软件漏洞的分类研究①

Study on Computer Vulnerability Taxonomy

王　颖　李祥和　（解放军信息工程大学 信息工程学院 河南 郑州 450002）

摘　要： 文章分析了 Microsoft 公司、CVE 安全组织和 Fortify Software 公司对软件漏洞的分类方法，指出了它们的分类方法不能有效反映出软件漏洞性质的不足。为此，文章首先提出了一个新的软件漏洞的分类方法，再以复杂巨系统理论结合系统工程中定性定量的方法对软件漏洞进行了具体分类，最后给出了漏洞分类同构模型、应用方法及复杂度计算方法。

关键词： 软件漏洞 分类方法 系统工程 同构模型 利用复杂度

软件漏洞又被称为软件的脆弱性，目前对软件漏洞的分类在国际上还没有统一的标准和通用的方法。文章对国际上有重要影响的计算机安全组织——Microsoft 公司、CVE 安全组织、和 Fortify Software 公司对软件漏洞的分类方法进行分析，总结出它们各自的特点，并提出了一种新的分类方法，目的是提供一个软件漏洞分类的通用方法，并为软件漏洞的防治和理论研究提供参考。

1　三个安全组织的分类方法

1.1 Microsoft 公司的分类方法

Microsoft 公司[1,2]在两个维度上对软件漏洞分类：1. 根据产品类别对漏洞分类。2. 根据漏洞产生的影响和利用方法对漏洞分类。如表 1 所示：

表 1　Microsoft 公司的漏洞分类方法

分类方法		软件漏洞类型
1. 产品类别		操作系统漏洞、Office 软件漏洞、服务器软件漏洞、开发软件漏洞等
2. 漏洞的影响和利用方法	严重等级	严重、重要、中等、低
	威胁类型	欺骗身份、篡改数据、否认、信息泄露、拒绝服务、特权升级等

Microsoft 公司分类方法适合该公司的安全策略，能够有效地保证软件产品的安全性，为用户提供较好的服务。但是，该分类法只是针对 Microsoft 公司的软件产品中存在的漏洞，对漏洞所作的分类并不全面；并且根据该分类方法对漏洞的分类不唯一，没有从系统关系中有效地反映漏洞的运行机制和本质特征。例如“MS07 – 058”漏洞对软件产品“Windows XP Service Pack 2”是“拒绝服务”威胁类型，而对同类软件产品“Microsoft Windows 2000 Service Pack 4”是“信息泄露”威胁类型。

1.2 CVE 安全组织的分类方法

CVE 组织[3–5]对漏洞的分类方法是一维的、基于抽象分类规则的枚举方法。有四条分类规则：1. 具有不同属性特征的漏洞划分为不同的类别。2. 存在范围不同的漏洞划分为不同的类别。3. 属性特征和存在范围都相同的漏洞划分为同一类别。4. 基于共同"codebase"的软件漏洞在分类时，要以引起漏洞的关键代码的性质为基础，而不仅仅根据产品的类型来分类。根据以上规则，目前 CVE 组织在其数据库（version 20061101）中把已知的安全弱点枚举为 28171 项漏洞条目，并把它们划分为 37 个漏洞分类，其中主要有跨站脚本、缓冲区溢出、SQL 注入、PHP 远程包含、目录遍历、信息泄漏、dos – malform、link、格式化字符串等漏洞类型。

① 基金项目：总装预研基金（5140001020 55B5201 – 1）

CVE 组织的漏洞分类方法为国际安全组织提供了一个漏洞标准列表,有效地提高了安全组织中各部门之间的协同工作能力。为用户进行风险评估提供参考。但这种分类方法还没有形成一个完整的方法体系。没有归纳出漏洞的属性特征,对漏洞的分类较多地依赖于经验。

