基于 OVAL 的漏洞管理与发现系统

夏映晖

摘要

漏洞管理系统需要具有与网络漏洞库实时同步、于本地系统高效扫描等特性。本文以一种开放漏洞描述语言 OVAL 为基础，构建了能满足高效同步、扫描的开放漏洞管理系统，能够在任何 CVE 条目更新时发现本地漏洞，并使用 XBase 作为 XML 数据库，从而对半结构化的漏洞数据进行高效管理。并且本系统以 OVAL 为基础实现了漏洞检查，比 OVAL 本身实现的系统检查时间减少 50%。另外，以该系统为核心，本文又进行了漏洞发生模式的预测，可为信息安全管理提供安全策略优先级建议。

关键词：漏洞管理；OVAL；CVE；漏洞预测

abstract

TBD.
Key Words: TBD.

目录

第 1 章　绪论

1.1　研究背景和意义

漏洞发现的方法研究有很多，如，静态分析测试、模糊测试、基于属性的安全测试、形式化安全测试、基于渗透的安全测试等。这些方法利用了不同的安全理论和技术，从不同的角度对软件安全性测试做出了探讨，有较多差异性。

从理论的形成思想可以将这些方法分为基于系统性考虑的方法以及基于有效性考虑的方法，前者重于通过模型或知识库充分地遍历软件状态，扫描可知的安全性漏洞，但无法避免效率低、误报率高的缺点；后者重于有效地发掘软件漏洞，常常可以发现未知的安全性漏洞，但局限性较大、漏报率高、不够全面和系统。

近年来，由于软件规模的增大，设计模式的多样化，测试人员很难从架构和设计中找到更多安全性漏洞，系统性的方法效率不高，因此以模糊测试为代表的完全自动化方法逐渐流行起来，成为了攻防两端都会关注的方法。此外，由于已发现的漏洞规模增大，以此为依据构建的知识库也成为了发现新漏洞的有效工具。

1.2　国内外研究概况

漏洞管理也是软件安全的重要课题。有多种漏洞形式化方法被提出，从语义网发展的本体方法是其中一种，Ju An Wang 等人建立了一个基于本体技术的漏洞管理系统，能够从漏洞数据库中获取信息安全概念, 并可推理漏洞可能产生的影响。对于自动化漏洞管理上，Martín Barrère 等人尝试利用 OVAL 增强 Android 实时安全性，实现了一个轻量级自动化漏洞评估系统；进一步地，他们针对漏洞生命周期提出了一个基于 OVAL 的预测模型，用于检测遗留系统的漏洞。XinMing Ou 等人提出了 MulVAL, 旨在通过 OVAL 自动发现大规模网络中的漏洞，并可以与入侵检测系统形成互补。

由于大规模数据分析近年来如火如荼，基于漏洞库的知识发现也成为学者们关注的焦点。Omar H. Alhazmi 等人探讨了漏洞的量化特征的提取与漏洞预测的可能性，并在商业和开源系统中均通过逻辑或线性回归模型获得了收敛的结果。他们在另一篇文章中比较了 Anderson 等人提出的漏洞发现模型（VDMs），以及该模型之前 Browne 等人提出的漏洞发掘模型（VEMs），还有 Alhazmi 等人提出的 Alhazmi-Malaiya Logistic Model（AML）等分布模型，并通过对现实数据的检测证明 AML 模型在漏洞记录较多的库中拥有较好性能。在 VDMs 之上，也有其他的研究，如 Jinyoo Kim 等人在多版本开源软件系统中挖掘漏洞，考虑了不同版本在同一软件上漏洞发现的影响，并用 Apache 和 Mysql 为例做了验证实验。Andy Ozement 对 VDMs 等基于概率的模型提出了质疑，他认为 VDMs 在软件安全评估上拥有一定的好处，但即使拥有大量漏洞发

现特征，该类模型在时间消耗、系统环境适配、漏洞发现独立性和静态代码依赖上都有一定的缺陷，因此 VDMs 仅能作为漏洞挖掘辅助系统。

除了 VDM 相关研究外，机器学习在漏洞挖掘上也有其他维度的探讨，Mehran Bozorgi 等人研究了系统管理中的漏洞防护优先问题，利用机器学习中的 SVM 分类器，可以预测漏洞是否发生、何时发生，与启发式专家知识系统相比拥有更好的准确性和补充性。同样应用了机器学习方法的辅助性自动软件漏洞检测有 Fabian Yamaguchi 等人提出的在代码级检测 API 使用模式的研究，也在真实环境下捕获了未知漏洞。

1.3 本文的主要工作

本文目标是基于 OVAL 描述语言构建漏洞管理及检测系统，在多种系统及软件版本下，分析系统环境条件自动检测已知漏洞，通过对知识库中大量 CVE 条目进行数据挖掘，可对未知的系统环境进行安全评估与漏洞预测。

首先，本系统通过实时获取 OVAL 提供的 CVE 条目构建软件漏洞管理系统，利用协议信息得到漏洞检查语义，匹配系统环境及各软件的依赖逻辑关系，从而自动化发现系统中存在的已知漏洞。

其次，通过对漏洞库的分析，抽取可用于分类未知软件漏洞的特征，使用机器学习的方法构造分类器来判断软件特定的模块可能产生的漏洞，并根据知识库得到该漏洞触发条件。并得到漏洞发生的优先序列，可用于系统的精益安全管理。

1.4 本文的组织结构

论文的结构和各章内容如下:

