软件安全漏洞分类研究综述

丁羽，邹维，韦韬

（北京大学互联网安全技术北京市重点实验室，北京 100080）

摘要：信息系统安全漏洞是信息安全风险的主要根源之一，是网络攻防对抗中的重要目标。由于信息系统安全漏洞的危害性、多样性和广泛性，在当前网络空间（Cyber Space）的各种博弈行为中，漏洞作为一种战略资源而被各方所积极关注。如何有效发现、管理和应用漏洞相关信息，减少由于漏洞带来的对社会生活、 国家信息安全的负面影响，即对漏洞及相关信息的掌控已经成为世界各国在信息安全领域工作的共识和重点。

软件安全漏洞是信息系统安全漏洞的一个重要方面。学术界不仅在安全领域有相关研究，还在编译理论、形式化分析等领域对软件安全漏洞研究方面均有贡献。本文简要介绍学术界关于漏洞分类学（Taxonomy）和漏洞分类（Classification）的调研。文章最后总结了漏洞分类的几个关键问题，并对未来的漏洞分类工作进行了展望。

关键词：软件漏洞, 分类学, 漏洞模型, 漏洞分析

中图分类号：TP393.2

A Summary of Software Classification/Taxonomy Techniques

DING Yu, ZOU Wei, WEI Tao

（Beijing Key Laboratory of Internet Security Technology, Peking University,Beijing100080,China）

ABSTRACT: Information system vulnerability is a root cause of information system risk and plays an important role in cyber-attack and defense. Due to the harmfulness, diversity and generality of information system vulnerability, information system vulnerability is a highly concerned strategic resource. The way of finding, applying and managing vulnerability information and eliminating the negative impact caused by software vulnerabilities is becoming a key work in information security research of all the countries in the world.

Software vulnerability is an import part of information system vulnerability. In this paper, we summarize previous works on software vulnerability classification and taxonomy. At the end of this paper, we conclude the key questions in this research area and give a forward looking.

Key words: software vulnerability, taxonomy, vulnerability model, vulnerability analysis

引言

在中共中央办公厅、国务院办公厅发布的《2006-2020 年国家信息化发展战略》第四章“我国信息化发展的战略重点”中明确指出“积极跟踪、研究和掌握国际信息安全领域的先进理论、前沿技术和发展动态，抓紧开展对信息技术产品漏洞、后门的发现研究，掌握核心安全技术，提高关键设备装备能力，促进我国信息安全技术和产业的自主发展。”

信息系统安全漏洞是信息安全风险的主要根源之一，是网络攻防对抗中的重要目标，信息系统安全漏洞的发现与利用在信息安全中处于十分重要的地位。由于信息系统安全漏洞的危害性、多样性和广泛性，在当前网络空间（CyberSpace）的各种博弈行为中，漏洞作为一种战略资源而被各方所积极关注。如何有效发现、管理和应用漏洞相关信息，减少由于漏洞带来的对社会生活、国家信息安全的负面影响，即对漏洞及相关信息的掌控已经成为世界各国在信息安全领域工作的共识和重点。

软件安全漏洞是信息系统安全漏洞的一个重要方面。分析、理解软件安全漏洞对于了解当下的安全威胁非常关键。对软件安全漏洞进行分类是搜集威胁信息、掌握安全威胁发展趋势的基础。更全面、精细的软件安全漏洞分类可以把安全事件、漏洞利用、软件平台等多方面组件在安全的视角下关联起来，从而帮助安全专家、分析人员等有效的进行分析，找到相应的解决方案。安全漏洞和漏洞利用的分析和分类对于安全模型的发展以及威胁预测有很关键的作用。）研究和漏洞分类研究十分相关。漏洞建模研究可以将漏洞的各方面特性进行量化。这种量化从一方面来看可以用于漏洞分类。同时对于漏洞分类的需求、研究也可以对漏洞建模有帮助。漏洞的分类是客观存在的，但不是一成不变的，而是根据需求所变化的。

