基于OVAL的安卓漏洞检测评估系统

万 燕 赵 希 王国林
(东华大学计算机科学与技术学院 上海 201620)

摘 要 传统漏洞检测工具检测时间长,占用大量系统资源,需要对系统进行模拟攻击,难以应对越来越复杂的安卓漏洞威胁。提出了一种"C/S"架构的、基于开放漏洞评估语言(OVAL)的安卓漏洞检测评估系统。这种架构将大部分评估工作放在控制台端执行,减少了对安卓系统性能的影响,其以OVAL作为漏洞评估标准,在保证评估高精度的同时也具有更好的开放性和可扩展性。

关键词 漏洞检测,开放漏洞评估语言,安卓

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.04.017

Android Vulnerability Detection and Assessment System Based on OVAL

WAN Yan ZHAO Xi WANG Guo-lin
(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

Abstract It is difficult to deal with more and more complex security vulnerabilities for the traditional detection tool, which takes a long time, takes up a large number of system resources and needs to simulate the attack. This paper presented a C/S, open vulnerability and assessment language(OVAL) based android vulnerability detection and assessment system. This architecture puts most of the evaluation work to the central control and reduces the impact on the android system performance. Using OVAL as vulnerability assessment standard, the architecture guarantees the high accuracy of the evaluation, and it also has better openness and scalability.

Keywords Vulnerability detection, OVAL, Android

1 引言

随着网络的发展与普及,越来越多的人开始使用安卓智能手机。随之而来的便是漏洞和病毒所造成的手机网络安全问题[1-2]。通过漏洞检测技术及时发现漏洞并利用补丁程序进行修复是实现网络安全的重要技术之一[3-4]。

目前,主流的安卓漏洞评估系统大多采用主动探测技术,利用各厂商分别掌握的漏洞数据库查找并分析信息系统存在的安全问题。由于各厂商往往采取不同的数据信息格式,没有一种规范的数据表示标准,因此在安全产品相互兼容和互操作性上存在很多问题,这也造成了误报率高的问题;而且,主动探测型漏洞评估系统采用全局的模拟攻击测试,扫描时间长,影响手机的正常工作,还需要开发大量的模拟攻击代码。

针对目前漏洞评估系统存在误报率高、扫描时间长且需要开发攻击代码的缺点,本文提出了一种基于OVAL[5]的新型安卓漏洞检测评估系统,该系统由控制台、数据中心和手机代理3大部分组成。这3个模块协同实现漏洞检测评估,与现有安卓漏洞评估系统相比,具有精度高、对目标系统的性能影响小、评估时间短和可扩展性强的优点,而且免除了传统漏洞评估系统所需的攻击代码开发工作。

2 相关工作

目前国内外对漏洞检测评估系统的研究主要分为两个方向:对漏洞检测与评估的标准化研究以及对漏洞检测与评估技术的应用开发和产品化研究。

2.1 漏洞检测与评估标准化

由美国US-CERT(美国计算机紧急情况反应小组)针对漏洞评估过程提出的开放漏洞评估语言(Open Vulnerability and Assessment Language,OVAL)是对计算机系统漏洞进行评估的新标准。

OVAL用于把安全工具和服务运行中所有范围内的传输信息标准化,从而为安全产品之间的交互提供条件。每个OVAL定义都具有对应的CVE[6]编号,通过CVE编号,可以方便地查找出每一个OVAL定义的漏洞对应的补丁情况。

OVAL社区开发了3种使用扩展标记语言(eXtensible Markup Language,XML)编写的模式。OVAL系统特征模式用于表示系统信息;OVAL定义模式用于表达一个指定机器状态;OVAL结果模式用于报告评估结果。

到稿日期:2015-11-30 返修日期:2016-02-27

万 燕(1970—),女,博士,教授,主要研究领域为图像分析、3D数据处理,E-mail:winniewan@dhu.edu.cn;赵 希(1990—),男,硕士生,主要研究领域为渗透测试、移动应用;王国林(1991—),男,硕士生,主要研究领域为渗透测试。

