Transitioning to the Security Content Automation Protocol (SCAP) Version 2

David Waltermire
Computer Security Division
Information Technology Laboratory

Jessica Fitzgerald-McKay
Cybersecurity Solutions
National Security Agency
Fort Meade, MD

This publication is available free of charge from:
https://doi.org/10.6028/NIST.CSWP.09102018

September 10, 2018

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

Abstract

The Security Content Automation Protocol (SCAP) version 2 (v2) automates endpoint posture information collection and the incorporation of that information into network defense capabilities using standardized protocols. SCAP v2 expands the endpoint types supported by SCAP v1 through the explicit inclusion of network equipment, Internet of Things (IoT), and mobile devices in its scope. To automate self-reporting of posture information from endpoint machines, SCAP v2 will integrate with existing network management protocols that include the Internet Engineering Task Force (IETF) Network Endpoint Assessment (NEA) protocols. SCAP v2 will streamline SCAP content acquisition and reuse through its use of the IETF Resource Oriented Lightweight Information Exchange (ROLIE) protocol. Improvements to software version identification and the incorporation of patch information will be made by transitioning from the Common Platform Enumeration (CPE) to Software Identification (SWID) Tags. SCAP v2 provides component-level interoperability via a modular and extensible architecture. This white paper provides a gap analysis of SCAP v1 and an overview of how SCAP v2 will address these gaps; describes the SCAP 2.0 architecture; and provides a plan for completing the work necessary to finalize SCAP v2.

Keywords

architecture; configuration; endpoint; endpoint security; SCAP; security automation; security content automation; Security Content Automation Protocol; software identification; SWID; vulnerability

Disclaimer

Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by NIST, nor does it imply that the products mentioned are necessarily the best available for the purpose.

Additional Information

For additional information on NIST's Cybersecurity programs, projects and publications, visit the Computer Security Resource Center, https://csrc.nist.gov. Information on other efforts at NIST and in the Information Technology Laboratory (ITL) is available at https://www.nist.gov and https://www.nist.gov/itl.

More information on NIST Software Identification and SCAP can be found at https://csrc.nist.gov/projects/Software-Identification-SWID and https://csrc.nist.gov/projects/security-content-automation-protocol respectively.

Table of Contents

List of Appendices

List of Figures

1 Introduction

This document defines a set of design goals for the next major revision of the Security Content Automation Protocol (SCAP), SCAP v2. It identifies a set of issues to address from the previous versions of SCAP, presents a draft SCAP v2 architecture, and outlines an iterative development process for SCAP v2. This document is intended to drive consensus around a vision for SCAP v2, and to identify how the security automation community can collaborate in a public-private partnership to achieve this vision. Some work on the underlying specifications that support SCAP v2 has started; however, much of the work on defining SCAP v2 still needs to be accomplished. This document outlines an approach for SCAP v2 to support a broad range of cyber defense capabilities and requests community input to improve this approach.

Knowing the state of your network is at the heart of all cyber defense capabilities—to know what endpoints are connected to it, if those endpoints are authorized to be on the network, what software is installed on each endpoint, and how that software is configured. In today's networks, endpoints encompass a wide range of connected devices including servers, workstations, network, mobile, and Internet of Things (IoT) devices. Understanding endpoint posture provides the information needed to automate management decisions that support proper network hygiene. Network hygiene practices, as identified by the NIST Framework for Improving Critical Infrastructure Cybersecurity Version 1.1 [1], are important foundational aspects of any enterprise strategy to identify, protect, detect, respond to, and recover from network attacks.

SCAP is a suite of specifications for expressing, exchanging, and processing security automation content. SCAP content describes what endpoint posture information to collect and how to compare this information against a desired state using implementation-neutral, standardized formats. SCAP tools use these formats to drive the collection and evaluation of posture information to determine if software vulnerabilities and misconfigurations exist on a specific managed endpoint. By identifying endpoints with these problems, organizations can make informed decisions about how to remediate these issues.

The initial release of SCAP v2, SCAP 2.0, will enhance the range of security automation capabilities supported by SCAP. This document presents a core SCAP 2.0 architecture and key components. In addition to the architecture, several design goals for SCAP v2 are identified that, when addressed, will increase the interoperability, modularity, and usability of SCAP tools. SCAP v2 implementations will enhance the assessment of network hygiene while providing a strong standardized foundation for more advanced cyber defense techniques. To support the SCAP 2.0 architecture and SCAP v2 design goals, some existing specifications used in SCAP 1.3 need to be revised, and new standards and extensions need to be defined.

The remainder of this document is organized as follows:

- Section 2 presents a set of gaps in the existing SCAP v1 specifications;
- Section 3 presents SCAP v2 design goals, an initial draft SCAP architecture, and how existing international standards can be leveraged to support the SCAP v2 design goals;
- Section 4 describes a transition plan for migrating from SCAP v1 to SCAP v2; and
- The appendices include additional supporting information, including how SCAP v2 can support vulnerability and configuration setting management use cases.

2 Gaps in SCAP v1

SCAP was created as a NIST program in 2006, and the first SCAP v1 specification, SCAP 1.0, was published in 2009. Since this initial version, there have been three SCAP v1 revisions. The current revision, SCAP 1.3, was released in February 2018 as NIST Special Publication (SP) 800-126 Revision 3 [2]. Over its lifetime, a few critical gaps have been identified in SCAP v1 that need to be addressed to provide a more robust solution:

Limited Coverage of Endpoint Types: Maintaining an adequate level of security for enterprise managed endpoints requires up-to-date knowledge of the posture of those devices. SCAP v1 has focused largely on desktops, laptops, and servers, leaving network (i.e., routers, switches, firewalls), mobile, and Internet of Things (IoT) devices lacking in support or not supported. In today's network, these under-supported devices are an increasing area of security risk that must be managed and protected.

