# SCAP Software Asset Management

First published in 2009, the Security Content Automation Protocol (SCAP) is a suite of specifications that standardize the assessment of endpoints across multiple enterprise security use cases including:

* Configuration Management
* Software Inventory Management
* Hardware Asset Management
* Vulnerability Management

SCAP does this by providing a common language with which to talk about different aspects of endpoint assessment including common identifiers for expressing vulnerabilities, platforms, and configuration items, a common format for expressing security configuration guidance, and a common language for expressing what information to collect from endpoints and how to assess it.

SCAP v2 aims to build on this foundation by integrating additional standardized data models, protocols, and interfaces to improve collection and assessment capabilities as well as improve interoperability among tools.

# What is Software Asset Management?

SCAP defines software asset management as the process of: (1) detecting the software installed on endpoints; (2) determining if required or prohibited software is present on endpoints, and; (3) determining whether or not software installed on endpoints is up-to-date.

Software policies can be written using open, standardized formats that allow these policies to be assessed using any SCAP-compliant tool. With these policies, an SCAP tool can gather software inventory information from endpoints and compare it against these policies using automated procedures. The result is a report of all endpoints that diverge from policy and what led to this determination.

A screenshot of a cell phone

Description automatically generated

Figure 1: Identifying Software and Available Patches for Endpoints

# SCAP Workflow

The SCAP architecture employs three components, each supporting different aspects of assessment and data management. The first component is the Collection Manager which manages the collection of information from endpoints and stores that information in the Configuration Management Database (CMDB). The next component is the CMDB which stores and hosts endpoint information and assessment results for later use. Finally, there is the Evaluation Manager which assesses endpoint information against defined policy. Together, these components securely communicate information from the endpoint to the Evaluation Manager in support of the software assessment management use case as shown below.

A picture containing object, microscope

Description automatically generated

Figure 2: SCAP Software Asset Management Workflow

# Benefits

Software asset management using SCAP provides a number of benefits over non-standards-based solutions.

* + **Transparency of Operations:** See exactly what information is collected from endpoints and how it is assessed
  + **Standardized Data:** Software inventory data is represented using open and standardized formats
  + **Event-Driven Updates:** Software inventory data is updated whenever software is installed, updated, or removed
  + **Interoperability Among Products:** Standardized formats, interfaces, and protocols enable interoperability and best-of-breed product selection
  + **Reusable Data:** Real-time software inventory data in the CMDB supports a wide range of use cases beyond software asset management such as configuration and vulnerability management

# Interested in Learning More or Getting Involved?

To learn more about SCAP, please visit the SCAP v2 website (<https://csrc.nist.gov/Projects/Security-Content-Automation-Protocol-v2>). If you have any questions or comments, please send an email to [scap@nist.gov](mailto:scap@nist.gov).

To get involved in the development of SCAP, please join the SCAP Community (https://csrc.nist.gov/projects/security-content-automation-protocol-v2/scapv2-community) and share your thoughts.