2.2 漏洞检测应用开发和产品化研究

市场上涌现的漏洞评估产品大多为主动攻击模拟式，著名的产品有 ISS 公司的漏洞评估产品 ISS Internet Scanner[7]、Symantec 公司基于网络的漏洞和风险评估工具 NetRecon、启明星辰公司管理漏洞和攻击方法的漏洞信息管理系统天镜漏洞信息资源库。这些产品大大节省了渗透测试所需的时间，但是它们使用大量时间并占用大量系统资源对被检测系统进行模拟攻击。

3 基于 OVAL 的安卓漏洞检测模型

基于 OVAL 的安卓漏洞检测系统主要由控制台、数据中心和收集代理 3 个部分组成，其系统结构如图 1 所示。

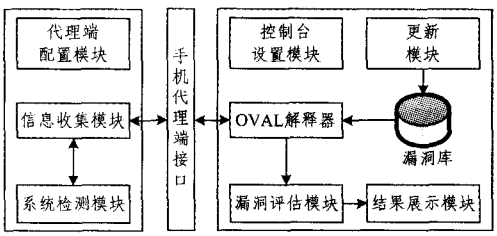

图 1 基于 OVAL 的安卓漏洞检测模型

各个模块的功能如下：

1)手机代理端接口：通过手机代理端接口，控制台中的 OVAL解释器把需要获取的信息发送给代理端，代理端中的信息收集模块把收集到的手机特征信息发送给控制台。

2)代理端配置模块和控制台配置模块：对代理端和控制台的配置进行管理，如端口和 IP 地址等。

3)更新模块：更新用户设置的更新时间和更新间隔，并对漏洞库进行更新，从 Internet 获取 OVAL 定义文档，更新并存储服务器 OVAL 定义文件。

4)信息收集模块：根据从控制台收到的消息收集手机的系统特征信息。

5)系统检测模块：收集安卓操作系统的基本信息和手机模块的基本信息。

6)漏洞评估模块：利用信息收集模块输入的特征数据和漏洞库提供的漏洞信息，判断系统当前状态是否安全，将存在的漏洞存储到文件中。

7)OVAL 解释器：根据 OVAL 定义文档相关的规范，将 OVAL 定义分析解释成代理端信息收集模块需要收集的手机特征信息。

8)结果展示模块：系统的检测结果由该模块呈现给用户；将评估结果存档并准备共享。

3.1 评估原理

本文提出的安卓漏洞检测评估系统需要利用新标准 OVAL在安卓手机上进行漏洞的相关检测。安卓系统的系统资源有限的，更有效地利用 CPU、内存以及最小化资源消耗是本系统的一个基本需求。形式化的漏洞评估原理如下：

$$M=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}, a_{i,j}\in\{0,1\}$$

矩阵 M 表示系统所需检测漏洞的描述。矩阵的一行表示一个漏洞，每个漏洞由多个系统特征 $a_{i,j}$ 组成。

漏洞库中多个漏洞间必定存在相同的系统特征。

$$h(M)=(\sum_{j=1}^{n}a_{1,j},\sum_{j=2}^{n}a_{2,j},\cdots,\sum_{j=1}^{n}a_{m,j})^{\mathrm{T}}$$

矩阵每行的和表示一个漏洞包含的系统特征数目。

$s=(s_1,s_2,\cdots,s_n),s_i\in\{0,1\}$

被检测的系统可以用矩阵 s 表示，s_i 表示系统的特征信息。

$$\omega=h(M)-[M*s^{\mathrm{T}}] \tag{1}$$

系统检测结果矩阵 ω 可以用上述式(1)计算得出，完整的公式如下所示：

$$\omega=\begin{pmatrix} \sum_{j=1}^{n}a_{1,j} \\ \sum_{j=1}^{n}a_{2,j} \\ \vdots \\ \sum_{j=1}^{n}a_{m,j} \end{pmatrix}-\left[\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}\times\begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}\right]$$