Stale Security Posture Information: Many SCAP v1 implementations use a periodic scanning approach when collecting and evaluating endpoint posture. Relying on periodic scanning allows critical security-relevant events to go unnoticed for lengthy periods of time. Even worse, attackers can clean up after themselves between collections, preventing detection. This reduces visibility of security-relevant changes and can create a false sense of security for enterprises.

No Component-level Interoperability: SCAP v1 products that support different use cases often require similar information, such as software inventory and configuration information. While SCAP v1 supports data interoperability, control of collected posture information and evaluation remains with each tool. The lack of standardized interfaces between tools prevents transport layer interoperability, forces redundant posture information collection and evaluation that adds to system and network load, reduces modularity of solutions, and increases integration costs.

Difficult Content Creation and Limited Content Availability: Development of content in SCAP v1 usually requires 1) familiarity with complex SCAP v1 languages; and 2) knowledge of the low-level system artifacts needed to direct collection and evaluate endpoint posture. Finding SCAP content developers that have the necessary skills and knowledge to do both can be challenging. This has increased the time and expense necessary to create content, causing limited content availability.

Limited Software Inventory and Patch Support: In SCAP v1, the Common Platform Enumeration (CPE) [3] provides the ability to identify the software product vendor, name, and version, along with a limited set of additional metadata. Unfortunately, CPEs inadequately express granular metadata about software releases and version ranges needed to accurately identify vulnerable software. Another limitation of CPE is the inability to identify software patches, which further hampers software inventory and vulnerability management capabilities.

A revised SCAP approach is needed to address these gaps. Such an effort is larger than what can be realistically accomplished in a minor SCAP v1 revision. Due to the scope of these changes, a major revision of SCAP, SCAP v2, is needed.

3 SCAP v2

SCAP v2 is the next major release of SCAP. SCAP v2 will focus on addressing the SCAP v1 gaps identified in section 2. SCAP v.2 will continue to support the same use cases of SCAP v1 while also adding support for use cases not supported by SCAP v1. SCAP v2 will be developed iteratively. The first version of SCAP v2, SCAP 2.0, will focus on improving support for software inventory and vulnerability management use cases. Additional use cases, including configuration assessment, will be addressed in future SCAP v2 revisions (e.g., 2.1, 2.2), which will incorporate new capabilities. By working iteratively, SCAP v1 capabilities will be gradually transitioned to SCAP v2, allowing new capabilities (e.g., improved software inventory support) to be adopted in parallel with existing SCAP v1 capabilities.

The following subsections describe the design goals for SCAP v2, outline an initial SCAP 2.0 architecture, and identify several international standards that can be adopted by SCAP 2.0. Engagement with the community is needed to refine and build consensus around the approach and to complete the remaining work.

3.1 Design Goals for SCAP v2

The following SCAP v2 design goals address gaps in SCAP v1 and support new capabilities.

Expanded Endpoint Type Support: In SCAP v2, all endpoint types are explicitly in scope. Supporting additional endpoint types will provide a more uniform endpoint posture assessment capability across all endpoints managed by an enterprise. This will help to unify software inventory, vulnerability management, and configuration setting management capabilities. SCAP 2.0 will focus on supporting desktop, laptop, and server computing devices, and support for additional endpoint types will be addressed in future SCAP v2 revisions.

Event-driven Data Collection: SCAP v2 supports the collection and reporting of endpoint posture in cyber-relevant timeframes (e.g., seconds, minutes) based on reporting of endpoint posture change events. This avoids the lengthy, multi-day scanning frequencies commonly used to assess enterprise endpoint posture today. The SCAP v2 architecture will incorporate standards that support endpoints pushing notifications to management servers when a critical security event is detected. This allows security tools to respond to events in near real-time, rather than hours, or even days later.

Component-Level Interoperability: To support component-level interoperability, SCAP v2 integrates data models with standardized protocols to allow data and results to be exchanged between products. The use of standardized interfaces provides transport layer interoperability, reducing the need for additional integration or connectors. SCAP v2 also incorporates the concept of accessible data repositories, open to authorized users and analytical capabilities. These repositories allow multiple business processes to operate on the same dataset, improving interoperability and supporting reuse of operational data.

Support for Streamlined SCAP Content Creation, Acquisition, and Reuse: SCAP v2 will develop and incorporate simpler formats for use in content creation, including updates to the Extensible Configuration Checklist Description Format (XCCDF) and the

Open Vulnerability and Assessment Language (OVAL), and will develop ways to use system artifacts in posture assessments that do not require deep system expertise. Additionally, by using standardized interfaces and protocols to discover and retrieve SCAP content, SCAP v2 products will be able to detect when new SCAP content is available and automatically acquire suitable content updates based on organizational policies. These same interfaces can be used to discover content for reuse.

Support for Software Patches and Enhanced Software Metadata: The International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 19770-2:2015 Software Identification (SWID) Tag standard [4] provides a data format to express endpoint software metadata that can be shared across the network to identify installed software and inform endpoint posture assessments. SCAP 2.0 will transition from the CPE data format to the SWID tag data format to fully address software inventory and vulnerability management use cases.

By realizing these design goals, SCAP v2 will support timely endpoint posture collection and evaluation that provides a strong foundation for enabling more advanced cyber defense capabilities and the next generation of security automation solutions.

3.2 The SCAP v2 Architecture

To enable advanced cyber defense capabilities, SCAP v2 defines an architecture of clearly scoped and defined security components. SCAP v2 architectural components must be scalable, interoperable, easy to use, extensible, and capable of sharing data with other SCAP 2.0 components. Additionally, the architecture supports the collection and distribution of all information needed by the humans supporting operations and security functions, to include systems administrators, incident responders, network operators, and other similar stakeholders. The SCAP v2 architecture is reflective of real world scenarios of how these stakeholders use data about their networks. The SCAP architecture illustrated in Figure 1 incorporates a number of components, which are described below, and standards, which are described in section 3.3.

