计算机科学2003Vol. 30№. 8

软件脆弱性分析

李新明 李 艺 徐晓梅 韩存兵
(装备指挥技术学院 北京101416)

Software Vulnerability Analysis

LI Xin-Ming LI Yi XU Xiao-Mei HAN Cun-BING
(Institution of Command and Technology of Equipment, Beijing 101416)

Abstract Software vulnerability is the root reason that cause computer system security problem. It's a new research topic to analyze vulnerability based on the essence of software vulnerability. This paper analyzes the main definitions and taxonomies of vulnerability, studies vulnerability database and tools for vulnerability analysis and detection, and gives the details about what caused the most common vnlnerabilities in the LINUX/UNIX operating systems.

Keywords Vulnerability, Vulnerability taxonomy, Vulnerability database, Vulnerability scanner

1 概述

随着计算机、通信、网络技术的发展，网络信息系统已经成为社会的基础设施，网络信息系统不仅是传播信息的手段，而且是管理和控制以及交互的基础工具。社会越发达，对网络信息系统的依赖程度就越高。如何保证网络信息系统的安全、可靠、正常的运行，已成为一个迫切需要解决的问题。小到一台 PC 机、一个公司的计算机系统如何防止黑客的攻击，大到一个国家如何在未来信息战中掌握主动权，系统安全都是一个至关重要的问题。

而系统安全受到各种安全威胁的根本原因是系统中存在脆弱性。网络与信息系统的脆弱性是一个系统问题，覆盖系统的各个方面，包括：系统中物理装备(如计算机硬件、通信线路等)的脆弱性；软件(如操作系统、网络协议簇、数据库管理系统、应用程序等)的脆弱性；人员管理、规章制度、安全策略的脆弱性。

操作系统和网络协议栈软件是网络信息系统的中枢，是网络信息系统的基础，是各种应用软件赖以运行的基础，基于操作系统和网络协议栈软件的脆弱性研究是计算机系统安全的重中之重。脆弱性研究必须解决计算机软件系统中产生脆弱性的原因、脆弱性可能造成的影响、如何利用脆弱性进行攻击、如何修补脆弱性、如何防止脆弱性被利用、如何探测目标系统的脆弱性、如何预测新的脆弱性的存在等一系列问题。

2 软件脆弱性定义

对于什么是软件脆弱性，目前没有形成一个一致接受的定义，由于软件脆弱性的复杂性和考虑的角度不同，对脆弱性存在着各种定义，但总的说来有两类：

一类是精确定义，试图对系统及其行为给出精确描述，并基于此给出系统脆弱性的精确定义，这对于形式化证明、计算机理论研究有较大的价值。最典型的是：

1)访问控制定义[2]：将系统状态通过一个主体、对象和访问矩阵构成的三元组来描述，其中访问控制矩阵指定了系统的安全策略，而利用脆弱性是一切能够引起操作系统执行违反安全策略的操作。这种定义的关键是将系统用三元组进行描述，由安全策略将所有可能的状态分成授权状态和非授权状态，而实际上，在 Unix、Macintosh OS、VMS、Windows NT 这样的操作系统中，都不存在这样清晰和详细的访问控制矩阵。

2)状态空间定义[4]：认为操作系统是由描述实体当前配置的状态组成的，系统运行实际上就是状态迁移。从一个给定的初始状态出发，经过使用一组状态迁移，可以到达所有的状态。依据安全策略的定义，状态迁移分成授权迁移和非授权迁移两类。如果从某个状态开始，经过一系列授权的状态转换可以到达某个非授权状态，则这种状态称为脆弱状态。这种定义也很难应用于 Unix 或者 NT 这样的系统，即使我们可以列举出系统中所有可能的安全和非安全状态，但是仍然不够，因为，对系统进行状态划分不是一个静态、封闭的过程，还必须考虑系统的运行环境，包括用户执行的动作。

第二类是模糊定义，通过描述性语言，对脆弱性的本质进行描述。这类定义通常出现在字典或各种脆弱性数据库中。典型的如：

1)Krsul 提出的基于安全策略的脆弱性定义[2]，认为软件脆弱性是软件规范、开发或配置中错误的实例，其执行结果将会违反安全策略。

2)在数据和计算机安全标准字典中：在计算机安全中，脆弱性是在自动系统安全过程、管理控制、内部控制等中的一个弱点，通过探测它可以获得对信息的非授权访问、或者破坏关键处理。

3)在我国的国家军用标准《军用计算机安全术语》[11]中，定义脆弱性是导致破坏系统安全策略的系统安全规程、系统设计、实现、内部控制等方面的弱点。

3 软件脆弱性分类

按照物体的属性，对物体进行分类是一种被广泛采用的研究方法。要研究软件脆弱性的实质，就必须寻找一种合理的软件脆弱性分类方法。一个成功的分类法首先必须确定物体的属性，这些属性满足客观性、确定性、可重复性和特殊性的