3.2 关键实现技术

3.2.1 OVAL 解释器

本文提出的安卓漏洞检测评估系统采用 C/S 的体系结构。OVAL 解释器在整个系统中充当控制台和代理端通信的中枢，将漏洞库中更新的漏洞定义文件解释成安卓代理端需要检测的系统特征元数据。

OVAL 漏洞定义文件是包含多个符合 OVAL 标准的 XML 文件，使用树形嵌套定义。定义文档有一个根节点 oval-definitions，根节点下有 6 个子节点：generator，definitions，tests，objects，states 和 variables[8]。其中 tests 和 objects 是漏洞定义文件最重要的两个子节点，tests 元素定义了此定义文件中所有的漏洞所要做的检测，objects 元素定义了所有漏洞存在时系统拥有的对象，也是检测的对象，所有的 objects 都在 tests 中被引用。

根据 OVAL 漏洞检测的基本原理，OVAL 解释器对漏洞定义文件进行解析的过程如图 2 所示。

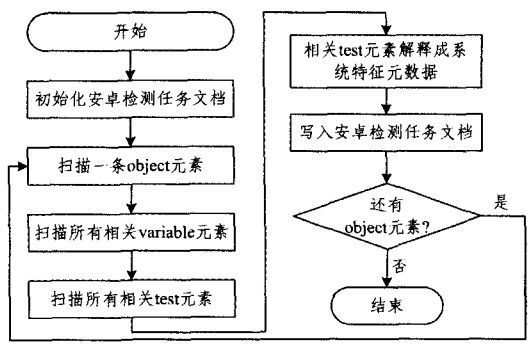

图 2 OVAL 解释器解析过程

图 2 中扫描的 object，variable，test 元素均来自漏洞库中更新的漏洞定义文件。

3.2.2 漏洞评估

手机代理执行系统检测和信息收集任务，获得了手机代理的系统特征信息。将系统特征信息和漏洞定义文件相结合，利用式(1)计算得到漏洞分析结果。计算过程如图 3 所示。

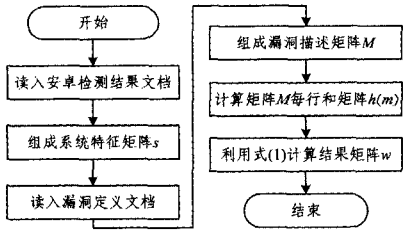

图 3　漏洞评估过程

4　实验与结果

实验主要考查精度和性能两方面,以客观反映系统对漏洞的识别率和性能。

测试环境:将控制台安装在一台装有 Windows 操作系统的台式电脑上。将代理端安装在一台中兴 N798(电信版)安卓手机上,此手机 CPU 为双核 1.0GHz,RAM 为 512MB,操作系统为 Android OS 4.0。

4.1　评估精度测试

在上述测试环境中进行测试。由于安卓系统版本较高,安全性增加,系统共检测到 3 个漏洞,如表 1 所列。

表 1　漏洞检测系统检测到的 3 个漏洞信息

CVE ID	漏洞描述
CVE-2012-2808	Android 4.0.4 DNS Poisoning
CVE-2013-6770	Android 4.3 Superuser Root Privilege Escalation
CVE-2013-6774	Android 4.2.x SuperuserUnsanitized Environment

通过 CVE 数据库得知安卓官网已提供对应漏洞的修补方法。将安卓手机系统升级后,重新用此系统对手机进行漏洞检测评估,3 个漏洞的值均由真转为假。

4.2　评估性能测试

为了测试原型的可伸缩性和内存调用。实验从 5 个 OVAL定义开始,每次增加 5 个 OVAL 定义,直到 OVAL 定义增加至 100 个。在每次实验中查看安卓手机的 CPU 利用率和内存占用情况,CPU 利用率和内存占用情况分别如图 4 和图 5 所示。

