which prevents 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) web site, located at http://checklists.nist.gov/, is the repository for SCAP-expressed checklists.

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

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

9 http://cce.mitre.org/lists/cce_list_references.html

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

11 OMB Memorandum 08-22, "Guidance on the Federal Desktop Core Configuration (FDCC)" (http://www.whitehouse.gov/omb/memoranda/fy2008/m08-22.pdf) mandates the use of SCAP-validated schools for scanning FDCC configurations.

validation testing.[12] Once accredited, laboratories test products using draft NISTIR 7511. This report contains a list of specific product requirements, needed vendor documentation, and a detailed write-up of the testing that the laboratory must perform. After a product is tested, the laboratory sends a report to NIST's SCAP product validation program. Product validation staff reviews the test report, and the program issues product validations.[13]

A product may be validated as conforming to one or more of the six SCAP component specifications, or separately as conforming to a particular SCAP capability. SCAP capabilities are not product types, but rather ways in which a product may use SCAP. Examples of SCAP capabilities are "authenticated vulnerability scanner" and "patch remediation". The list of capabilities will evolve over time as SCAP is applied to more types of security tools. A current list of available SCAP capabilities and their definitions is available within NISTIR 7511 and on the SCAP validation program Web site.[14] The term "SCAP 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 a set of SCAP capabilities and/or one or more relevant SCAP component specifications. Ordinarily, a product's validation expires after one year unless the product undergoes re-validation—this is intended to keep products aligned with updated SCAP technology since many of the SCAP component specifications are developing and expanding rapidly to meet the evolving needs of the IT security community. Another purpose is to ensure that products continue to incorporate SCAP reference data on an ongoing basis (e.g., lists of known security-related software flaws and configuration issues within selected products). Expiration also causes products that are re-tested to be evaluated using new and improved testing methods.

Validation applies only to the actual version of the product that was tested because NIST can technically only validate conformance to a tested product. However, it is unlikely that future versions of validated products will lose the SCAP functionality that has already been tested within the one-year validation window. 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.

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

[14] http://nvd.nist.gov/validation.cfm

3. Recommendations for Common Uses of SCAP

This section describes several common uses of SCAP version 1.0 and makes related recommendations to SCAP users. Four categories of common uses are discussed in this section: security configuration verification, requirements traceability, standardized security enumerations, and vulnerability measurement. 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 Configuration Verification

Many organizations produce security configuration 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 configuration settings. SCAP enables this via the creation of SCAP-expressed security configuration checklists that can be processed by SCAP-validated authenticated configuration scanners.[15] 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. This can be done before systems are deployed to ensure that they have been secured as intended. The checklist comparison can also be performed as part of auditing and continuous monitoring of deployed systems' security, to ensure that the checklist settings are maintained. It is not normally sufficient to configure a computer once and assume that the settings will be maintained—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.

SCAP-expressed security configuration 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[16] 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.

To help automate security configuration verification, organizations should identify and obtain SCAP-expressed security configuration checklists relevant for their systems' operating systems and applications. In some cases, a security configuration is mandated in policy (for example, the Federal Desktop Core Configuration [FDCC] mandated for Federal agency Windows XP and Vista 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 Federal Acquisition Regulation (FAR) Part 39, Federal agencies must procure IT products with relevant NCP checklists applied.[17] NCP checklists are publicly vetted, and many offer manufacturer-endorsed methods of configuring and evaluating products.

---

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

[16] For the purposes of this discussion, the term "security assessment" is used in a broad sense to encompass many types of verification of system security, including audits.

[17] 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

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. (It is important to note that the current version of SCAP does not provide a capability to automatically implement a security configuration. As of this writing, there are plans to add remediation capabilities to SCAP in the future. However, tools are already available that take SCAP-expressed checklists and apply their settings using proprietary methods.) Once systems have been securely configured, organizations should consistently monitor their security configurations using SCAP-validated authenticated configuration scanners and SCAP-expressed checklists.[18] This allows changes that negatively affect system security to be identified and remediated rapidly, thus minimizing their potential impact.

Federal agencies should use SCAP-expressed checklists in conjunction with SCAP-validated tools to ensure conformance to NIST and OMB security configuration guidance.[19] 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.

In addition to comprehensive checklists, such as a checklist to secure an operating system, more specialized SCAP content is also valuable for security configuration. SCAP's capabilities 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 on patches can be made publicly available by software vendors for their products; organizations can download this data and use it through their SCAP-capable tools.[20]

Another example of checking system characteristics using more specialized SCAP content is identifying signs of a successful system compromise. Many known attacks leave detectable traces on the systems that they compromise. If the method for finding evidence of a particular attack can be determined—such as the checksum of a malicious file or the existence of a particular service—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 check information 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.