李新明 硕士，教授，主要从事操作系统和网络安全的研究。

特点，然后可以采用一种或多种分类方案进行分类，常用的分类方案有任意选择、决策树、自然分类、进化分类和自然聚合等。在脆弱性分类中通常采用决策树方案，每一个脆弱性沿着决策树回答一系列问题，直到到达树的叶节点，成功地被归为某一类。

目前，软件脆弱性分类法很多，有一定影响的至少有20多个，如基于引入原因、基于语法、基于错误分析、基于属性、基于起源、基于访问需求、基于利用难度、基于部件、基于影响、基于威胁、基于利用复杂性、基于攻击、基于生命周期、基于入侵检测和基于特定系统，如 TEX 系统和 Tester 系统等的脆弱性分类[4]，每一种分类法从各自不同的角度对软件脆弱性的本质和特性进行了描述，都有其特点和不足。但没有一个被普遍接受。典型的软件脆弱性分类方法如下。

3.1 Aslam 基于引入原因的分类

脆弱性的引入原因是脆弱性最本质的属性之一，为什么会产生一个脆弱性？普渡大学的 Aslam 和 krsul 等人研究了 UNIX 操作系统中的一系列的脆弱性，1995年提出了基于引入原因的分类方法[1]，至今仍在不断完善[2]，将脆弱性引入原因分成5种：

·设计错误，包括需求分析和软件设计等过程中的错误；

·环境错误：由于与软件设计时假定的环境不匹配造成的错误。软件系统在设计和实现中，对软件系统的执行环境自觉和不自觉地进行了各种假设，但在实际的执行过程中，这些环境假设并不成立；

·代码错误，程序编码中的错误；

·配置错误，软件本身没有错误，但运行系统时的配置错误；

·其他错误。

上述每一种错误，又有许多小的分类。软件开发是一个很复杂的过程，问题的复杂性、设计的复杂性和程序的复杂性，都会增加程序员设计和编写软件系统的难度，增加出现错误的机会。软件生命周期的各个阶段都可能产生脆弱性，如图1所示。

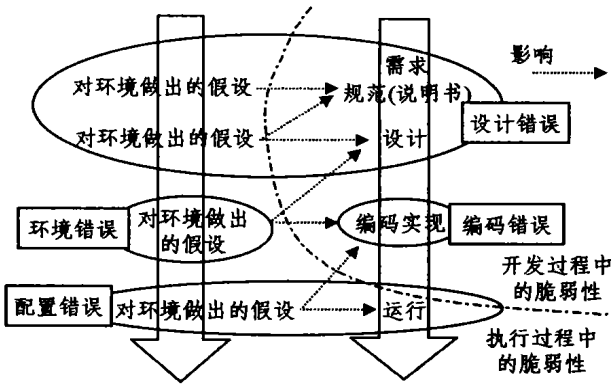

图1 脆弱性在软件生命周期的分布图

这种分类方法存在着二义性，如 xterm 日志脆弱性可以被归结到许多类中[5]。

3.2 基于脆弱性所在部件的分类

各种脆弱性数据库一般都根据脆弱性所在部件将脆弱性进行分类，通过将脆弱性定位到一个确定的系统部件，可以很方便地进行脆弱性验证和探测。脆弱性所在的部件可以用图2来表示。

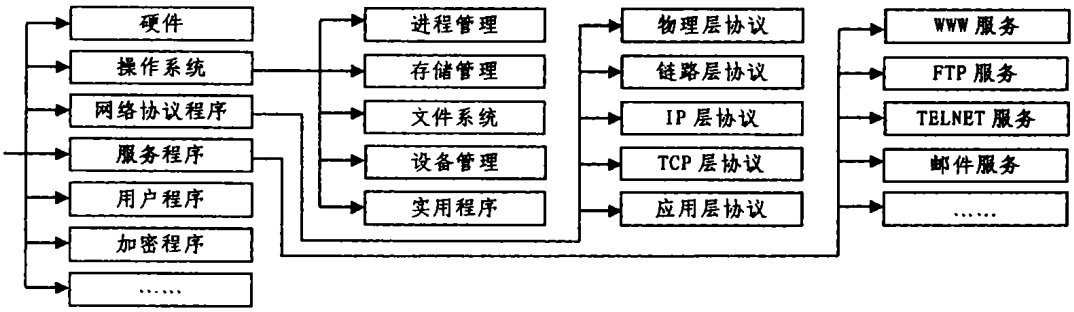

图2 基于脆弱性所在部件的分类法

3.3 Richard 等人基于语法的分类

1987年，圣托马斯学院的 Richard 等提出一种基于语法的故障分类法。“语法是语义的载体”，程序的任何错误自身都表现为在代码中的语法错误，本分类方法是根据确定并修复一个错误的方法来进行，分成以下几种：