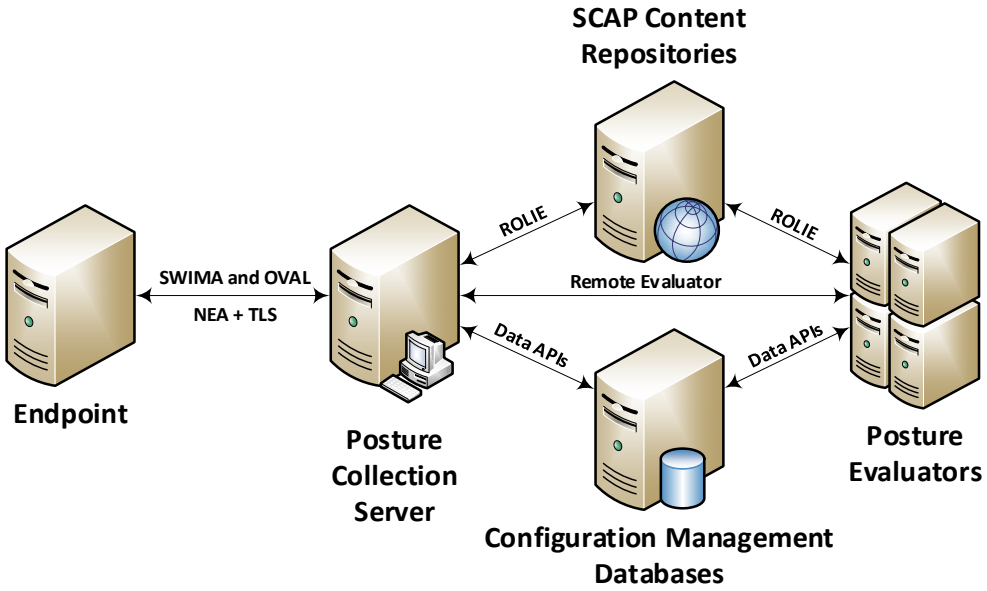

Figure 1: SCAP v2 Architectural Components and Protocols

The SCAP 2.0 architecture, illustrated in Figure 1, consists of the following components:

- **Endpoint:** An Endpoint is the target of a posture assessment. Collection applications will collect software inventory, configuration settings, and other endpoint posture and publish this posture to Posture Collection Servers using standardized protocols and interfaces. These standards will enable Endpoints to be monitored and for changes in posture to be reported automatically. These standards will also allow Posture Collection Servers to query an Endpoint's posture when new or refreshed posture information is required. The initial release of SCAP 2.0 will define protocol requirements for desktops, laptops, and servers; however, all endpoint types are in scope for SCAP v2. Later revisions will define requirements for other endpoint types (e.g., network devices, mobile devices, IoT devices).
- **Posture Collection Server:** This server is responsible for efficiently coordinating the collection of posture from multiple Endpoints. Posture collected from an Endpoint will be published by the Posture Collection Server to the Configuration Management Database (CMDB) where it can then be accessed by other tools. This reduces duplicate posture collection and load on Endpoints that have constrained capabilities, that are under significant load, or that have strict performance requirements. The Posture Collection Server acts as a point of integration for access control and for standardized management protocols including Network Endpoint Assessment (NEA), the Simple Network Management Protocol (SNMP), the Network Configuration Protocol (NETCONF), and mobile device management (MDM) protocols.
- **Posture Evaluator:** By decoupling collection and evaluation and creating standardized interfaces, SCAP v2 enables the distribution of Posture Evaluator service instances across the network. A Posture Evaluator retrieves previously collected posture from a CMDB and requests new posture collection from Posture Collection Servers. Through these interactions, Posture Evaluators gain access to current posture information without having to collect all posture information directly from Endpoints. Posture Evaluators can store evaluation results in CMDBs, making these results available for additional evaluation and use by other tools.
- **Configuration Management Database:** A Configuration Management Database (CMDB) maintains stores of collected endpoint posture information and evaluated data. Before directing the collection of additional posture information from Endpoints, evaluators can query one or more CMDBs to retrieve previously collected information, streamlining data collection needs. CMDBs serve as a consolidated point of access control for authorizing access to posture information, reducing the need for peer-to-peer trust relationships that open attack vectors on the network.
- **SCAP Content Repository:** SCAP v2 will continue to support interoperable, standardized content formats, enabling community-driven content production. Additionally, SCAP v2 supports discovery and retrieval of content through standardized interfaces. One or more SCAP Content Repositories can serve SCAP content including SWID tags, vulnerability records, configuration checklists, and other security automation content. Authorized SCAP components will be able to automatically retrieve new and updated content from local and/or remote SCAP Content Repositories, keeping used content current and improving content reuse. NIST, software vendors, and third-parties will be able to self-publish and manage their own content, while allowing tools to automatically access this content.

Together, the SCAP v2 architectural components provide the timely posture information needed to support advanced analytical processes, to improve automation of security processes, and to support continuous monitoring. This architecture enables Endpoints to self-report posture changes and for other applications to use this posture information. Using this automation, enterprises can generate and use shared cyber threat information, orchestrate automated courses of action, support license management, manage software configurations, and remediate software vulnerabilities. Appendix A— provides examples of using this architecture to identify Endpoint software vulnerabilities and misconfigurations.

3.3 Adoption of International Standards in SCAP v2

The existing specifications in SCAP v1 will be used as a starting point for SCAP v2. As illustrated in Figure 1, SCAP v2 will use both existing and new standards to provide the interfaces, transport protocols, and data models to address the SCAP v2 goals. The following are some of the standards considered for adoption by SCAP v2 that were not in SCAP v1.

ISO/IEC 19770-2:2015: This standard defines structures called SWID tags. A SWID tag provides a structured representation of software package, installation, and patch metadata, including identifying, descriptive, licensing, dependency, and compositional information for the software. SCAP v2 will address software identification gaps by transitioning from the use of CPE to the use of SWID tags for software identification and software reporting. SCAP v2 will continue to use CPE for hardware identification until a suitable replacement is identified.

