SCAP compliant Android Vulnerability Scanner in OVAL

The Theory :

SCAP: The Security Content Automation Protocol (SCAP), provides a standard approach to maintain the security of enterprise systems by automatically verifying the presence of patches, checking system security configuration settings, examining systems for any vulnerabilities, etc. SCAP was defined and started by the National Institute of Standards and Technology (NIST) and its partners in industry. SCAP is basically a super-standard consisting of many individually maintained standards.

The SCAP suite of specifications standardize the nomenclature and formats used by the automated vulnerability management, measurement, and policy compliance products. The SCAP comprises of several open standards that are widely used to enumerate software flaws and configuration issues related to security. Applications which conduct security monitoring use the standards when measuring systems to find vulnerabilities, and offer methods to score those findings to evaluate the possible impact. A vendor of a computer system configuration scanner can get his product validated against SCAP, demonstrating that it will interoperate with other scanners and express the scan results in a standardized way.

The latest version of SCAP: SCAP 1.3 has around 12 different specifications as follows:

- Languages
    1) XCCDF: XCCDF (The Extensible Configuration Checklist Description Format) is a test definition language for writing security checklists, benchmarks, etc. An XCCDF document represents a structured collection of security configuration rules for some set of target systems.
    2) OVAL: The Open Vulnerability and Assessment Language (OVAL) standardizes the three main steps of the assessment process: representing configuration information of systems for testing, analyzing the system for the presence of the specified machine state (vulnerability, configuration, patch state, etc.) and reporting the results of this assessment.

       (Basically, XCCDF states “what” to evaluate while OVAL states “how” to evaluate)
    3) OCIL: The Open Checklist Interactive Language (OCIL) defines a framework for expressing a set of questions to be presented to a user and corresponding procedures to interpret responses to these questions.
    4) Asset Identification: The Asset Identifier is used to uniquely identify assets based on known identifiers or known information about the assets. It plays an important role in an organization's ability to quickly correlate different sets of information about assets.

5) ARF: The Asset Reporting Format (ARF) is a data model to express the transport format of information about assets, and the relationships between assets and reports. The standardized data model facilitates the reporting, correlating, and fusing of asset information throughout and between organizations.

- Identification schemes:

6) CCE: Common Configuration Enumeration (CCE) provides unique identifiers for configuration issues to facilitate correlation of configuration data across multiple information sources and tools.

7) CPE: Common Platform Enumeration (CPE) is a standardized method of describing and identifying classes of applications, operating systems, and hardware devices present among an enterprise's computing assets. It is used for mapping platforms to vulnerabilities or policy statements.

8) CVE: Common Vulnerabilities and Exposures (CVE) is a dictionary of common names or identifiers for publicly known information security vulnerabilities. If a CVE identifier is found in any of the reports of security tools, then the remedy for it can be searched in the CVE compatible databases.

9) SWID: Software identification (SWID) tags define a structured metadata format that identify the software product, characterize the product's version, the organizations and individuals that had a role in the production and distribution of the product etc.

- Metrics

10) CVSS: The Common Vulnerability Scoring System (CVSS) algorithm scores a given vulnerability based on its likely danger on a scale of 0 to 10. It is mainly used for prioritizing responses to the detected vulnerabilities and weighing the cost of remedies for that vulnerability against allowing a vulnerability to persist.

11) CCSS: Similar to CVSS, The Common Configuration Scoring System (CCSS) is a set of measures of the severity of software security configuration issues.

- Integrity

12) TMSAD: The Trust Model for Security Automation Data (TMSAD) describes a common trust model which is composed of recommendations on how to use existing specifications to represent signatures, hashes, key information, and identity information in the context of an XML document within the security automation domain.

The thesis idea :

The inspiration for this thesis is primarily derived from the research paper : "Ovaldroid: An OVAL-based vulnerability assessment framework for Android" where a lightweight android scanner was developed in OVAL to assess the vulnerabilities on Android Operating System. The authors had developed this project in the Year 2013 however, MITRE released the official schemas for OVAL for android in 2015. So I am planning to use these schemas to write an OVAL definition file for android and to assess the Android operating system by considering a few defined vulnerabilities from NVD (National Vulnerability Database). To start with, I will be

concentrating on the vulnerabilities present on Android 6.x.x which can be found at : **https://web.nvd.nist.gov/view/vuln/search-results?adv_search=true&cves=on&cpe_version=cpe:/o:google:android:6.0.1**
After achieving this preliminary objective, I will work towards considering the vulnerabilities in the previous versions as well as the 7.0.x version of android.

As a future scope to this thesis, I would like to link the scanner to a server where all the processing of the system files will take place on the server so that in an actual scenario of a scanner scanning for all the 453 known vulnerabilities as given by the NVD, the battery of the phone won't be affected.