收稿日期： 2012-6-15

作者简介：丁羽（1988－），男（汉族），北京市，北京大学 2010 级在读博士生。

通信作者：邹维，研究员，E-mail：zou_wei@pku.edu.cn

1 软件安全漏洞分类概述

漏洞分类是漏洞研究的基础，漏洞分类的标准即是将漏洞进行分类的根据，又是软件漏洞的关键点。分类有两个概念：归类（Classification）和分类学（Taxonomy）。Classification 侧重于，对于一个给定的元素，找到最适合他的那个类别。而 Taxonomy 则侧重于系统的对所有元素进行不重、不漏的分配一个类别标识。Taxonomy 侧重于系统性的整体研究，不漏是分类学的基本要求。而 Classification 侧重于对一个元素集合进行归类的操作，对于不漏的性质则没有太多要求。好的漏洞分类理论可以将漏洞不重、不漏的根据明了的规则进行分类，同时对漏洞相关研究起指导作用。现有的漏洞分类理论大体从以下五个视角对漏洞进行描述，并选取其中一个或一些作为分类标准：

关注对象 部分软件漏洞分类体系结构首先对软件类型进行分类。例如：Web 服务程序漏洞、Web 客户端程序漏洞、操作系统漏洞、应用软件漏洞等。

攻击效果 攻击效果指的是攻击者可以利用漏洞进行的攻击类型，例如指针篡改、敏感信息泄露、拒绝服务攻击（Denial of Service，DoS）、权限提升等。

受害组件 受害组件指的是软件漏洞所在的软件模块，例如输入解析模块、登陆模块、数据模块等。

涉及标准 软件漏洞的涉及标准指的是软件漏洞所涉及的行业标准、协议等，例如网络协议 FTP、HTTP，文件格式 DOC、JPG、编程语言 Python、Java 等。

触发条件触发条件一般包括数据长短超标（过长、过短）、数据范围异常、操作顺序异常等。触发条件是诸多软件安全漏洞分类视角中最注重细节的一个。许多工作就从微观角度衡量软件安全漏洞的触发条件并创造了各具特色的描述方式来进行漏洞刻画和分类。

此外，漏洞的描述方法也是软件安全漏洞分类的重要部分。对安全漏洞有效、简练的描述可以对漏洞检索、归类都有重要帮助。自从 1976 年起就有在漏洞描述方面的学术研究[1]。现在的漏洞数据库系统[29-46]对漏洞都有各自的描述方法，并且各有特色。

现有的软件安全漏洞分类技术的分类标准基本均属于上述五种分类视角。此外，现有的漏洞分类研究可以分成两大类：归类系统（Classification）和分类学（Taxonomy）。归类系统侧重于将每一个软件安全漏洞分配一个类别标识；分类学研究侧重于系统的将软件安全漏洞进行整体分类，探索他们之间的关系。本文依次介绍软件漏洞归类系统和漏洞分类学研究。

2 软件安全漏洞归类（Classification）

南加州大学信息科学学院在 1978 年发表了一份报告[3]。该报告出自于美国国防部高级研究计划署信息处理技术办公室对操作系统安全性、漏洞与自动防御技术的一次研究，并构造了一个漏洞分类系统。该分类系统可以搜集错误（raw error），对这些错误进行标准化表示，之后精简这些标准化表示，再对搜集到的这些精简过的错误的标准化表示进行“正规化”（Normalization）和分类。在整个过程中，对操作系统的特征抽取用于帮助进行精简和分类。在这篇文章中讨论了最理想的错误描述应该是“触发错误的条件”这一概念，并且指出使用这个条件在当时的计算能力下是不可行的。作为简化，该工作使用了触发错误的一个值以及控制这个值的直接条件作为漏洞特征。该工作抽取操作系统提供的系统调用，以及其参数表作为漏洞刻画的辅助特征，并且找出所有直接给参数赋值的语句，并且从中识别出与控制流相关的指令。在最后的漏洞分类时，将收集到的带参数的漏洞信息与从操作系统中抽取出来的信息作比较，记录每个控制流的相关跳转，依据这些跳转情况对漏洞进行分类。该工作在当时流行的操作系统（GCOS，MULTICS，UNIX）上搜集到了 100 多个错误并进行了分类，但是这些数据没有公开。南加州大学的工作体现了“数据流分析”的思想，讨论了回溯数据流这一方法的可行性，这一方法现今已成为污点分析（Taint based analysis）[4,5,6]的核心思想。