---

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

[18] Continuous monitoring of security controls is recommended in Section 2.7 of NIST SP 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*, and in Appendix F, control CA-7 of NIST SP 800-53 Revision 2, *Recommended Security Controls for Federal Information Systems*. Both publications are available at http://csrc.nist.gov/publications/PubsSPs.html.

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

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

3.2 Requirements Traceability

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. Organizations often find it challenging to demonstrate that they have implemented their security controls in accordance with the requirements: to achieve traceability between the low-level controls and the high-level requirements. To assist with this, SCAP content can characterize these mappings.[21] For example, NIST has created SCAP mappings between low-level Windows XP and Windows Vista settings to the high-level controls in NIST SP 800-53. An SCAP-expressed checklist can map 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.

Requirements traceability can provide a substantial savings in effort and cost. It also offers an unambiguous way to communicate the security configuration and the rationale for it. For example, security operations teams can use SCAP content to communicate security configurations to configuration, change, and asset management teams for integration in standardized builds and images.

To produce FISMA compliance evidence for many NIST SP 800-53 controls, Federal agencies should use SCAP-validated authenticated configuration scanners[22] along with SCAP-expressed checklists. The checklists have embedded mappings that can be used to automatically generate NIST SP 800-53 assessment and compliance evidence. In addition, the checklists contain mappings to other high-level policies (e.g., ISO 27001, DOD 8500, FISCAM), and SCAP-validated tools may also output those mappings. Appendix A provides further details on FISMA and SCAP.

3.3 Standardized Security Enumerations

An SCAP enumeration is 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. The context for the identifier may be globally unique (e.g., a CVE identifier), organizationally unique (e.g., an OVAL identifier), or locally unique (e.g., an XCCDF rule identifier). Within SCAP, an enumerated value is often expressed as a pairing of an identifier and a definition.

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. The common understanding achieved through the use of standardized enumerations makes it easier to use security tools, share information, and issue guidance to address security issues. 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 also helps minimize confusion regarding which issue is being referenced, and enables organizations to quickly identify additional information about the issues (e.g., remediation advice).

---

[21] SCAP content can contain mappings to multiple high-level sets of requirements at the same time.

[22] As of this writing, an authenticated configuration scanner is "a product with the ability to audit and assess a target system to determine its compliance with a defined set of configuration requirements using target system logon privileges". See http://nvd.nist.gov/validation.cfm for the current definition.

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.

3.4 Vulnerability Measurement

SCAP enables quantitative and repeatable measurement and scoring of software flaw vulnerabilities across systems through the combination of CVSS, CVE, and CPE. The ability to accurately and consistently convey the characteristics of a vulnerability allows organizations to institute consistent and repeatable mitigation policies throughout the enterprise for software flaw vulnerabilities. For example, an organization could establish a policy that specifies how quickly vulnerabilities must be mitigated based in part on their measures or scores, such as patching the most severe vulnerabilities within a certain amount of time after patches become available.[23] Organizations could also have separate requirements for different types of software to ensure that a vulnerability in a critical application is remediated more quickly than a similar vulnerability in a non-critical application. Another helpful feature is that the major properties of each vulnerability are documented as part of generating each CVSS score. This allows users to understand the basis for each score and to take these properties into account when planning mitigation strategies.

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 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.

[23] 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.

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.
- **Make security settings available through automation.** Software developers should ensure they expose the ability to automatically check their software's underlying configuration settings 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 because there is not a reliable automated method available.
- **Develop security software with SCAP validation requirements in mind.** Before beginning development of SCAP-enabled security software, developers are encouraged to familiarize themselves with the SCAP product validation program test requirements.

4.2 SCAP Content Producers

The following recommendations are applicable to all organizations and individuals who develop SCAP content, including software developers:

- **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 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 six SCAP components, NIST encourages vendors to use all applicable SCAP components for improved effectiveness and interoperability.
- **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[24] 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.

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

Appendix A—Details on Using SCAP for FISMA Compliance

Section 3.1 describes in general how SCAP can benefit an organization in verifying compliance with high-level security requirements, such as those in FISMA. This appendix provides additional details that show how SCAP components are used to automate production of FISMA technical control compliance evidence.