1.3 Fortify Software 公司的分类方法

Fortify Software[6] 公司以软件漏洞的根本起因—代码缺陷为出发点,通过对代码缺陷分类,间接地实现了对软件漏洞的分类:由同类代码缺陷所引起的软件漏洞是一类。Fortify Software 公司对代码缺陷的分类分为两个层次:1. 在程序编码的层次上归纳出在软件实现过程中的各种代码缺陷;2 根据漏洞攻击的特征,相应的把具有相同特征的各种代码缺陷划归为一类。Fortify Software 公司把代码缺陷分为八类,并且根据对软件安全的重要性由前到后把它们排序如下:①程序输入②API 调用错误③安全策略④时间和状态⑤错误处理⑥代码质量⑦区分⑧环境。

Fortify Software 公司对代码缺陷的分类在总体适用于大部分编程语言,同时在每一类中,具体的代码缺陷分别适用于具体的编程语言。这种分类方法对于开发安全的软件产品、设计有效的漏洞防治工具,制定可靠的软件安全策略等安全问题,可以提供实践上的指导。

以上三种分类方法都是以商业服务为目的、从商品经营的角度对软件漏洞进行分类,不能从系统的整体关系中掌握漏洞的利用机制和本质特征,没有体现出漏洞利用过程的复杂度,不能对漏洞的防治在原理和方法论上提供参考。Anil Bazaz[7,8] 等学者在最近相关的研究工作中,论证了软件漏洞是由于软件在系统中执行时与系统中其它对象之间产生了非正常的关系,而使系统处于不安全的状态所导致的系统运行错误。基于以上思想,文章从系统工程的观点,提出一个基于软件漏洞利用过程的分类方法,目的是利用该分类方法,系统地描述在漏洞利用过程中系统整体关系的变化规律,以及软件代码缺陷的利用方法,为深入地理解漏洞的本质、掌握漏洞的利用机制提供参考,并为漏洞利用的难度和复杂度测量提供一个定量的计算方法。

2 基于利用过程的分类方法

2.1 理论基础

根据钱学森[9] 对复杂巨系统理论的论述,计算机系统是开放的复杂巨系统——社会系统的子系统,现在能用的、惟一能有效处理开放的复杂巨系统的方法,就是定性定量相结合的综合集成方法。软件漏洞如上所述,是计算机系统关系的整体性问题,只有应用系统工程的方法,才能进行有效处理。文章采用定性定量相结合的综合集成方法对漏洞进行分析,并以此为基础对漏洞进行分类。

2.2 分析软件漏洞的定性定量相结合的综合集成方法

1). 定性方法:软件漏洞的利用过程是软件漏洞发生机制的具体体现,与软件漏洞之间是一一对应的关系,是对软件漏洞本质的定性反映。我们根据利用过程对软件漏洞进行定性分析。

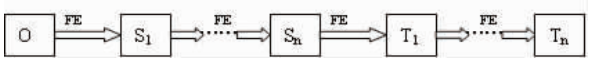
2). 定量方法:文章采用崔宝江和周继军在文献[10]中提出的、从程序语言性质的角度对代码缺陷进行分类的方法,根据漏洞利用过程所对应的代码缺陷序列对漏洞进行定量分析。我们首先引入如下概念(详见参考文献[10]):

(1). 系统错误 (2). 数据性错误(3). 操作性错误(4). 运行错误(5). 数据性代码缺陷(6). 操作性代码缺陷。(7). CZ 分类法:文章把崔宝江和周继军提出的代码缺陷的分类方法称为 CZ 分类法。(8). 代码缺陷的分类结构:代码缺陷总的分为数据性代码缺陷和操作性代码缺陷。在每一类中,根据 CZ 分类法递归地分类,可以使分类达到系统分析所要求的、能体现漏洞特征微小区别的精细划分。