SWID tags support the identification of installed patches, extensible metadata attributes, and software integrity measurement data not supported by CPE. Through these additional capabilities, SWID tags can support more comprehensive software inventory (including patch details), tamper detection, and application whitelisting. Through the fine-grained identification of software products and patches using SWID tags, improved automation of vulnerability and configuration management is possible in SCAP v2. NIST Internal Report (NISTIR) 8060 [5], provides details on the SWID tag format and discussion of the SWID tag operational model, which is intended to be used as part of SCAP v2.

Internet Engineering Task Force (IETF) NEA and SWIMA: The NEA family of protocols [6] supports the standardized collection of endpoint posture information, and provide an extensible framework for defining standardized and proprietary posture collection capabilities. The Software Inventory Message and Attributes for PA-TNC (SWIMA) [7] is a NEA extension that supports the collection of software inventory information, including SWID tag data. By using the NEA framework, SWIMA can focus on software inventory data collection mechanics, while leaving the details of network transport and delivery to the underlying NEA layers. These same underlying layers can support other NEA extensions in future SCAP releases.

IETF ROLIE: The Resource Orientated Lightweight Information Exchange (ROLIE) protocol [8] provides a standardized mechanism for an SCAP Content Repository that allows security automation content to be discovered, syndicated, and exchanged. ROLIE is a profile of the Atom Syndication Format [9] and the Atom Publication Protocol [10]. SCAP Content Repository implementations based on ROLIE will provide organized collections of SCAP content, vulnerability records, and other security automation data used to automate the collection and evaluation of endpoint posture.

4 SCAP v2 Development Plan

Standardizing solutions centered on the use of SWID tags, NEA protocols, and ROLIE will go a long way towards realizing the SCAP v2 architectural goals. To achieve these goals and the architecture described in section 3.2, standardized schema, protocols, and interfaces are needed. SCAP v2 will be developed over multiple revisions. Initially, SCAP 2.0 will address the standardization needed to support software inventory and vulnerability management use cases. Several missing pieces necessary to complete SCAP 2.0 are being developed now, and require input and review from the security automation community, including:

- **Endpoint Applicability Language:** Security and operations personnel need to know if a vulnerability or set of configuration settings apply to their networks. To do this, they need to map their knowledge of the software installed on Endpoints to vulnerability reports and configuration checklists. In SCAP v1, the CPE Applicability Language [11] is used to associate software and hardware products with a vulnerability or configuration setting information. If software inventory of endpoints is reported using SWID tag information, the CPE Applicability Language needs to be replaced with an applicability language that can match against the larger set of information provided by SWID tags. Development of this applicability language is currently underway and is expected to be published as a NIST Internal Report once the initial draft is complete. Once completed, the applicability language will be piloted as part of the National Vulnerability Database (NVD) vulnerability feeds.[1]
- **ROLIE Extensions:** ROLIE, defined by RFC8322 [8] provides a standardized method to easily identify and retrieve vulnerability records, SWID tags, configuration setting checklists, and other security content to support vulnerability and configuration setting assessment. ROLIE provides a content syndication approach that allows software creators, vulnerability reporters, and configuration setting checklist developers to establish federated repositories of the security content they produce. While ROLIE standardizes the repository communication protocol, each class of information ROLIE serves requires a ROLIE extension to define how the information is indexed and supported. Extensions to ROLIE to support SWID tag [12], vulnerability, checklist, and incident [13] information are currently being developed in the IETF Managed Incident Lightweight Exchange (MILE) [14] and the IETF Security Automation and Continuous Monitoring (SACM) [15] working groups. A ROLIE repository discovery mechanism [16] and a JSON-based version of ROLIE are also being developed in IETF MILE. Once completed, these solutions will be implemented in the NVD, and can be implemented by vendors to support SCAP content discovery and exchange.
- **NIST IR 8085:** In support of the transition from CPE to SWID, an automated method is needed to support the generation of a CPE Name from a SWID tag to support legacy SCAP v1 tools. Draft NISTIR 8085 [17] is a start to support generation of CPE names from SWIDs, but more work is needed to complete this specification.

Once this work is completed, NIST will be able to finalize SCAP 2.0 guidance and requirements.

[1] The National Vulnerability Database provides XML and JSON data feeds that can be accessed at: https://nvd.nist.gov/vuln/data-feeds. A standardized method of interacting with this structured data would allow for more fine-grained data access as compared to the current non-ROLIE feeds.

Subsequent SCAP v2 revisions (i.e., 2.1, 2.2) will focus on supporting additional endpoint types, the configuration setting management use case, and standardizing the data access Application Programming Interfaces (APIs) used to interact with CMDBs. To complete these revisions, the following additional work is needed:

- **Posture Evaluator Protocol**: To support the separation of collection and evaluation, SCAP v2 will need to define a remote evaluation protocol to enable Posture Evaluators to be plugged into the architecture. This protocol will need to define how to request an evaluation and how a Posture Evaluator can request Endpoint posture to be collected from a Posture Collection Server.
- **XCCDF Language Improvements:** XCCDF currently uses the CPE Applicability language. XCCDF [18] needs to be updated to support the Endpoint Applicability Language and SWID tags. There are also several extensibility enhancements requested by the SCAP community that need to be addressed. These updates are expected to be worked in an international standards organization.
- **OVAL-Based Collection and Language Improvements:** A messaging model is needed to allow for the collection of configuration setting information from an endpoint using OVAL [19]. A NEA extension will be developed to allow for event-driven and ad-hoc OVAL-based collection of configuration setting information. The OVAL and SCAP communities have requested improvements to OVAL that need to be considered to better integrate OVAL with other SCAP specifications, to improve OVAL functionality, to make OVAL content easier to create, and to increase overall OVAL adoption. These updates are expected to be worked on in an international standards organization.

