AUTOMATED PROCESSES IN COMPUTER SECURITY

Maroš Barabas
Doctoral Degree Programme (3), FIT BUT
E-mail: ibarabas@fit.vutbr.cz

Supervised by: Petr Hanáček
E-mail: hanacek@fit.vutbr.cz

ABSTRACT

This article describes security standards for automating processes and their implementation in Open-Scap library. The main aim is to explain new approaches to vulnerabilities in computer security and to analyze the open-scap library as an implementation of these standards.

1 INTRODUCTION

In September 2008 on the 4th Annual IT Security Automation Conference were presented standards for automation of vulnerability management, security measurement, and compliance [1]. These standards are collected in Security Content Automation Protocol (SCAP) that is a synthesis of interoperable specifications such as Common platform enumeration or public well known Common Vulnerability and Exposures (CVE). SCAP comprises these specifications for managing and expressing security related information and provides related reference data such as unique vulnerability identifiers in standardized ways. The third party tools use standards to measure systems to find vulnerabilities and assess the score of the findings to evaluate the possible impact.

Our work is to implemenet these standards in a simple and easy to use library as a first layer above the operating system to allow software developers to make tools for security measurement, compliance and vulnerability management.

2 SCAP STANDARDS

The SCAP specification is a protocol that describes what and how we communicate in a process of automated securing of systems. Standardization of this approach enables interoperability of various products and service of various manufacture and reduces content-based variance in operational decisions and actions.

2.1 SCAP PROTOCOL

In order to explain how SCAP communicates we need to split the protocol into single individual specifications. Figure 1 shows a table of common specifications that are included in SCAP. In the first and second column is an abbreviation and the name of the particular standard and in the third column is a short definition.

CVE	Common Vulnerability Enumeration	Standard nomenclature and dictionary of security related software flaws
CCE	Common Configuration Enumeration	Standard nomenclature and dictionary of software misconfigurations
CPE	Common Platform Enumeration	Standard nomenclature and dictionary for product naming
XCCDF	eXtensible Checklist Configuration Description Format	Standard XML for specifying checklists and for reporting results of checklist evaluation
OVAL	Open Vulnerability and Assessment Language	Standard XML for test procedures
CVSS	Common Vulnerability Scoring System	Standard for measuring the impact of vulnerabilities

Figure 1: SCAP standards overview

Below is the list of definitions with further explanation of standards with common use cases.

Common Vulnerability Enumeration [2] is a dictionary of publicly known information security vulnerabilities and exposures. The content of CVE dictionary is a list of CVE identifiers that can reference the appropriate CCE or CPE IDs. The dictionary is stored in NVD (National Vulnerability Database). Common use case for CVE content is to check the machine for the presence of particular vulnerabilities and exposures by iterating through the CVE dictionary.

Common Configuration Enumeration [3] provides unique identifiers for system configuration issues. The main aim is to facilitate fast and accurate correlation of configuration data across multiple tools or sources. CCE Identifiers are used to associate checks in configuration assessment tools with statements in configuration best-practice documents. That means that pthe olicy in an XCCDF document has a unique identifier in all documents defining same problem.

Common platform enumeration [4] is a structured naming scheme for information technology systems, platforms, and packages. It is based upon the syntax for Uniform Resource Identifiers (URI). For example CPE *cpe:/a:microsoft:office:2003* references the application Microsoft Office version 2003. CPE includes a formal name format, a language for describing complex platforms, a method for checking names against a system, and a description format for binding text and tests to a name.

The eXtensible Configuration Checklist Description Format [5] is a specification language for writing security checklists, benchmarks, and related kinds of documents. An XCCDF document represents a structured collection of security configuration rules for some set of target systems. It is an XML document that can be easily transformed to many document formats for generating security guidance or showing results in a user readable formats.

Open Vulnerability and Assessment Language [6] is an information security standard for defining and transfering security content across security tools and services. OVAL includes an open language used to encode system details and repositories with contents for various systems. OVAL Language contains three XML schemes for the next three steps of the assessment process. An OVAL System characteristics schema for representing system information, an OVAL

Definition schema for expressing a specific machine state, and an OVAL Results schema for reporting the results of an assessment. Main use cases for OVAL in information security products are vulnerability assessment, configuration and patch management and policy compliance.

Common Vulnerability Scoring System [7] is a vulnerability scoring system designed to provide an open and standardized method for rating IT vulnerabilities. CVSS consists of three groups: Base, Temporal and Environmental. Each group provides a score (from 0 to 10) and a vector (textual representation) of the security impact of a vulnerability. The Base group represents the intrinsic qualities of a vulnerability, the Temporal group reflects characteristics of a vulnerability that change over time and the Environmental group represents user's environment related characteristics. The main purpose of CVSS is to provide an easily comparable score that reflects the overall security impact of each vulnerability.