隶属于美国能源部下国家核安全局的劳伦斯利弗莫尔国家实验室在 1988 年发表了一篇文章[7]，里面对安全操作系统的设计准则进行了讨论，然后对操作系统安全漏洞进行了分类。这篇文章把操作系统安全漏洞分成了 9 类：访问管理的错误实现、最小权限原则的违反、完全仲裁原则的违反、经济原则的违反、最小公共机制的违反、用户可访问原则的违反、整体安全机制的缺失编码失误以及安全责任的移位。这篇文章从操作系统设计的原则出发，对设计操作系统时可能出现的漏洞进行分类。这个分类对操作系统设计、实现人员具有直接的指导作用。

普渡大学的 Wenliang Du 在 1997 年的国家信息系统安全会议发表了一篇对软件错误（Software Errors）进行分类的文章[8]。该文章主要关注与引起安全隐患的软件错误的分类。首先该文章从软件工程的角度对一个安全类型的错误（Security error）的生命周期进行了讨论，得出结论：一个软件错误被发现之后首先会找到其诱因（cause），然后研究其会导致的安全问题（impact），最后得到修补（fix）。这篇文章从诱因、导致的安全问题以及修补三个视角对软件错误进行分类。从诱因可以将软件安全错误分为：验证错误、认证错误、串行化错

误、边界检查错误、域错误、不正确或者弱的设计错误以及其他可被利用的逻辑错误。从对安全的影响（Impact）可以分为：任意代码执行、目标资源篡改、读取目标资源以及拒绝服务。从软件错误修补可以将软件错误分为四类：伪造实体、缺失实体、错位实体以及不正确实体。

普渡大学的 Ivan Victor Krsul 的博士毕业论文以编程者在软件开发过程中围绕的基本假设作为分类视角进行分类。Krsul 使用了威胁特征、环境假设特征、漏洞自然特征共同作为漏洞分类的依据。其中威胁特征包括行为特征：漏洞利用可以导致的(1)越权对象访问、(2)越权对象删除、(3)越权对象修改以及(4)越权对象创建，和结果特征：(1)操作系统可用性发生变化、违反安全策略的信息泄露、信息的错误表达、信息的丢失、系统完整性的变化以及系统可信度的丢失。上述特性均被量化为“是”、“否”和“不适用”的三值属性。同样，环境假设特征也被量化为这样的三值特征，包括许多程序员在编程时的假设条件。漏洞特征也被三值化，包括：受影响对象、对象收到的影响、使用到的机制以及接受的输入来源。每一项包括许多可选项，这些可选项是三值化的。文章使用了大量漏洞数据对这种分类方法进行了验证，验证结果证明了这种分类方法是很可行的。

比利时鲁汶大学的 Frank Piessens 在 2001 年的 IICIS 会议上发表了一篇软件安全漏洞分类综述文章[9]。该文章观点独特，作者认为：软件漏洞绝大多数都是符合一定模式的。从此出发作者把软件安全漏洞按照产生漏洞的模式分成了以下几类：缺少防御性输入检查、在危险环境下的软件重用、牺牲安全性来换取功能和方便、依靠不安全的抽象、不安全的默认配置和困难的配置、意料之外的服务滥用和功能交互、非原子性的检查和使用、编程失误。该文章对软件开发人员提出了建议和忠告，给出了一些如上安全漏洞的例子，并提出了如何避免产生这些漏洞的方法。

挪威基约维克高等学院（Gjovik University College）的 Hanno Langweg 和 Einar Snekkenes 在 2004 年的 IEEE Security and Privacy 期刊上发表文章[10]，总结了前人的分类方法，然后对恶意软件攻击进行了分类学研究。这篇文章从三个角度来进行分类：产生 fault 的位置、原因和漏洞带来的影响。位置从接受输入的方式分为三类：交互式输入（用户提供的输入）、主动输入（如 API 调用）和被动输入（如回调函数）。产生 fault 的原因和 Wenliang Du 的工作[8]大同小异。