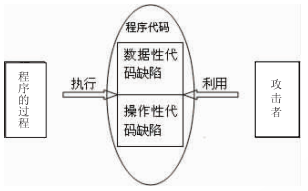
基于以上概念,我们给出对漏洞利用过程的定量描述:(1). 攻击者进行漏洞利用的过程是利用软件代码缺陷引起软件运行错误的过程,漏洞的产生是由一系列的运行错误组成。(2). 漏洞利用过程中的每一步,都对应一个由代码缺陷所引起的运行错误。攻击者对代码缺陷的利用方式有三种:利用数据性代码缺陷、利用操作性代码缺陷、同时利用数据性和操作性代码缺陷。(3). 每一个漏洞利用过程都对应一个代码缺陷序列,序列中的每一位都是一个代码缺陷集合,分别是利用过程中的每一步所利用的代码缺陷的全体。集合有三种类型:包含数据性代码缺陷的集合、包含操作性代码缺

陷的集合、包含数据性和操作性代码缺陷的集合。

3). 软件漏洞的系统分析与测量方法：我们采用定性定量的综合集成方法，根据以上对软件漏洞性质的分析，构建软件漏洞的定性定量模型——软件漏洞的利用模型，如图 1



(a)漏洞利用过程



(b)代码缺陷的利用方式

图 1　软件漏洞的利用模型

在图(1a)中，O 代表软件的原始运行状态，FE 代表由代码缺陷所引起的运行错误。我们定义 S 是系统不安全状态集合，T 是系统错误状态集合，$S_1 \cdots S_n \in S$，$T_1 \cdots T_n \in T$。CZ 分类法是从程序语言的角度和层次上对代码缺陷分类，它可以使我们从系统关系的性质上对代码缺陷进行深入地研究分析。在此基础上，我们根据利用模型，能够对所有的软件漏洞进行定性定量相综合的系统分析：每个与漏洞相对应的代码缺陷序列都是反映漏洞属性的矢量，都是对漏洞利用的难度和复杂度的描述，这样就实现了根据代码缺陷对漏洞的定性定量的分析。

我们根据利用模型设计出一个数学方法，定量定性地计算软件漏洞的利用难度和复杂度。软件漏洞的利用机制由两个方面的因素决定：1. 代码缺陷的利用方式，包括利用强度和利用长度。其中利用强度是指漏洞利用过程中的每一步所利用的代码缺陷的数量；利用长度是指被利用的代码缺陷序列的长度；2. 编写软件的程序语言的性质，这主要表现在并发性与连续性两个方面，它们分别与软件漏洞利用机制的利用强度和利用长度相对应。现分别解释：(1). 程序语言的并发性是指在理想状态下，程序语言编写的程序在系统中执行时，各部分之间的并发操作在空间上的配合关系，即各部分操作之间协调与配合的紧密程度；(2). 程序语言的连续性是指在理想状态下，程序语言编写的程序在系统中执行时，各部分的操作在时间上的连续性，即每一种顺序操作前后承接的耦合程度。我们给出漏洞利用难度和复杂度的计算公为：

$$p = \begin{bmatrix} s & t \end{bmatrix} \begin{bmatrix} c \\ l \end{bmatrix} \qquad (1)$$

其中 p 代表漏洞利用的难度和复杂度；s 和 t 分别是描述程序语言连续性和并发性的参数。各种程序语言所对应的 s 参数与 t 参数的值要用知识工程中的知识表达的方法来确定，即需要在实践经验的基础上，用定性定量相结合的方法来求出每一种程序语言所对应的值，例如根据专家系统来确定 s 参数与 t 参数的值；c 和 l 分别对应利用强度的算术均值和利用长度。设漏洞利用过程共有 n 步，则 $C = (C1 + C2 + \cdots + Cn)/n$，其中 C1，C2，…，Cn 分别是每一步所利用的代码缺陷的个数。$l = n$，即 l 表示漏洞利用的步骤数。这样我们就可以根据公式用数学的方法定量地计算和比较各种、以及各类软件漏洞的利用难度和复杂度。

2.3 软件漏洞的分类方法

文章根据以上对软件漏洞性质的分析结果，应用系统工程的方法，通过一一映射的关系，建立定性定量的同构模型，对软件漏洞进行分类。

(1)分类规则：根据共性对软件漏洞分类。通过第 3.2 节的分析可知：软件漏洞的利用过程对应唯一的代码缺陷序列，是对软件漏洞定性定量的描述。代码缺陷序列表达了软件漏洞利用机制的共性：利用方法的决策过程和利用过程的复杂程度。我们把它作为软件漏洞的分类标准。我们建立分类规则如下：

