A Vulnerability Assessment Tool based on OVAL in Linux System

Youngmi Kwon[1], Hui Jae Lee[2], Geuk Lee[3]

[1] Dept. of InfoCom, Chungnam National University, Daejeon, South Korea
ymkwon@cnu.ac.kr
[2] Dept. of Computer Engineering, Hannam University, Daejeon, South Korea
lhuijae@ai.hannam.ac.kr
[3] Dept. of Computer Engineering, Hannam University, Daejeon, South Korea
leegeuk@hannam.ac.kr

Abstract. Open Vulnerability Assessment Language(OVAL) is a standard language which is used to detect the vulnerability of local system based on the system characteristics and configurations. It is suggested by MITRE. OVAL consists of XML schema and SQL query statements. XML schema defines the vulnarable points and SQL query detects the vulnerable and weak points. This paper designed and implemented the vulnerability assesment tool with OVAL to detect the weak points in Linux System. It has more readability, reliability, scalability and simplicity than traditional tools.

1 Introduction

The vulnerability assessment tool is a security tool to diagnose the computer system and detect the weakness in advance to keep the system's status safe. The vulnerability assessment tools are broadly classified as host-based assessment tool, network-based assessment tool and application assessment tool to detect the specific applications' attack-ability. Existing vulnerability assessment tools detect the system's weakness by executing the attack code such as exploit scripts [1]. But, individual tools don't have common criteria with vulnerability detection and vulnerability assessment scripts are implemented with various programming languages. So it is difficult to know which tools provide more correct diagnoses, as well as the prices to develop and maintain the assessment script gets higher. MITRE suggested the OVAL (Open Vulnerability Assessment Language) to overcome these limitations. OVAL is a standard language to assess the fragility of the local system based on the information of the system's characteristics and the configurations. Basically OVAL defines the weakness of CVE with XML schema. Using these XML schemas, it constructs and executes the query statements to detect the weak points.

This paper designed the host-based vulnerability assessment tool in the RedHat Linux System with OVAL which has been proposed by MITRE. In the chapter two, related works, we analyzed and compared the existing assessment tools with OVAL.

2 Related Work

2.1 The Vulnerability Assessment Tool

The vulnerability assessment tool is a sort of security tool to keep the systems more safe by diagnosing the weak points in the computer systems in advance and providing the solutions and the proper patch information. It is also called as vulnerability scanner or security scanner. These scanners are classified as a host scanner and network scanner in accordance with the checking contents [2]. Host scanner is installed at each operator's platform. It searches the security problems which can be caused by the administrator's mistakes or mis-configurations [3]. The network scanner assesses the portable weak points which can be attacked by the external hackers.

The vulnerability scanner usually uses the detection scripts such as exploit to find weak points. But currently used commercial or free codes have a problem that the detection results are not reliable, because they apply some different criteria in the vulnerability assessment and the codes are made with different description languages with wide variety. Table 1 shows the free vulnerability assessment tools and their used languages in detection scripts.

Table.1. Free Vulnerability Assessment Tools and Their Languages

Name of Tools	Type	Used Languages
Tiger	**Host scanner**	**C, Shell Script**
COPS	**Host scanner**	**C, Shell Script**
Nessus	**Network scanner**	**NASL**
SARA	**Network scanner**	**C, Perl**
SAINT	**Network scanner**	**C, Perl**
Sscan	**Network scanner**	**C**
Vlad	**Network scanner**	**Perl**

2.2 OVAL

OVAL is the common language for security experts to discuss and agree upon technical details about how to check for the presence of vulnerabilities on a computer system. The end results of the discussions are OVAL queries, which perform the checks to identify the vulnerabilities [1, 4].

OVAL queries are written in SQL and use a collaboratively developed and standardized SQL schema as the basis for each query. OVAL queries detect the presence of software vulnerabilities in terms of system characteristics and configuration information, without requiring software exploit code. By specifying logical conditions on the values of system characteristics and configuration attributes, OVAL queries characterize exactly which systems are susceptible to a given vulnerability.