For the SCAP v2 work to be successful, significant input from vendors, customers, and academia is needed to develop secure and scalable standards. Additional standards work is needed for the SWID Applicability Language, ROLIE, SWID-CPE transition, OVAL and XCCDF updates, and OVAL-Based Collection over NEA, which will be worked on in appropriate standards bodies identified by NIST based on input from the SCAP community, the National Security Agency (NSA), and the Department of Homeland Security (DHS). Those interested in current IETF standards work for SCAP 2.0 should join us in the IETF to contribute to these efforts. NIST has and will be posting updated public drafts of NIST IR 8085 for comments. NIST, DHS and NSA welcome your input, and look forward to working together on updating SCAP. Please send public feedback to the SCAP Discussion List (scap-dev@nist.gov) or send questions to scap@nist.gov. To join the SCAP Discussion List, please find instructions at: https://csrc.nist.gov/projects/Security-Content-Automation-Protocol/SCAP-Community.

To participate in the IETF please visit the following web pages:

IETF SACM Working Group: https://datatracker.ietf.org/wg/sacm/

Subscribe to the SACM email list: https://www.ietf.org/mailman/listinfo/sacm

IETF MILE Working Group: https://datatracker.ietf.org/wg/mile/

Subscribe to the MILE email list: https://www.ietf.org/mailman/listinfo/mile

Appendix A—Applications of the SCAP 2.0 Architecture

A high-level architecture was presented in section 3.2 of this paper. A key feature of this architecture is that data can be collected once and used many times. This allows data collection to serve multiple use cases. The following subsections in this appendix describe how the architecture can be applied to support vulnerability and configuration assessment respectively.

A.1 Vulnerability Assessment

Vulnerability assessment is a process by which a system/network administrator can decide if their network is subject to a newly announced vulnerability. Vulnerability assessment today relies on manual intervention by vulnerability researchers, vulnerability database owners, and systems/network administrators. Using the SCAP v2 architecture, much of this manual work can be automated and streamlined. Because software inventory data is continuously collected from Endpoints, Endpoints do not need to be rescanned when a new vulnerability is reported. Because software inventory information is communicated using standardized data formats, it can be stored and reused for other purposes, from the mundane (e.g., asset counting, license management) to the cutting edge (e.g., anomalous behavior detection, threat information sharing). This architecture requires pluggable solutions to enhance security operations, and to meet new, as-yet-unknown challenges that will affect our networks in the future.

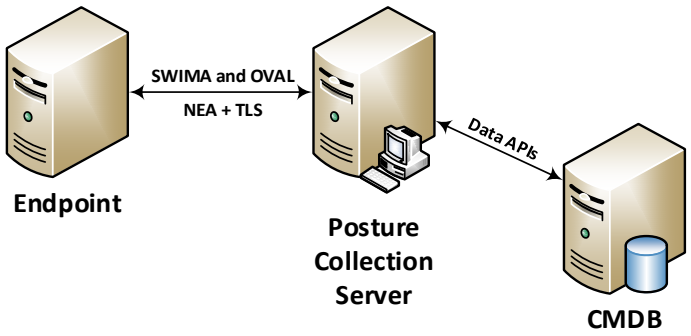

Figure 2: SCAP v2 Architecture: Endpoint Posture Information Collection

In the most basic configuration of the SCAP v2 architecture, as illustrated in Figure 2, the Posture Collection Server collects a baseline of software inventory information from endpoints connected to an enterprise network and stores this posture in the CMDB. Endpoints can push data to the Posture Collection Server when a change is detected in their software load using standardized protocols and interfaces. Using this approach, the CMDB is continuously updated with the latest software inventory information from the Endpoints.