Spire Security 实验室的 Pete Lindstrom 在 2004 年时候发布了一份技术报告，专门对 Web 服务的漏洞进行分类，给出了一份 Web 服务漏洞 Top10 的参考排名。Web 服务有几个特点：标准众多、这篇文章把 Web 服务漏洞分成如下几类：XML/SOAP 操作、漏洞滥用、不可信配置、XML 处理、不可信实体以及遗留部件。由于 Web 服务的复杂性和松耦合性，各个组件的安全问题叠加在一起就容易产生更明显的安全漏洞。接下来这篇文章给出了最常见的 10 种 Web 服务安全漏洞类型，他们是：强制解析、参数污染、递归载荷、超长载荷、模式污染、网页服务描述语言扫描、路由绕行、外部实体攻击、SQL 注入、重放攻击。这十类攻击基本囊括了所有常用的 Web 攻击技术。经过近几年的发展，可能排名有所变化，但是这十类攻击依然是最常见的攻击 Web 服务的方法。

哥伦比亚大学的 Herberth. Thompson 在其发表的专著[11]中将软件漏洞分成了五大类、19 个小类，并且提供了一系列例子和信息来讲述如何发现这样的漏洞和如何修补这样的漏洞。其中五大类包括：系统级漏洞、数据解析漏洞、通信级漏洞、网站级漏洞以及信息泄露漏洞。这本专著主要侧重于讲述典型的漏洞以及其成因，帮助软件开发者从 Security 的视角重新审视自己开发的代码，增强安全性。同样类型的专著还有微软公司的 Michael Howard 撰写的[12]。

2005 年卡耐基梅龙大学的 Robert Seacord 发表了一篇技术报告[13]，讨论了一项叫做“结构化安全漏洞分类”的技术。该分类技术与之前所有的技术不同的是：采取了“属性－值”对（attribute-value pair）的漏洞信息刻画方式进行分类。这种分类系统的好处是：相对于传统的“是”或“否”的漏洞刻画方式，这种漏洞刻画方式可以更方便的进行自动化分类。这篇文章使用 The Resource Description Framework（RDF）作为表示方式进行信息表达和分类。实验证明这种表达方式可以更有效的进行漏洞分类[13]。

加州大学戴维斯分校的 Sophie Engle 等在 2006 年时提出了一种基于树状数据结构的漏洞分类方法[14]。该分类方法的使用场景是：每个具体的漏洞由一组属性来描述，这一组属性是按顺序排列的，并且位于同样位置的属性是同一级别的。例如在分类协议漏洞时，根节点是协议漏洞，协议漏洞按照漏洞成因可以分成配置错误、设计问题以及实现问题。配置错误又可以分成：软件配置错误、硬件配置错误。在分类一个具体的软件漏洞时，只需从根节点开始向下进行依次匹配，就可以找到具体的漏洞分类。

3 软件安全漏洞分类学（Taxonomy）

学术界第一个关注软件安全漏洞分类的是 1976 年美国国家标准局所主持的安全操作系统研究项目（Research In Secured Operating Systems，RISOS）[1]。在项目报告中作者 Konigsford 对操作

系统完整性缺陷（Integrity flaw）首次进行了完整的分类学研究。该文章使用形式化的方法对系统完整性缺陷分类标准进行了定义，在该方法下每一个操作系统安全漏洞都可以使用以下语法进行描述：

一个**用户群**通过一系列**攻击方法**实施的**漏洞利用**触发了一类客体的**完整性瑕疵**，并导致对一类**资源**的非授权访问。

其中用户群包括：程序、服务以及入侵者。攻击方法包括：侦听、扫描、先占以及占取。漏洞利用包括三大类（拒绝使用/占有、拒绝排他使用/占有和修改）以及9小类。客体包括：物理保护、用户资料、计算机硬件、应用程序、操作系统。资源包括：信息、服务、设备。RISOS 的分类系统通过如上定义进行分类，分类的主要依据属于攻击效果视角，同时按照受害组件和触发条件进行辅助说明。该文章还特别针对计算机操作系统的完整性瑕疵进行详细讨论和分类。操作系统的完整性瑕疵被分为了7类：不完备的指针验证、不一致的参数验证、隐式的机密数据共享、不同步验证/不恰当的串行化、不充分的身份鉴别/授权/鉴定、可被违反的禁令/限制、可被利用的逻辑错误。接下来 RISOS 项目对当时的几个操作系统进行了安全性分析，并将各自的系统安全措施进行比较。