①. 把对应同一代码缺陷序列的利用过程合并为同一集合，表示为同一种软件漏洞利用过程。

②. 根据代码缺陷序列对软件漏洞利用过程分类：包含 n 个代码缺陷集合的代码缺陷序列为 n 次序列；对应于 n 次序列的漏洞利用过程称为 n 次利用型，其中 n 是自然数。

③. 把软件漏洞利用过程与其所引起的漏洞对应起来，建立它们之间的映射关系。这样，对应同一种漏洞利用过程的软件漏洞可以用同一代码缺陷序列失量进行描述，归为同一种漏洞。对应于 n 次利用型的软件漏洞称为 n 次型漏洞。

④. 在软件漏洞利用过程中的每一步，攻击者对代

码缺陷的利用方式只有三种，漏洞的利用过程就是一个选择代码缺陷的过程。我们从系统工程的角度，建立软件漏洞利用过程的决策树，并应用③中所确定的对应关系，构建软件漏洞的分类结构的同构模型，如图 2。

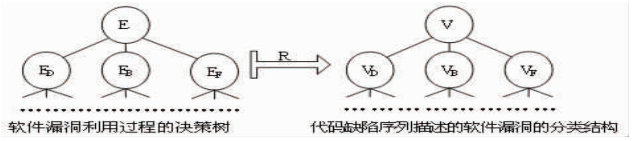


图 2　同构模型

在上图中，左边的三叉树代表软件利用过程的决策树，右边的三叉树代表根据代码缺陷序列确定的软件漏洞的分类结构。其中 R 表示一一映射关系；E 代表利用过程的起点，E_D、E_F、E_B 分别代表漏洞利用的三种方式；V 代表软件代码缺陷序列的起点，V_D、V_B、V_F 分别代表三种类型的代码缺陷集合。在漏洞利用过程的决策树中，从根节点到树中的任一节点之间的路径都是一个决策过程，都代表了一种可能的漏洞利用方式。相应的，在右边软件漏洞分类结构的三叉树中，从根节点到树中任一节点之间的路径都代表了一个代码缺陷序列，确定了一种软件漏洞。

⑤根据定性定量相结合的方法，用代码缺陷序列作为描述软件漏洞属性的矢量，把代表漏洞分类的三叉树中同层节点所对应的漏洞划分为一类。

⑥. 应用 CZ 分类法对代码缺陷递归分类，把数据性代码缺陷和操作性代码缺陷分别划分为 m 项子类与 n 项子类，使得 m 项子类之间、n 项子类之间、m 项子类与 n 项子类之间的差别达到系统分析所要求的精确程度。我们把这 m 项子类与 n 项子类分别带入每种软件漏洞所包含的代码缺陷序列中，所得到的漏洞称为每种软件漏洞的实例，他们分别属于该种软件漏洞的实例集合。我们对漏洞实例的划分是根据实际的系统分析所进行的动态的划分，对代码缺陷作不同次数的递归分类，所得到的漏洞实例也不相同。

(2)软件漏洞的分类结构：文章对软件漏洞的分类是三层结构：1. 包含相同数目的代码缺陷集合的软件漏洞为同一类。我们根据每类中所包含的代码缺陷集合的数目把它们分为一次型、二次型、…、n 次型，n 是自然数；2. 包含相同代码缺陷序列的漏洞为同一种软件漏洞。n 次型软件漏洞包含 n 个代码缺陷集合，共有 3n 种可能的软件漏洞；3. 每种软件漏洞的实例。现在已发现的各个软件漏洞根据其所包含的代码缺陷序列，作为漏洞实例分别被划分到每种软件漏洞实例的集合中。