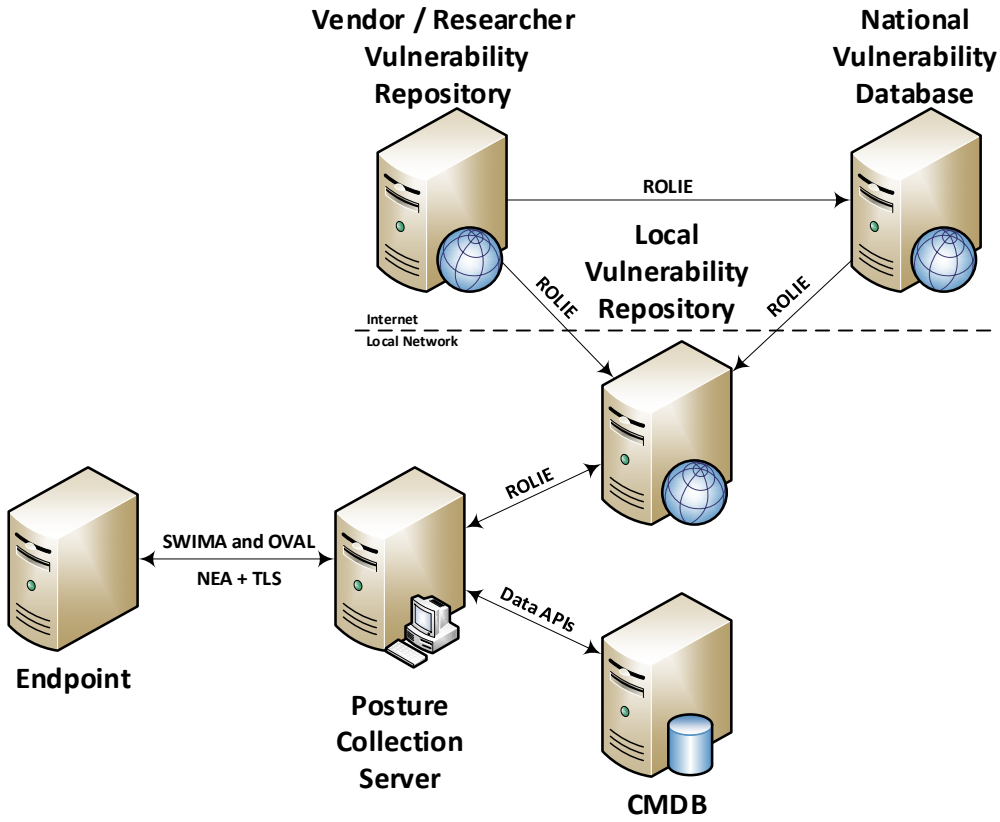

Figure 3: SCAP v2 Architecture: Vulnerability Information Sharing

Vendors and researchers discover vulnerabilities in software products and publish vulnerability bulletins. They can provide publicly available data feeds using an SCAP Content Repository, illustrated in Figure 3 as a Vendor Vulnerability Repository. Although labeled specifically as a "vulnerability repository", these SCAP Content Repositories can also host additional types of security automation information. By publishing vulnerability information in this way, enterprise vulnerability management tools can discover and download this information. Information from vulnerability bulletins are often aggregated by some party, such as the NVD. The NVD will be enhanced to provide an SCAP Content Repository capability, allowing the NVD to be another source for discovering and retrieving vulnerability information under SCAP v2.

Organizations may use a combination of data sources for vulnerability information. An optional Local Vulnerability Repository, implemented as an SCAP Content Repository, is depicted in Figure 3. This local repository may be used by an organization to keep a local copy of vulnerability information to reduce network latency, to ensure this information is highly available, and to provide local control over the data.

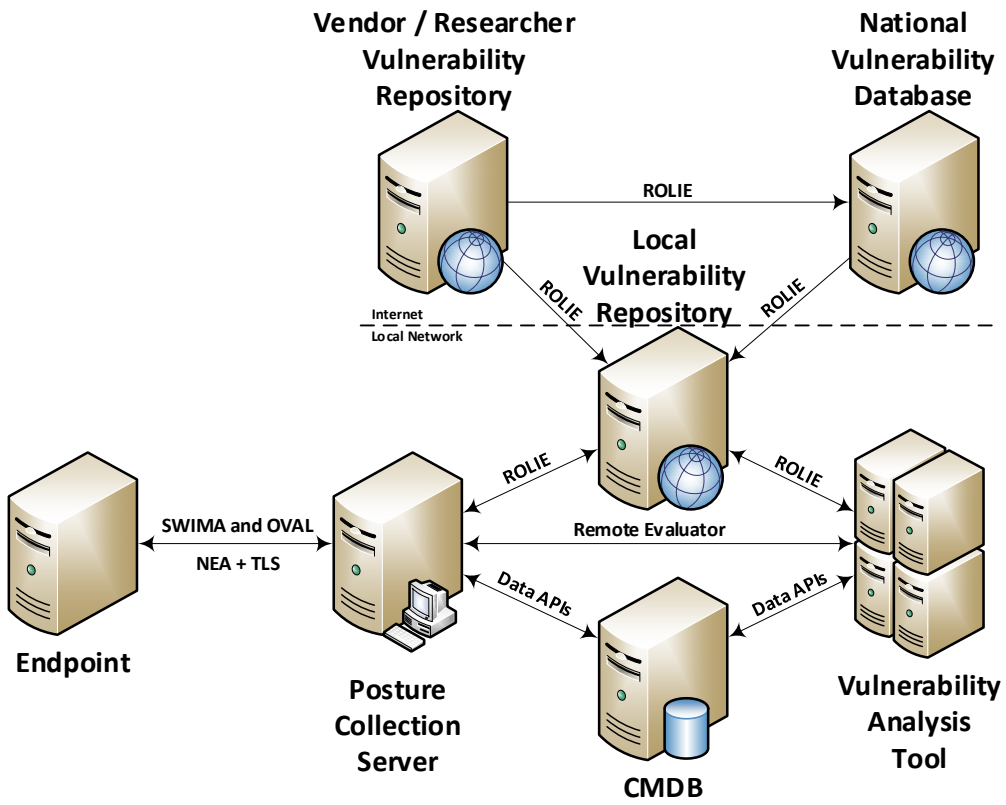

Figure 4: SCAP v2 Architecture: Vulnerability Analysis Capabilities

As illustrated in Figure 4, a Vulnerability Analysis Tool, a type of Posture Evaluator, receives information about a new vulnerability from the Local Vulnerability Repository, a type of SCAP Content Repository. The Vulnerability Analysis Tool uses this information to determine what software versions and configurations must be present for the Endpoint to be vulnerable. The Vulnerability Analysis Tool queries the pre-collected posture information in the CMDB to identify endpoints that have the vulnerable software version loaded. This analysis can also take into consideration collected patch information to determine if the vulnerability has been remediated. If more information is needed (e.g., if the vulnerability only exists when the software is configured in a particular way), the Vulnerability Analysis Tool sends a request to the Posture Collection Server to query all Endpoints with vulnerable software loaded. The collected posture information is stored in the CMDB by the Posture Collection Server. The Posture Collection Server then notifies the Vulnerability Analysis Tool that the requested posture is available in the CMDB. The Vulnerability Analysis Tool can now present information about the vulnerable Endpoints to a user, raise an automated alert, or trigger an automated process to manage the vulnerability.

A.2 Configuration Setting Checklist Assessment

A key feature of the SCAP v2 architecture is that it incorporates standardized interfaces to facilitate data being collected once and used many times. Consider how the SCAP v2 architecture can be applied to configuration setting assessment. Figure 5 illustrates the application of the SCAP v2 architecture to support configuration setting assessment, which is like the vulnerability assessment approach described above.