美国海军研究实验室（NRL）在1993年发表了一篇论文[15]，阐述了他们在计算机操作系统方面的漏洞分类方法。该文章提出了三种对操作系统漏洞做分类的视角：按起源、按引入时间，和按位置。这篇文章用以下三个问题作为动机来展开针对以上三个视角的讨论：

如何进入的系统？

何时进入的系统？

系统中的哪个位置可以证实？

NRL 对在按时间进行分类的视角中将漏洞分成三大类：开发时、维护时、和操作时。按位置把漏洞分成两大类：软件漏洞和硬件漏洞。软件漏洞又分为：操作系统漏洞、程序漏洞和支持程序（Support）漏洞。这篇文章以分类学的观点对漏洞进行了三个视角的分类，体现了漏洞在攻击方法、诞生时间以及位置上的三个维度的性质。

普渡大学的 T.Aslam 针对 UNIX 操作系统的安全错误（Security Faults）进行了分类学研究[16,17]。这篇文章从软件工程和软件测试的角度看软件中从“错误”(error）到“故障”(fault）的形成过程，从而构造一个类似生物分类的“分类树”，如图2.3。Aslam 将软件安全故障分成操作故障、环境故障、以及编码故障三大类。操作故障包括：安装位置错误、参数设置错误、文件存储的访问越权错误；编码故障包括：同步错误（竞争条件错误、不正当串行化错误）、条件错误（条件缺失，条件不正确，条件谓词缺失）；环境故障包括操作环境限制导致错误、编译器或者操作系统故障导致的错误，其他模块交互错误以及异常处理机制导致的错误。Aslam 采用了实体－联系模型数据库来存储漏洞数据（区别于现今最常用的关系数据库）。T.Aslam 从软件源代码的角度发现潜在错误，再对其进行分类。发现错误主要通过软件代码静态分析、上下文无关的路径分析等手段进行。最后分析结果再套入到其分类学规则下进行分类。

瑞典查尔姆斯理工学院的 Ulf Lindqvist 和 Erland Jonsson1997年时在 IEEE Security and Privacy 期刊上发表了一篇讨论入侵行为分类的文章 [18]。 相比前述从操作系统等观点进行分类，从入侵行为角度进行分类的方法适用面更广，不再仅 仅适用于操作系统开发者，而对安全专业人员等都有帮助。这篇文章把入侵行为 分成了9类，他们是：外部误用、硬件误用、伪装、 冒充、设置后期误用、绕过管理控制、主动资源误用、被动资源误用、迟钝导致的误用、在其他类型的误用做间接帮助。在定义了9个入侵类型之后，作者将入侵结果也进行了分类。入侵结果被分成了三大类：暴露、拒绝服务，以及错误输出。然后将当时比较流行的入侵攻击技术在这两个维度（入侵方式、入侵结果）进行了分类，从分类结果可以看出，绝大多数入侵攻击技术只局限在少数几类组合中。说明入侵方式在一定程度上影响入侵结果，两者呈现一定相关性。