OVAL queries are based primarily on the known vulnerabilities identified in Common Vulnerabilities and Exposures (CVE), a dictionary of standardized names and descriptions for publicly known information security vulnerabilities and exposures developed by The MITRE Corporation in cooperation with the international security community. CVE common names make it easier to share data across separate network security databases and tools that are CVE-compatible. CVE also provides a baseline for evaluating the coverage of an organization's security tools, including the security advisories it receives. For each CVE name, there are one or more OVAL queries. Fig.1 shows the operational procedure of the assessment tool based on OVAL.

Fig.1. Operational Procedures of OVAL

2.3 OVAL Schema

XML and SQL languages have a strong point of defining the vulnerable points most logically and clearly. Those languages can be understood by computer systems, and much readable to security experts. XML schema's purpose is to define vulnerabilities in the system and consists of the common schema and the per-platform schema. The common schema describes the fundamental information required to define vulnerabilities. And the per-platform schema describes operational elements to check on each platform.

3 Design of Vulnerabilities Assessment Tool

3.1 Overall Structure

In this paper, we designed the vulnerability assessment tool designed for RedHat Linux Platform with OVAL schema suggested by MITRE. Its overall structure is as in Fig. 2.

Data File consists of “INSERT DATA” part and “OVAL queries” part. In “INSERT DATA” part, the lists of data to be collected by the “System Information Collecting Module” are presented. In “OVAL queries” part, the conditions to detect the system’s vulnerability based on the system information collected from input data using “query interpreter” module are described in the form of SQL query statements.

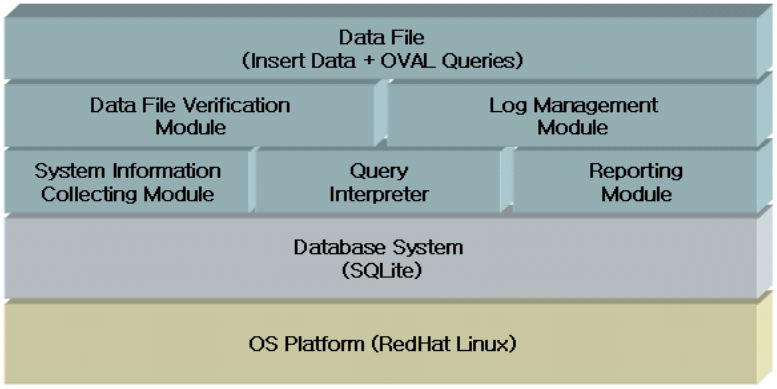

Fig.2. Overall Structure of Vulnerability Assessment Tool

“Data File Verification Module” verifies whether the given data file is correct or not. “Log Management Module” deals with the errors which can be occurred in the system. “System Information Collecting Module” has two roles in the vulnerability assessment tool. The one role is to collect various system information such as configuration setting information, software installation information, file information and process information based on “INSERT DATA.” And the other role is to update database status based on the collected data. Because the “OVAL Queries” part is described with SQL language, OVAL-based vulnerability assessment system should contain DBMS (Database Management System). In our design, we used SQLite as DBMS. It operates in a file-base. We summarized the general characteristics of SQLite in table 2.

Table.2. Characteristics of SQLite

Characteristics	Description
SQL compatibility	Support almost syntax of SQL92
Speed	Two times faster than the conventional DBMS in general command processing

Size	About 25K lines C code. Very lightweight DBMS.
Database	All of Database is included in one file.
Operational Environment	Can be executed without the help of other libraries.

3.2 Database Design: Schema

System Data gathered by "System Information Collecting Module" is stored in database. OVAL Query statements are applied to this database to find corresponding vulnerabilities. Tables of database are constructed using OVAL schema of individual OS platform. In the RedHat-series Linux Platform, we designed the required schema; File Metadata Schema, Internet Server Daemon Schema, Passwd File Schema, Shadow File Schema, Process Information Schema, RPM Information Schema, Information for Comparison with Susceptible RPM Versions Schema and Operating Platform Information Schema. As an example, the File Metadata Schema is shown in Table 3.