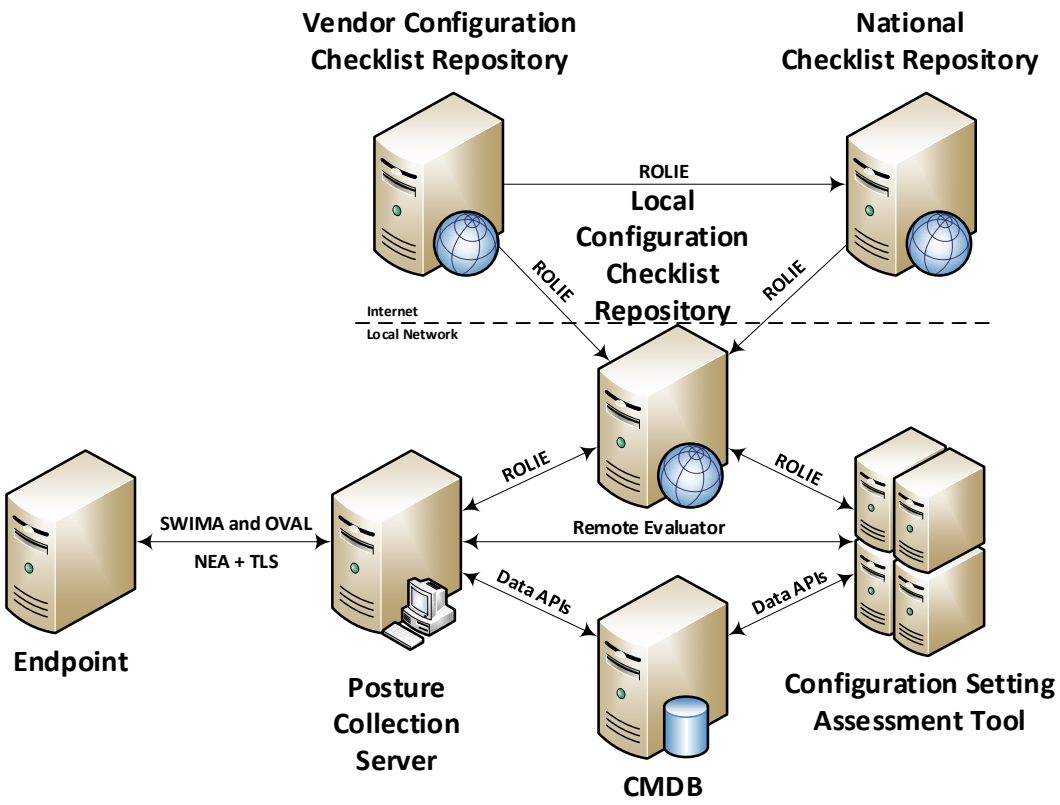

Figure 5: SCAP v2 Architecture: Configuration Setting Checklist Assessment

The same software inventory information is continuously collected by the Posture Collection Server from the Endpoint and stored in the CMDB. Using the collected software inventory of an Endpoint and knowledge the tool has about the operational context of the Endpoint, a Configuration Setting Assessment Tool, another type of Posture Evaluator, can determine the appropriate set of configuration setting policies for each installed software application. The tool can use this information to retrieve SCAP configuration checklist content for each Endpoint from a Configuration Checklist Repository, an instance of an SCAP Content Repository. SCAP configuration checklists can be retrieved from a number of SCAP Content Repository sources, to include Vendor Checklist Repositories, the National Checklist Program (NCP) Repository[2], or a local Configuration Checklist Repository.

Under SCAP v1, NCP checklist resources are provided in different packaging formats, using a variety of irregular distribution mechanisms, requiring manual, human intervention to find, retrieve, and use a given checklist resource. The NCP Repository will be enhanced to provide configuration checklists and checklist metadata using ROLIE, offering a fully automated, standardized method to discover and retrieve SCAP checklist content.

SCAP configuration checklists describe the expected setting values for a target application or operating system. The retrieved checklist content can be used to automatically determine if the

[2] The National Checklist Program Repository can be found at: https://web.nvd.nist.gov/view/ncp/repository.

Endpoint is configured according to policy. The Configuration Setting Assessment Tool can do this by sending the SCAP content, or a pointer to the content, to the Posture Collection Server, which will ask the Endpoint to report on whether its configuration matches the setting values defined in that content. The Endpoint can then report back any divergence from the policy. The Configuration Setting Assessment Tool can then determine if the Endpoint needs to be remediated, monitored, quarantined, or allowed to remain operational on the network based on organizational policies.

By collecting the endpoint's posture information once, SCAP enables better data reuse and cooperation between key security functions. Vulnerability assessment and configuration management both rely on the same basic posture information to secure the network. By continuously collecting the data that is necessary to ensure the security of the network when the posture of the endpoint changes, SCAP v2 can more easily automate solutions to problems that require a great deal of redundant data collection and processing based on SCAP v1.

Appendix B—Acronyms and Abbreviations

Selected acronyms and abbreviations used in this paper are defined below.

API	Application Programming Interface
CMDB	Configuration Management Database
CPE	Common Platform Enumeration
CVE	Common Vulnerabilities and Exposures
DHS	Department of Homeland Security
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
MDM	Mobile Device Management
MILE	Managed Incident Lightweight Exchange
NEA	Network Endpoint Assessment
NETCONF	Network Configuration Protocol
NCP	National Checklist Program
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NVD	National Vulnerability Database
OVAL	Open Vulnerability and Assessment Language
ROLIE	Resource-Oriented Lightweight Information Exchange
RFC	Request for Comments
SACM	Security Automation and Configuration Management
SCAP	Security Content Automation Protocol
SNMP	Simple Network Management Protocol
SWID	Software Identification
SWIMA	Software Message and Attributes for PA-TNC
TNC	Trusted Network Communication
US	United States
XCCDF	Extensible Configuration Checklist Description Format

Appendix C—Glossary

Component	An individually implementable, functional part of the SCAP v2 architecture based on standardized interfaces and transport protocols, that support machine-to-machine communication.
Configuration Management Database	The SCAP component that maintains stores of collected and evaluated data.
Endpoint	"Any computing device connected to a network." Source: RFC 5209 [20] Endpoints are targets of evaluation. Example types of endpoints include servers, workstations, routers, and mobile devices.
Posture	"Configuration and/or status of hardware or software on an endpoint as it pertains to an organization's security policy." Source: RFC 5209 [20]
Posture Assessment	"The process of collecting posture for a set of capabilities on the endpoint (e.g., host-based firewall) such that the appropriate validators may evaluate the posture against compliance policy." Source: RFC 5209 [20] as "Assessment" Validators in SCAP v2 are Posture Evaluators.
Posture Collection Server	The SCAP component responsible for efficiently coordinating the collection of posture from multiple Endpoints.
Posture Evaluator	The SCAP component that analyzes Endpoint posture information.
Posture Information	Data representing the observed posture of an endpoint at the time of data collection. This data can be used to evaluate the security and operational state of an endpoint.
SCAP Content Repository	The SCAP component that serves a variety of SCAP content, allowing authorized SCAP components to automatically discover and retrieve new and updated SCAP content.