美国能源部的 Sandia 国家实验室在1998年的时候也从入侵角度进行了漏洞分类[19]。这篇文章的分类对象是安全问题（Security Incident），这个角度有点类似于 Wenliang Du 的工作。首先，这篇文章把一个 Security Incident 按照 Incident、Attack、Event 三个层次进行分解。其中 Event 类似于攻击效果，指的是一个安全问题最后产生的实际效果，由“操作”和“目标”组成，例如：扫描进程、读取数据等。“操作”包括攻击者比较常用的攻击操作，例如：扫描、淹没、绕过等；“目标”包括敏感的数据等，例如：账户、进程、数据等等。而一个攻击除了包括一个Event之外,还包括工具、漏洞和未授权的结果。一个安全事件，则包括一个或几个 attack，同时还包括攻击者和其目的。Sandia 实验室这项工作的主要目的是设计一个通用描述语言（Common Language）。这个通用描述语言的想法具有很强的前瞻性。先进安全研究者越来越多，每个人都有自己的一套描述体系，因此工具、协议等难以互通，数据也难以交叉验证。通过 Sandia 实验室的这项工作，可以很清楚的描述一个攻击事件，和其内部所有的关键因素，帮助人们高效刻画攻击事件以及进行分类学研究。西班牙的 Gonzalo Alvareza 和 Slobodan Petrovic 在2003年时，在 Howard 工作[19]基础上进行了深入研究，

将 Howard 的分类学理论应用到了 Web 安全中。为了应用 Howard 的分类理论,Gonzalo 将 Web 攻击周期分成了9段，他们是：入口、漏洞、服务、行动、长度、HTTP 元素、目标、场景、权限。

马里兰大学的 Kanta Jiwnani 2002年在国际软件维护大会上发表了一篇文章[20]讨论从软件测试视角进行的软件安全瑕疵分类。这篇文章以 Du 的工作[8]作为基础，同样从三个方面对漏洞进行描述：开发中的问题（Developing Issues）、漏洞位置（Location）、以及带来的影响（Impact）。相比 Du 的工作，Jiwnani 的分类学更细致，并且在实用时表现的更好。这篇文章使用了一个很大的样本集来实验分类效果，其中包括从 Harris 公司获取的1200个 WindowsNT 漏洞和 RedHat Linux 错误信息中的160个样本。 由于这个工作是从软件测试角度出发的，其分类过程和分类结果对于软件测试人员具有较大的帮助作用。

加州大学洛杉矶分校的 Algirdas Avizienis 等人在2004年时发表了一篇论文[21]讨论了关于软件故障（faults）的分类学。这篇文章从软件工程的角度进行讨论，首先定义了一个“可靠性与安全树”的概念，描述了可靠性与安全的三个侧面：属性（可用性、完整性、可维护性等）、威胁（错误、故障等）与方法（错误容忍、错误避免）。然后基于这三个侧面将软件错误分成开发错误、 物理错误以及交互错误三大类和十几个小类。 这篇文章中定义了软件安全漏洞为：“an internal fault that enables an external fault to harm the system”，即由一个可以引起外部故障，从而危害到系统的内部故障。

美国硅谷软件安全领军企业 Fortify Software 的安全专家 Katrina Tsipenyuk 等在2004年的 IEEE S&P 期刊上发表了一篇关于软件错误分类学的文章。这篇文章从软件开发的角度出发，列举了开发者在以下问题(Kingdoms)上常犯的错误：输入验证和表达、API 滥用、软件开发者并不熟悉软件的安全特性、时间和状态、并发控制问题、错误、错误处理函、代码质量、封装、环境。

IBM 的托马斯•沃森研究中心的专家 Sam Weber 等对于软件的瑕疵（Software Flaw）进行了明确定义，并且对软件瑕疵进行了分类学研究[22]。该文章从软件瑕疵是否是故意留下的（Intended）还是非故意的（Inadvertent）进行初级分类。故意留下的软件瑕疵包括：恶意的（逻辑、时间炸弹）和非恶意的（隐蔽信道、不一致的访问路径）。非故意留下的软件瑕疵包括：验证错误（定位错误、检查位置错误、不良的参数检查、认证/验证不足）、抽象错误（对象重用、内部信息暴露）、同步瑕疵（并发错误）、子部件重用/错误（信息泄露、责任理解错误）和功能瑕疵（异常处理错误、其他安全瑕疵）。这个工作从漏洞产生的主观性出发对软件瑕疵进行分类，分类的结果有助于帮助开发者排查、寻找软件瑕疵。