Table.3. File Metadata Schema

Field	Description
FilePath	Absolute path of a file
FileType	Directory, normal file, device file, etc.
UserID	Owner ID of a file
GroupID	Group ID of a file
time	Access time (Atime), Status Change Time (Ctime), Data Update Time(Mtime)
MD5	MD5 hash for a file
Permission bits	SUID, SGID, STICKY, UREAD, UWRITE, UEXEC, GREAD, GWRITE, GEXEC, OREAD, OWRITE, OEXEC

3.3 Construction of System Information Base

"System Information Collecting Module" plays two roles; 1) collecting the required system information to assess the vulnerabilities in the system, 2) reflect that information to the database designed in subsection 3.2. The data list which this module should collect is listed up in "INSERT DATA" part in Fig. 2. OVAL uses this "INSERT DATA" part to reduce the time of collecting required system information. In other words, "INSERT DATA" part lists up not all the information of installed packages and files, but only the information items required to assess the vulnerability of the system. "System Information Collecting Module" consists of 8 sub-modules. Their names are taken after the corresponding schema. They are File Metadata Collecting Sub-module, Internet Server Daemon Information Collecting Sub-module, RPM Information Collecting Sub-module, RPM Version Comparison Sub-module, Password File Information Collecting Sub-module, Process Information Collecting Sub-module, Shadow File Information Collecting Sub-module and Operating Platform Information Collecting Sub-module.

Fig. 3 is the File Metadata Table which is one of the tables produced by the operation of "System Information Collecting Module." As same as this table, other information required to assess the system is collected in the form of SQLite Database Table.

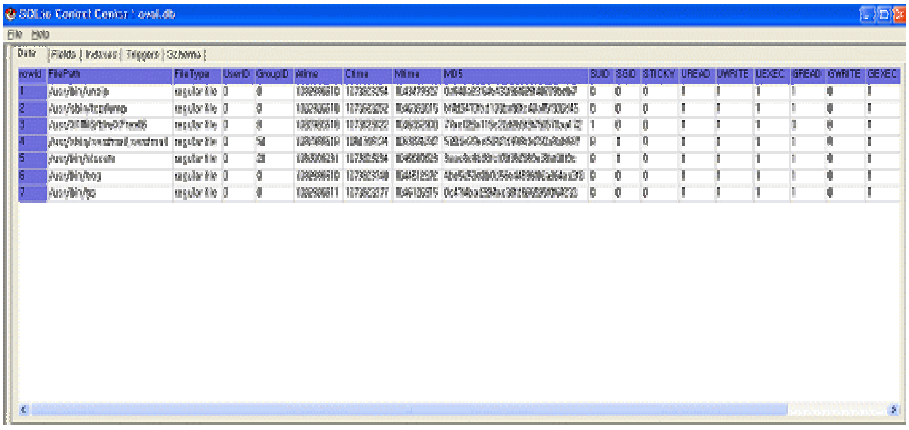

Fig.3. File Metadata Table produced by the "System Information Collecting Module**"**

3.4 Execution of Query Interpreter Module

"Query Interpreter" detects the existence of vulnerability of the system by applying the OVAL queries stored in "Data File" to the system information stored in SQLite Database. The return value of the OVAL query statement is CVE ID if the vulnerability is detected. In the case of detection, "Reporting Module" reports this susceptibility on the operator's screen. Fig. 4 is an example report output when "Query Interpreter" detects some vulnerability in the system. In this example, detected CVE IDs are CAN-2003-0165, CAN-2003-0547, and so on.