第二章: 探讨了漏洞管理相关理论与技术，对漏洞分类、基于 OVAL 的漏洞库设计以及 CVE 的管理模式进行了分析。

第三章: 研究了漏洞发生模式的挖掘技术，分别从先验角度和时间序列角度对漏洞发生模式进行讨论。

第四章: 阐述了本文中漏洞管理系统的设计与实现，并集成了漏洞挖掘及基于 OVAL 的漏洞库。

第五章: 阐述了本文中漏洞管理系统的测试工作。

第六章: 总结全文，分析系统中存在的问题，并探讨了未来工作。

第 2 章　漏洞管理相关理论与技术

2.1　漏洞分类

漏洞可从成因、严重程度等角度进行分类，探讨目前存在的分类法

2.2　基于 OVAL 的漏洞库设计

OVAL 包含了三种模式：定义模式、系统特征模式以及结果模式。

表 2.1: OVAL 三种模式介绍

定义模式	系统特征模式	结果模式
对漏洞进行定义 对漏洞进行定义 对漏洞进行定义 对漏洞进行定义 对漏洞进行定义 对漏洞进行定义	对漏洞检测行为进行描述	对漏洞影响进行描述

2.2.1　CVE 的漏洞管理模式

CVE 提供了公共系统中漏洞的命名规则，并进行编号及描述，是安全漏洞挖掘中重要的参考依据。

第 3 章　漏洞发生模式的挖掘

3.1　条件先验知识下的漏洞发生模式挖掘

3.1.1　算法描述

3.1.2　实现

3.2　时间序列上的漏洞发生模式挖掘

利用基于时间序列的数据挖掘，我们可以在漏洞库上找到规律性的趋势、模式、突变。时间序列模式可以使用 T(Trends)、C(Circle)、S(Season)、I(Irregular) 来分别表示趋势、周期、季节、无规律四种基本模式。

时间序列的变量 Y 可表示为：

$Y = T \times C \times S \times I$

或：

$Y = T+C+S+I$

3.2.1　算法描述

3.2.2　实现

第 4 章　漏洞管理系统的设计与实现

4.1　需求分析

4.1.1　功能需求

本系统包括以下几个主要功能模块：

（1）漏洞展示模块：用于对漏洞的可视化展示，包括主页的可视化图及漏洞发生的日历

（2）漏洞同步模块：实时同步并显示发生的漏洞

（3）漏洞预测模块：从现有漏洞库中挖掘发生模式

（4）漏洞管理模块：提供详尽的视图对漏洞进行增删改查

（5）漏洞扫描模块：通过 OVAL 语言对本地系统进行检查

（6）系统日志模块：记录本系统的任何行为

（7）系统设置模块：设置本系统的各项参数配置

4.1.2　非功能需求

本系统需要实现以下几个主要非功能需求：

（1）时效性：漏洞发生具有时效性，因此本系统也需要对新漏洞及时响应

（2）兼容性：漏洞会在不同系统中发现，故系统需要满足跨平台特性

（3）效率：系统拥有 1 秒内的漏洞查询性能、1 分钟以下的漏洞检查性能

（4）易用性：用户友好的界面、易学的操作

（5）环境：运行于本地，可于本地对漏洞进行查询与检查

4.1.3　用例图

用户用例包括查看漏洞信息、查看漏洞预测、进行漏洞扫描、查看系统日志、设置管理系统等五个主要用例。在查看漏洞信息中，包括了查看漏洞趋势和查看漏洞统计两个子用例，并扩展了漏洞实时同步这个子用例，见图 4.1。

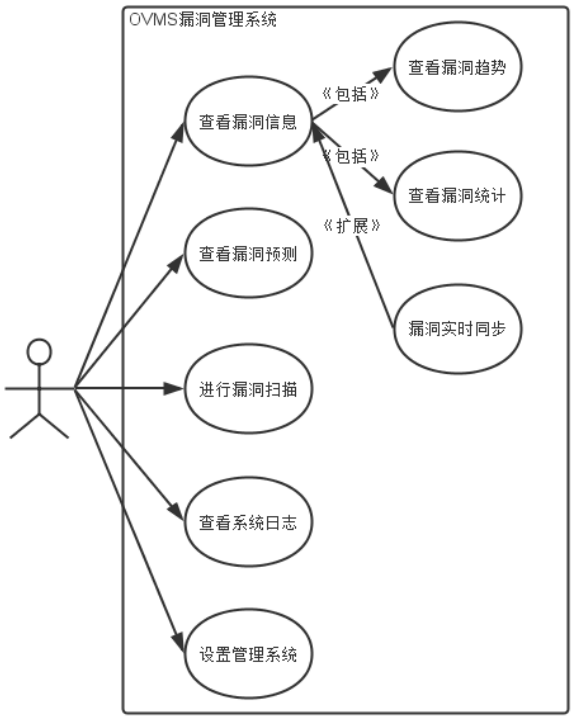

图 4.1: 用例图

4.2 系统设计

4.2.1 系统架构图

4.2.2 功能设计

4.2.3 包图

4.2.4 类图

4.2.5 时序图

4.2.6 ER 图

4.2.7 界面设计

4.3 系统实现

4.3.1 漏洞扫描模块的编写

第 5 章　漏洞管理系统的测试

5.1　单元测试

5.1.1　具体模块单元测试编写

5.2　系统测试

5.2.1　功能测试

5.2.2　非功能测试

5.3　系统性能比较

5.4　时间序列算法评价

第 6 章　总结

对更多平台的适用性。与更多漏洞库进行同步。在漏洞发生模式预测上产生更好结果。