Checklists intended for use in the Federal government are more valuable if they map to FISMA security control baselines. NIST SP 800-53 provides a catalog of security controls for FISMA compliance. It uses control groupings to create three minimum baseline security control sets for Federal information systems—low, moderate, and high impact, as specified in Federal Information Processing Standards (FIPS) Publication 199, *Standards for Security Categorization of Federal Information and Information Systems*.[25] Every system needs to be protected, but the level of protection may vary based on the value of the system and its data—a designation of low, moderate, or high impact estimates the potential impact of a security breach involving that particular system. Accordingly, FISMA specifies the most stringent minimum-security controls for high impact systems, and the least stringent for low impact systems.

To support FISMA compliance, SCAP documents can contain separate policies for low, moderate, and/or high impact systems because in many cases a system is unlikely to be used at all three impact levels (e.g., an enterprise firewall not being low impact). Checklist users can reference the same SCAP documents to assess similar systems that are at different impact levels, which is much more convenient and efficient than having separate documents and files for each impact level.

Another way to tailor SCAP documents to support FISMA compliance is to have them take into account the different operational environments in which systems function. For example, a system located in a secured, agency-owned building and connected to a protected internal network might have different security needs than a similar system used on an employee's home network or directly connected to the Internet. Having profiles that take these environmental differences into account would help checklist users by reducing the amount of time needed to customize the checklists for their systems' environments.

SCAP identifies the following operational environments:

- **Managed or Enterprise** are typically large organizational systems with defined, organized suites of hardware and software configurations, usually consisting of centrally managed workstations and servers protected from the Internet by firewalls and other network security devices.
- **Standalone or Small Office/Home Office (SOHO)** describes small, informal computer installations used for home or business purposes. Standalone encompasses a variety of small-scale environments and devices that can range from laptops, mobile devices, and home computers to telecommuting systems, small businesses, and small corporate branch offices.
- **Custom** environments contain systems whose functionality and degree of security do not fit the other two environments. Three examples of typical Custom environments are **Specialized Security-Limited Functionality**, **Legacy,** and **Federal Desktop Core Configuration**:
  - **Specialized Security-Limited Functionality (SSLF).** An SSLF environment contains systems and networks at high risk of attack or data exposure, with security taking precedence over functionality. It assumes that systems have limited or specialized functionality (not general purpose workstations or systems) in a highly threatened environment, such as an outward-facing firewall or public Web server, or that the systems' data content or mission purpose is of such

[25] http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

value that aggressive trade-offs in favor of security outweigh the potential negative consequences to other useful system attributes such as legacy applications or interoperability. Checklists for this environment are not recommended for home users or for large-scale general-purpose systems. An SSLF environment could also be a subset of another environment.

- **Legacy.** A legacy environment contains older systems or applications that may use older, less-secure communication mechanisms. Other machines operating in a legacy environment may need less restrictive security settings to enable them to communicate with legacy systems and applications. A legacy environment could be a subset of a standalone or managed environment.
- **FDCC.** An FDCC environment contains systems that need to be secured using the FDCC configuration. FDCC configurations are intended to be deployed primarily to managed systems, so FDCC environments usually have characteristics similar to those of managed environments.

While separate XCCDF profiles can be created for each applicable operational environment in which a system might be deployed, it is more helpful to create profiles that take into account the three impact levels and the four operational environments described above. This means that for a particular type of target, the XCCDF document could contain up to 12 different profiles. In most cases, all of these profiles will not be needed because the target is not expected to operate in certain environments, is assigned specific impact levels, or is not expected to have certain impact/environment combinations. For example, an enterprise intrusion detection and prevention system would not normally be run in a SOHO environment, and because of its importance as a security measure, would not have an impact level of low. Another example is that an SSLF environment would be unlikely to have low impact systems.

Table A-1 shows an example of how the minimum password length requirement for a Windows XP Professional system might vary based on impact level and operational environment. N/A entries reflect unlikely impact/environment combinations, so the XCCDF document for this target system would contain 10 profiles. For profiles that show a value of 8 in Table A-1, the profile would use the MinimumPasswordLength-8 rule; for the entries with a value of 12, the profile would use the MinimumPasswordLength-12 rule. Table A-2 provides another view of how these two rules are used by the 10 profiles.