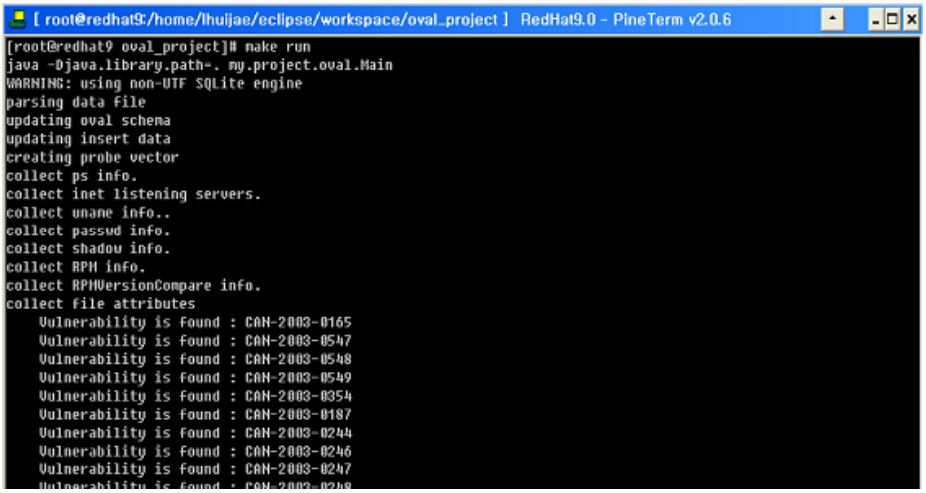

Fig.4. Vulnerability report output when the vulnerabilities are detected in the system

4. Comparison with Previous Tools

We designed and implemented the OVAL-based vulnerability assessment tool operating on RedHat Linux Platform. There are some other existing tools used in UNIX-like platform such as Tiger or SATAN. They have specific scripts and specific goals. Our design follows the standard guideline suggested in the MITRE. So our tool is very general-purpose assessment tool and has as similar benefits as OVAL concept. They are following:

- A simple and straightforward way to determine if a vulnerability exists on a given system
- A standard, common schema of security-relevant configuration information
- For each CVE entry, one or more SQL queries precisely demonstrate that the vulnerability exists
- Reduces need for disclosure of exploit code as an assessment tool
- An open alternative to closed, proprietary, and replicated efforts
- A community effort of security experts, system administrators, software developers, and other experts
- Freely available vulnerability content for public review, comment, or download from the Internet
- Industry-endorsed via the OVAL Board and OVAL Community Forum

5. Conclusions

OVAL is the common language which has many benefits and capabilities of checking for the presence of vulnerabilities on a computer system by specifying logical condi-

tions on the values of system characteristics and configuration attributes. The vulnerability assessment tool suggested in this paper is based on OVAL scheme, so it is more efficient and flexible. We designed the overall structure with five modules, one Data File and SQLite DBMS.

Existing assessment tools only check the existence of the vulnerabilities by checking the checklists mainly listed in [3]. But the suggested tool can not only check the weak points but also define new checklists in the form of XML and SQL syntax.

Traditional tools only check the mainly weak points which have been aimed to by the attackers. But the suggested tool can check all the weak points registered in CVE list at once.

In addition to them, because existing tools apply somewhat different description languages with wide variety each other, their detection results are not reliable. OVAL-based vulnerability methods are getting higher estimation by the security experts, so the tools on various OS platforms will be developed continuously in the future.

Acknowledgement

This work was supported by a grand No.R12-2003-004-02002-0 from Korea Ministry of Science and Technology.

References

1. Jung Hee Kim, “The Trends of International Standardization on Vulnerability Assessment of Computer Systems,” The Trends in International Information Security, Korea Information Security Agency, June 2003
2. Ragi Guirguis, "Network and Host-based Vulnerability Assessments," http://www.sans.org, SANS Institute, February 2004.
3. UNIX Security Checklist v2, http://www.cert.org, AusCERT, October 2001
4. http://oval.mitre.org
5. M. Wojcik, T. Bergeron, T. Wittbold and R. Roberge, “Introduction to OVAL,” http://oval.mitre.org/documents/, MITRE Corporation, November 2003.