Appendix D—References

[1] "Framework for Improving Critical Infrastructure Cybersecurity Version 1.1," National Institute of Standards and Technology, Gaithersburg, MD, 2018. https://doi.org/10.6028/NIST.CSWP.04162018

[2] D. Waltermire, S. Quinn, H. Booth, K. Scarfone and D. Prisaca, "Technical Specification for the Security Content Automation Protocol (SCAP) Version 1.3," NIST Special Publication (SP) 800-126 Rev. 3, National Institute of Standards and Technology, Gaithersburg, MD, 2018. https://doi.org/10.6028/NIST.SP.800-126r3

[3] B. A. Cheikes, D. Waltermire and K. Scarfone, "Common Platform Enumeration: Naming Specification Version 2.3," NIST Internal Report (NISTIR) 7695, National Institute of Standards and Technology, Gaithersburg, MD, 2011. https://doi.org/10.6028/NIST.IR.7695

[4] International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), "Information Technology - Software Asset Management - Part 2: Software Identification Tag," ISO/IEC 19770-2:2015, 2015.

[5] D. Waltermire, B. A. Cheikes, L. Feldman and G. Witte, "Guidelines for the Creation of Interoperable Software Identification (SWID) Tags," NIST Internal Report (NISTIR) 8060, National Institute of Standards and Technology, Gaithersburg, MD, 2016. https://doi.org/10.6028/NIST.IR.8060

[6] Internet Engineering Task Force, "Network Endpoint Assessment (NEA) Working Group Documents," 2018. Available: https://datatracker.ietf.org/wg/nea/documents/ [Accessed 7 September 2018].

[7] C. Schmidt, D. Haynes, C. Coffin, D. Waltermire and J. Fitzgerald-McKay, "Software Inventory Message and Attributes (SWIMA) for PA-TNC," RFC 8412, Internet Engineering Task Force, 2018. https://doi.org/10.17487/RFC8412

[8] J. Field, S. Banghart and D. Waltermire, "Resource-Oriented Lightweight Information Exchange," RFC 8322, Internet Engineering Task Force, 2018. https://doi.org/10.17487/RFC8322

[9] M. Nottingham and R. Sayre, "The Atom Syndication Format," RFC 4287, Internet Engineering Task Force, 2005. https://doi.org/10.17487/RFC4287

[10] J. Gregorio and B. d. hOra, "The Atom Publishing Protocol," RFC 5023, Internet Engineering Task Force. https://doi.org/10.17487/RFC5023

[11] D. Waltermire, P. Cichonski and K. Scarfone, "Common Platform Enumeration: Applicability Language Specification Version 2.3," NIST Internal Report (NISTIR) 7698, National Institute of Standards and Technology, Gaithersburg, MD, 2011.

https://doi.org/10.6028/NIST.IR.7698

[12] D. Waltermire and S. Banghart, "Definition of the ROLIE Software Descriptor Extension," draft-ietf-sacm-rolie-softwaredescriptor-02, Internet Engineering Task Force, 2018. Available: https://tools.ietf.org/id/draft-ietf-sacm-rolie-softwaredescriptor-00.html [Accessed 7 September 2018].

[13] S. Banghart and J. Field, "Definition of ROLIE CSIRT Extension," draft-ietf-mile-rolie-csirt-00, Internet Engineering Task Force, 2018. Available: https://tools.ietf.org/id/draft-banghart-mile-rolie-csirt-03.html [Accessed 7 September 2018].

[14] Internet Engineering Task Force, "Managed Incident Lightweight Exchange (MILE)," 2018. Available: https://datatracker.ietf.org/wg/mile/about/ [Accessed 7 September 2018].

[15] Internet Engineering Task Force, "Security Automation and Continuous Monitoring (SACM)," 2018. Available: https://datatracker.ietf.org/wg/sacm/about/ [Accessed 7 September 2018].

[16] S. Banghart and D. Waltermire, "ROLIE Discovery Mechanism," draft-banghart-mile-rolie-discovery-00, Internet Engineering Task Force, 2018. Available: https://tools.ietf.org/id/draft-banghart-mile-rolie-discovery-00.html [Accessed 7 September 2018].

[17] D. Waltermire and B. A. Cheikes, "Forming Common Platform Enumeration (CPE) Names from Software Identification (SWID) Tags" NIST Internal Report (NISTIR) 8085 (Draft), National Institute of Standards and Technology, Gaithersburg, MD, 2015. Available: https://csrc.nist.gov/publications/detail/nistir/8085/draft [Accessed 7 September 2018].

[18] International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), "Information technology -- Specification for the Extensible Configuration Checklist Description Format (XCCDF) Version 1.2," ISO/IEC 18180:2013, 2013.

[19] The Center for Internet Security, "OVAL Community," 2018. Available: https://oval.cisecurity.org/community [Accessed 7 September 2018].

[20] P. Sangster, H. Khosravi, M. Mani, K. Narayan and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements," RFC 5209, Internet Engineering Task Force, 2008. https://doi.org/10.17487/RFC5209

[21] National Institute of Standards and Technology, "Common Platform Enumeration (CPE)," 7 December 2016. Available: https://csrc.nist.gov/projects/security-content-automation-protocol/scap-specifications/cpe. [Accessed 7 September 2018].