Table A-1. Example of Minimum Password Lengths by Impact and Environment

Environment	High	Moderate	Low
Enterprise	12	8	8
SOHO	12	8	8
SSLF	12	N/A	N/A
Legacy	12	8	8

These profiles would each use a somewhat different combination of rules to specify the requirements imposed by the different impact levels and operational environments. Table A-2 also illustrates how several sample rules might be used by the profiles. Some of the sample rules, such as those for password history enforcement and account lockout reset, are used by all of the profiles. Other rules—such as password length and account lockout threshold—are used by selected profiles only.

Table A-2. Examples of Rule Usage for Windows XP Professional Profiles

Rule Identifier	SP 800-53 Control	Enterprise			SOHO			SSLF	Legacy		
		H	M	L	H	M	L	H	H	M	L
PasswordHistoryEnforcement	IA-5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MinimumPasswordLength-8	IA-5		✓	✓		✓	✓			✓	✓
MinimumPasswordLength-12	IA-5	✓			✓			✓	✓		
AccountLockoutDuration	AC-7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AccountLockoutThreshold-10	AC-7	✓			✓			✓	✓		
AccountLockoutThreshold-50	AC-7		✓	✓		✓	✓			✓	✓
AccountLockoutReset	AC-7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

In addition, Table A-2 illustrates how FISMA requirements can be mapped to specific technical checks. The table's second column lists the SP 800-53 control to which each rule maps. These controls can be specified by creating a group for each SP 800-53 control (e.g., IA-5, AC-7) and making each rule a member of the appropriate group. This makes it easy to determine which SP 800-53 controls a particular profile partially or fully checks, and allows scores to be produced for each defined control. If groups for the controls within a family (e.g., AC-1, AC-2, AC-3) are also placed into a separate group (e.g., AC), then scores can be generated for each family as well as for the individual controls.

FISMA requires many of the same controls as other high-level mandates—for example, DoD 8500.2/8510. Appendix G of NIST SP 800-53, Revision 2 maps controls between the two initiatives to demonstrate that the majority of SP 800-53 controls correspond to one or more controls from DoD 8500.2, with slight differences. Therefore, many checklist components, such as XCCDF rules and OVAL criteria and tests, could be used for both FISMA and other security mandates as long as the components are mapped correctly to the corresponding high-level requirements that originate from each mandate.

Appendix B—Acronyms and Abbreviations

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

API	Application Programming Interface
CCE	Common Configuration Enumeration
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
OMB	Office of Management and Budget
OS	Operating System
OVAL	Open Vulnerability and Assessment Language
SCAP	Security Content Automation Protocol
SOHO	Small Office/Home Office
SOX	Sarbanes-Oxley
SP	Special Publication
SSLF	Specialized Security-Limited Functionality
URL	Uniform Resource Locator
US-CERT	United States Computer Emergency Readiness Team
XCCDF	Extensible Configuration Checklist Description Format
XML	Extensible Markup Language

Appendix C—SCAP Resources

This appendix lists selected SCAP-related resources.

Table C-1. SCAP Component Specifications

Resource	URL
Common Configuration Enumeration (CCE)	http://cce.mitre.org/
Common Platform Enumeration (CPE)	http://cpe.mitre.org/
Common Vulnerabilities and Exposures (CVE)	http://cve.mitre.org/
Common Vulnerability Scoring System (CVSS)	http://www.first.org/cvss/
Extensible Configuration Checklist Description Format (XCCDF)	http://nvd.nist.gov/xccdf.cfm
Open Vulnerability and Assessment Language (OVAL)	http://oval.mitre.org/

Table C-2. 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
NIST National Checklist Program (contains U.S. government SCAP checklists)	http://checklists.nist.gov/
National Voluntary Laboratory Accreditation Program (NVLAP)	http://ts.nist.gov/standards/accreditation/index.cfm
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
NISTIR 7511, *Security Content Automation Protocol (SCAP) Version 1.0 Validation Program Test Requirements (Draft)*	http://csrc.nist.gov/publications/PubsNISTIRs.html
NISTIR 7511 Revision 1, *Security Content Automation Protocol (SCAP) Version 1.0 Validation Program Test Requirements (Draft)*	http://csrc.nist.gov/publications/PubsNISTIRs.html
NIST SP 800-70 Revision 1, *National Checklist Program for IT Products—Guidelines for Checklist Users and Developers (Draft)*	http://csrc.nist.gov/publications/PubsDrafts.html
NVD Official CPE Dictionary	http://nvd.nist.gov/cpe.cfm
SCAP Product Validation Program	http://nvd.nist.gov/validation.cfm
Security Content Automation Protocol (SCAP) homepage Web site	http://nvd.nist.gov/scap.cfm