McAfee 公司的安全专家 MarkDowd 撰写了一本类似于软件安全开发教学专著[23]，其后的[12]与其非常相似。这本书系统的讲述了几个特定类型的漏洞，如内存破坏漏洞（Memory Corruption）、字符串和超字符漏洞（Strings and Metacharacters）等。该书主要针对的是软件开发人员和评测人员，为他们提供了一些审核代码安全性的实用方法等。

法国 INRIA-Amazones 团队的 PierreParrend 工程师在2008年时发表文章讨论了 Java 面向服务编程组件中的漏洞分类。该文章从两个视角对 OSGi 平台上 Java 组件的安全漏洞进行分类学研究，一个是漏洞的类别，另一个是借用漏洞进行攻击的目的。该文章首先搜集了大量的 Java SOP 漏洞，对它们进行了详细研究，发现所有的 Java SOP 漏洞都是 Object 漏洞，其一个子集是 Class 漏洞，只有少数几个漏洞是独立的漏洞（Standalone）。针对 JavaSOP 的这个特性，这篇从漏洞类型角度文章将 Java 组件漏洞分成：独立漏洞、类共享漏洞和面向服务的漏洞，同时给出了每个类型漏洞的例子。从漏洞利用攻击的目标分类为：过度访问和拒绝服务。同时相关实验证明了分类的有效性。

4 总结和展望

软件安全漏洞分类是软件安全领域的一项基础研究，已有许多不同种类的分类方法和分类学体系。从本质上来说，分类的动机取决于需求。软件开发者希望能够参考以软件开发视角进行的漏洞分类；系统设计者倾向于从系统设计视角分类漏洞；安全研究者更偏好于从漏洞成因进行的分类，例如从攻击角度进行的分类（如 CAPEC）。分类的视角不同会导致分类的结果不同。在需要进行软件安全漏洞分类时，看清分类需求，才能选择合适的分类方法和分类学体系。

随着计算能力高速的发展，许多不可能或是代价太大的计算任务变得可行而且简单。例如最近很流行的基于动态污点分析[4,5,6]的漏洞特征刻画方法可能会成为今后漏洞分类的一个新方向。云计算的引入更进一步加强了计算能力，恰恰精确的识别漏洞特征也需要这样的能力。基于云计算的漏洞信息搜集和分析、分类可能成为今后几年软件安全的研究课题之一。

参考文献

[1] RP Abbott, JS Chin, JE Donnelley, W L Konigsford. Security analysis and en hancements of computer operating systems [R]. Tech. rep., 1976.

[2] The Owasp Foundation. Open web applicatio n security project (OWASP)[Z],2009.

[3] R BisbeyII. Protection analysis: Final report [R]. Tech. rep., 1978.

[4] Edward J. Schwartz, Thanassis Avgerinos, Da vid Brumley. All You Ever Want- ed to Kno w about Dynamic Taint Analysis and Forw ard Symbolic Execution (but Might Have Been Afraid to Ask)[C]. 2010 IEEE Sympo sium on Security and Privacy. IEEE, 201 0, 317–331.

[5] W. Cheng, S. Hiroshige. TaintTrace: Effi cient Flow Tracing with Dynamic Binary Rewriting[M]. IEEE, 2006.

[6] James Newsome, Dawn Song. Dynamic Ta int Analysis for Automatic Detec- tion, Anal ysis, and Signature Generation of Exploits on Commodity Software[J].

[7] CB Hogan. Protection imperfect: The se curity of some computing environ- ments [J]. ACM SIGOPS Operating Systems Re view. 1988.

[8] Wenliang Du, Aditya P Mathur. Categorizatio n of Software Errors that led to Security Bre aches[C]. 21st National Information System s Security Conference.1997, 392–407.

[9] Katholieke Universiteit Leuven. Developi ng Secure Software, A survey and classi fication of common software vulnerabilities [C]. IICIS Conference. 2001, 194–204.

[10] H Langweg. A classification of malicious software attacks[J]. Performance, Comp uting, and. 2004.

[11] HH Thompson. The software vulnerabil ity guide[M]. Charles River Media, 200 7.

[12] George Kurtz, Matt Bishop. REVIEWS FOR 24 DEADLY SINS OF SOFT- WARE SECURITY[M]. McGraw-Hill Osborne M edia, 2009.