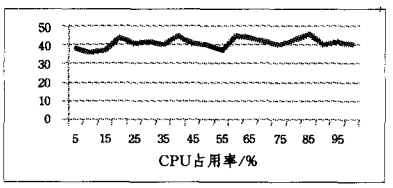

图 4　CPU 占用率统计结果

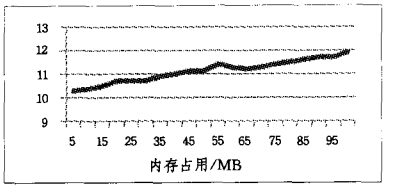

图 5　内存占用统计结果

结束语　本文提出一种基于 OVAL 的安卓漏洞检测评估系统,实验测试结果充分表明该系统具有以下特点:漏洞检测评估精度高;基于 OVAL,评估结果精度高并且能与其他安全产品共享利用;采用 C/S 的架构降低了对安卓手机性能的影响。

参 考 文 献

[1] ENCK W,ONGTANG M,MCDANIEL P. Understanding Android Security[J]. IEEE Security & Privacy Magazine,2009,7(1):50-57.

[2] SHABTAI A,FLEDEL Y,KANONOV U,et al. Google Android:A Comprehensive Security Assessment[J]. IEEE Security & Privacy,2010,8(2):35-44.

[3] BARTEL A,KLEIN J,TRAON Y L,et al. Automatically securing permission-based software by reducing the attack surface:an application to Android[C]// Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. ACM,2012:274-277.

[4] HANNA S,HUANG L,WU E,et al. Juxtapp:A Scalable System for Detecting Code Reuse among Android Applications[M]// Detection of Intrusions and Malware,and Vulnerability Assessment. Springer Berlin Heidelberg,2013:62-81.

[5] The MITRE Corporation. OVAL[EB/OL]. (2015-07-09)[2015-11-15]. http://oval.mitre.org.

[6] The MITRE Corporation. CVE[EB/OL]. (2015-07-24)[2015-11-15]. http://cve.mitre.org.

[7] Internet Security SystemsTM. Vulnerability assessment[EB/OL]. (2015-07-26)[2015-11-15]. http://www.iss.net/find_products/vulnerability-assessment.php.

[8] WANG X D,GAO L,ZHANG L. Design and implementation of OVAL-compatible VAS on multi-platform[J]. Computer Engineering and Applications,2009,45(36):82-85. (in Chinese)
王旭冬,高岭,张林. 兼容 OVAL 的多平台 VAS 设计与实现[J]. 计算机工程与应用,2009,45(36):82-85.

(上接第 59 页)

[8] BYRKA ,JAROSŁAW,GRANDONI F,et al. An Improved LP-based Approximation for Steiner Tree[C]// Proceedings of the Forty-second ACM Symposium on Theory of Computing. ACM,2010:583-592.

[9] ZHAO W,ZHANG L,LIU Y,et al. SNIAFL:Towards a Static Non-Interactive Approach to Feature Location[C]// International Conference on Software Enginee-ring. IEEE Computer Society,2004:293-303.

[10] EISENBARTH T,KOSCHKE R,SIMON D. Locating Features in Source Code[J]. IEEE Transactions on Software Engineering,2003,29(3):210-224.

[11] FU K,QIAN W Y,PENG X,et al. Feature Location Method Based on Call Chain Analysis[J]. Computer Science,2014,41(11):36-39. (in Chinese)
付焜,钱文亿,彭鑫,等. 一种基于调用链分析的特征定位方法[J]. 计算机科学,2014,41(11):36-39.

[12] BYRKA,JAROSŁAW,GRANDONI F,et al. An Improved LP-based Approximation for Steiner Tree[C]// Proceedings of the Forty-second ACM Symposium on Theory of Computing. ACM,2010:583-592.