2.2 SCAP CONTENT

SCAP content is divided into two main parts. In the first part are security checklists (or benchmarks) that provide detail guidance on how to check or secure the configuration of selected operating systems. The second part is focused on SCAP related reference data such as enumerations and mapping data feeds. For more information on the contents of specifications see the section 2.1.

Main use case for the assessment process from the view of an SCAP content is scanning the given operating system for presence of known vulnerabilities. In dependence on the operating system we choose one security guidance that is represented in an XCCDF format. Each security policy of the guidance consists of a human readable text definition of the vulnerability and a reference (unique ID) to the entry in the definition file of the checking system. A policy may also include other enumeration IDs such as CCE or CVE. We get the test to express the machine state for the given vulnerability from the definition file and assess its presence on the tested machine.

The contents for many various systems and for various security guidances are held throughout the community and in the repositories of the U.S. National Institute of Standards and Technology (NIST) and the MITRE Corporation as independent not-for-profit organization.

3 IMPLEMENTATION AND OPEN-SCAP

There are many implementations of SCAP standards in various, mainly proprietary products, and very few free or open-source. As mentioned in the introduction, we are working on a library that implements all SCAP standards with the main intent to abstract the low level implementation of specifications. Figure 2 shows the design of the open-scap library with emphasis on the security aspect of the implementation. The first layer is the operating system on the given local machine. The first security layer is represented by probes - very small binaries that are responsible for executing commands from the library. Probes are separated from each other by selinux (Security-Enhanced Linux) - mechanism with definitions of policies for access control. With selinux we restrict the behavior of each probe to access only system functions it needs and is approved to access. The developer of a security tool can access probes only through the public API of the library. The final application that uses opens-scap is represented by a daemon in figure 2 that belongs to a less secure layer and the security has to be taken care of by the developer of the application.

The input of the library is a definition file in an XML format with SCAP content and the output is a results file and a system characteristics file for the tested machine. Results are stored in an internal model and can be exported to an XML file. Processing of the results data is left to application.

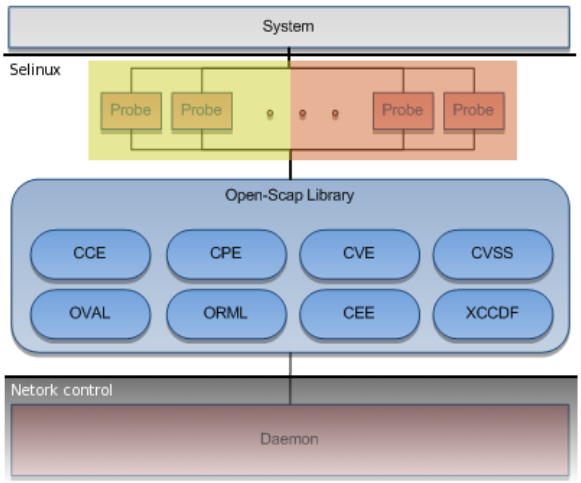

Figure 2: Open-scap library

4 CONCLUSION

SCAP standards will become a very effective way to measure and assess computer security. The fact that it is all based on open standards can bring the broadest possible range of use cases and improvements from security automation community. Open-scap library is an open source implementation of SCAP standards that provides easy way to access higher layer of abstraction and implement new tools to automate security processes upon security automation protocol. In its current state, open-scap library supports all primitive functionality of particular specifications including loading given contents to an abstract model and executing benchmark models on the local machine. The priority for future development is platform independence of library, coverage of all the possible options of definitions and simplicity of use. In the development participate Red Hat, Inc. and G2, Inc.

My personal contribution to project is analysis and development work on library, implementation of particular SCAP standards and responsibility for SCAP content creation and validation for Red Hat operating systems.

Acknowledgement: This work was partially supported by the BUT FIT grant FIT-S-10-1 and the research plan MSM0021630528.

REFERENCES

[1] Barrett, M.: Introduction to SCAP Standards, In: 4th Annual IT Security Automation Conference, Gaithersburg, 2008

[2] Christey S. M., Baker D. W., Hill W. H., Mann D. E.: The Development of a Common Vulnerabilities and Exposures List, In: Second International Workshop on Recent Advances in Intrusion Detection, Purdue University, West Lafayette, Indiana, USA. September 8, 1999

[3] Mann, D.: An Introduction to the Common Configuration Enumeration, URL: `http://cce.mitre.org/about/documents.html`, 2008

[4] Buttner A., Ziring N.: Common Platform Enumeration (CPE) Specification, Version 2.2, March 2009

[5] Quinn, S.D., Ziring, N.: Specification for the Extensible Configuration Checklist Description Format (XCCDF), Version 1.1.4, NIST a NSA, 2008

[6] OVAL Design Document, Version 5.0, June, 2006

[7] A Complete Guide to the Common Vulnerability Scoring System Version 2, URL: `http://www.first.org/cvss/cvss-guide.pdf`, June 2007