[13] RC Seacord. A structured approach to clas sifying security vulnerabilities[R]. Tech. Rep. January, Carnegie Mellon University, 2005.

[14] Sophie Engle, Sean Whalen, Damien Howar d. Tree approach to vulnerability classificat ion[J]. Dept of Computer. 2006, (May):1–1 0.

[15] CE Landwehr, AR Bull, JP McDermott. A taxonomy of computer program secur ity flaws, with examples[J]. Computing Sur veys. 1993, 3(26):1–36.

[16] T Aslam. A taxonomy of security faults in t he unix operating system[D]. Ph.D. thesis, 19 95.

[17] T Aslam, Ivan Krsul. Use of a taxonomy of security faults[C]. the 19th National Informa tion Systems Security Conference. 1996.

[18] Ulf Lindqvist. How to systematically classi fy computer security intrusions[J]. Security and Privacy, 1997.

[19] John D Howard, Thomas A Longstaff. A Common Language for Computer Securi ty Incidents[R]. Tech. Rep. October, Sandia National Laboratories, 1998.

[20] K. Jiwnani, M. Zelkowitz. Maintaining sof tware with a security perspective[J]. Internati onal Conference on Software Maintenance, 2 002 Proceedings:194–203.

[21] a. Avizienis, J.-C. Laprie, B. Randell, C. La ndwehr. Basic concepts and taxono- my of dependable and secure computing[J]. IEEE Transactions on Dependable and Secure Co mputing. Jan. 2004, 1(1):11–33.

[22] PA Karger, A Paradkar. A software flaw tax onomy: aiming tools at security[J]. ACM SI GSOFT Software. 2005:1–7.

[23] M Dowd, J McDonald. The art of soft ware security assessment: Identifying and preventing software vulnerabilities[M]. Ad dison-Wesley, 2006.

[24] Matt Bishop. Computer Security: Art an d Science[M]. Addison-Wesley, 2002 .

[25] Ivan Victor Krsul. Software Vulnerability A nalysis[D]. Ph.D. thesis, 1998.

[26] CV Berghe, James Riordan. A vulnerabi lity taxonomy methodology applied to w eb services[J]. 10th Nordic Workshop on Secure IT Systems (NordSec).2005:1–14.

[27] S Christey. PLOVER: Preliminary list of v ulnerability examples for researcher- s[J]. NI ST Workshop Defining the State of the Ar t of Software. 2005.

[28] Web Application Security Consortium. Web application security consortium threat classification[Z], 2009..

[29] WASC. The WASC Threat Classificatio n[Z].

[30] W.D. Yu, D. Aravind, P. Supthaweesuk. Software Vulnerability Analysis for Web S ervices Software Systems[J]. 11th IEEE Sy mposium on Computers and Communication s (ISCC'06). 2006:740–748.

[31] SecurityFocus. Security focus vulnerabilit y database[Z], 2009.

[32] SANS. SANS Top 20 Security Risks[Z], 2007.

[33] MITRE Corporation. Common Attack Patter n Enumeration and Classification (CAPEC)[Z]

[34] MITRE Corporation. Common Weakness Enumeration (CWE)[Z].
[35] NIST. National vulnerability database (NVD)[Z].
[36] NIST. National vulnerability database RSS feed[Z].
[37] Canvas. Canvas[Z]. http://immunityinc.com/
[38] Metasploit. Metasploit Framework[Z].
[39] Open Security Foundation (OSF). Open Source Vulnerability Database (OS- VDB)[Z].
[40] IBM Internet Security Systems X-Force. Alerts and advisories[Z].
[41] VUPEN Security. Vulnerability management and penetration testing[Z].
[42] Microsoft Corporation. Security bulletins [Z].
[43] SecWatch.org. Search portal[Z]. URL http://secwatch.org/.
[44] SANS. Newsletter[Z]. URL http://www.sans.org/newsletters/.
[45] Milw0rm. Milw0rm[Z]. URL http://milw0rm.com.
[46] Secunia. Secunia[Z]. URL http://secunia.com.