3　分类方法的应用

3.1 应用举例

下面分别应用文章和三个公司的漏洞分类方法分别对 Microsoft 公司在 2007 年 12 月份发布的两个漏洞 MS07－068 和 MS07－069 进行分类，比较它们对漏洞性质的分析能力。在计算软件漏洞的复杂度时，由于这两个漏洞都对应于同一种程序语言，我们设向量[st]=[11]。比较结果如表 2。

表 2 四种分类方法对相同软件漏洞所作的分类

分类方法	分类方法对 MS07－068 漏洞性质的描述	分类方法对 MS07－069 漏洞性质的描述
Microsoft 公司的分类方法	严重等级：严重；威胁类型：执行远程代码	严重等级：严重；威胁类型：执行远程代码
CVE 组织的分类方法	堆缓冲区溢出漏洞	内存破坏漏洞
Fortify 公司的分类方法	“程序输入”类代码缺陷	“API 调用错误”类代码缺陷
文章的分类方法	利用过程：{参数验证}；复杂度：1	利用过程：{参数验证，文件操作}；复杂度：2

我们看到，对上述两个漏洞，同三个安全组织的分类方法相比，文章的分类方法不同的是：把代码缺陷序列作为描述漏洞利用过程的向量，描述了这两个漏洞的共同点——它们都利用了“参数验证”这一代码缺陷，显示了它们的区别：代码缺陷的利用次数和利用复杂度不相同。根据文章的分类方法对代码缺陷序列进行系统关系的整体性分析，可以找出产生漏洞的本质原因。例如 MS07－068 漏洞的利用过程是对{参数验证，文件操作}代码缺陷序列进行的顺序操作，我们据此分析出这个漏洞的利用原理：完成一个特定功能的

所有连续操作之间的整体性关系松散,即系统中连续操作之间没有相互照应。另外根据文章的分类法,可以制定有效的漏洞防治策略,例如我们可以根据利用复杂度确定漏洞检测的顺序:因为利用复杂度小的漏洞更容易检测,先检测利用复杂度小的漏洞可以提高检测效率。综上所述,文章提出的分类方法是系统的分类方法,能有效地表达出软件漏洞的性质。

4 结论

文章从系统工程研究角度入手,提出了一个定性定量相结合的软件漏洞的分类方法。目的是建立一种较为通用的分类方法,作为软件厂商开发安全产品的测试和审计工具,并且为科技人员对漏洞的防治和漏洞性质的深入研究提供参考。

参考文献

1 Microsoft Corporation. Update Management Process. 2007. http://www.microsoft.com/technet/security/guidance/patchmanagement/secmod193.mspx.

2 Microsoft Corporation. Microsoft SecurityResponse Center Security Bulletin Severity Rating System. 2002. http://www.microsoft.com/technet/security/bulletin/rating.mspx.

3 Steven Christey, Robert A. Martin. Vulnerability Type Distribution in CVE. 2007. http://www.cve.mitre.org/docs/vuln-trends/index.html.

4 Steven M. Christey. CVE Abstraction Content Decisions: Rationale and Application. (Version 1.0). 2005. http://www.cve.mitre.org/cve/editorial_policies/cd_abstraction.html.

5 MITRE Corporation. CVE (version 20061101) and Candidates as of 20071226. 2007. http://www.cve.mitre.org/data/downloads/allitems.html.gz.

6 Fortify Software Inc., Gary McGraw. Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors. 2006. http://www.fortifysoftware.com/docs/Fortify_TaxonomyofSoftwareSecurityErrors.pdf.

7 A Bazaz, J D. Arthur. Towards A Taxonomy of Vulnerabilities. Proceedings of the 40th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, Washington, DC, USA. 2007. ISBN ~ ISSN:1530-1605, 0-7695-2755-8. On page(s): 163a-163a.

8 Bazaz A, Arthur, J D. Tront. J G. Modeling Security Vulnerabilities: A constraints and assumptions perspective. To be presented at IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC06), Indiana, 2006.

9 钱学森,于景元,戴汝为. 一个科学新领域——开放的复杂巨系统及其方法论. 自然杂志, 1990 (1): 3—10.

10 崔宝江,周继军. 软件代码缺陷的分类研究. 待发表.