1）实体虚假：修复错误时需要将虚假实体的特征子串删除；

2）实体缺少：修复错误时需要插入一个符合语法的实体；

3）实体位置错误：修复错误时需要在代码中改变实体的位置；

4）实体错误：所有其他错误。

由于修复一个错误有很多种方法，根据不同的修复方法，同一个脆弱性被划分到不同的类型中，这是不对的，存在二义性。

3.4 Longstaff 基于利用难度的分类

1997年，Longstaff 提出了基于利用脆弱性的难易程度进行的分类，将脆弱性分成以下几种：

1）简单命令

2）可用的工具包

3）需要专家

4）用户采取动作就能确认

5）管理员采取动作才能确认

实际上，我们无法对每一种脆弱性决定其利用的难易程度，因为无法知道是否有可用的利用脚本或者工具集，可能在已经为某种分类选择取值之后，才出现一种工具或程序。因此这种分类的取值与时间相关，过去一个需要专家才能利用的脆弱性，现在可能是一个有现存工具包就能利用的脆弱性。

3.5 基于影响的分类

这种分类要确定脆弱性的影响，包括确定直接或者间接的影响。直接影响是指当脆弱性被利用时立即就能觉察到的影响，而间接影响是指利用某脆弱性后最终所造成的影响。这种分类在脆弱性数据库中非常常见。基于影响，将脆弱性分成

以下几类：

1)非法访问数据：包括非法访问系统或用户级数据或者导致数据丢失

2)非法执行命令：如用户软件通过执行用户级命令而破坏管理员设定的访问控制集

3)非法执行代码：包括在系统或用户权限下非法执行机器语言代码或脚本

4)拒绝服务：包括系统资源被用完或被删除

这种脆弱性的分类方法有多义性，例如，Unix 中攻击者重写文件/vmunix 的脆弱性，直接影响是系统无法工作，同时信息被修改。最终的影响是拒绝服务和丢失系统数据。

3.6 Power 基于威胁的分类

1996年，Power 基于脆弱性所产生的威胁，进行脆弱性分类如下：

1)威胁可用性和用途，如删除数据、拒绝用户访问等

2)威胁整体性和验证，如产生错误的数据、伪装身份等

3)威胁可信性和所属关系，如非法访问、拷贝、窃取数据等

4)其他

这种分类也存在多义性的。如“威胁可信性和所属关系”一类脆弱性的继续划分时，有可能访问和窃取同时发生。另外，该分类在确定脆弱性的威胁类型时，没有显式指定间接影响的最大值。在 Unix 和 Windows NT 等管理员具有无限制权利的系统中，使用管理员帐号意味着所有威胁。例如，如果攻击者获得了 Unix 的 root 帐号，那么他就可能删除硬盘(属于1类)、修改密码文件(属于2类中产生错误数据一类)、假扮其它用户发送 email(属于2类中伪装身份一类)、读取其它用户的 email(属于3类中非法访问一类)等。

3.7 Cohen 基于攻击的分类

1995年开始，Cohen 对系统的一百个可能的攻击集进行了收集和分析，并据此对脆弱性进行分类如下：

1)错误和疏忽，如忘记删除缺省密码、不正确地设置保护等

2)调用时的无效值：用无效的值调用系统导致操作系统出错

3)隐含的信任攻击：如一个程序不适当地信任另一个程序

4)数据欺骗：非法地修改数据而跟踪操作系统，并产生错误的结果

5)过程旁路：旁路某些控制不完全的控制过程

6)输入溢出：攻击没有对输入长度进行检查的程序

7)分布式协同攻击：通过中间系统攻击

……

4 脆弱性数据库

目前许多组织都建立和维护了各种公用和私用的脆弱性数据库和 BUG 跟踪系统，它们提供的信息可以帮助进行脆弱性的标识和定位。但提供多少信息则需要权衡，一方面要给系统管理员以足够的帮助，另一方面又要避免给恶意实体有用的信息。例如 CERT 脆弱性数据库中就没有提供利用脆弱性的信息。脆弱性数据库管理提供了接口接收用户提供的发现的任何脆弱性信息，同时对各种脆弱性进行确认、归类、发布。一般，各个软件厂商都维护着各自产品的脆弱性数据库，这些数据库不是公开的，厂商一般只发布各种修补脆弱性的软件补丁。

脆弱性数据库是一个非常有用的资源，但是有关脆弱性数据库有许多问题要解决，谁来支持脆弱性数据库、允许谁访问脆弱性数据库信息、基于什么判断访问是否有效、是否用匿名数据、如何在往脆弱性数据库中增加信息的同时防止潜在的黑客攻击、脆弱性数据库如何组织，目前有一个组织 TRUSTED[7]试图对脆弱性数据库的组织进行研究，并提出了集中式、联合式、开放资源和现在的障碍式的数据库组织形式，他们认为现在处于第四种形式，虽然有各种脆弱性数据库，但互相之间没有合作，互相访问很困难，甚至没有可能。

最典型的脆弱性数据库是美国计算机紧急情况反应小组协作中心 CERT/CC 维护的 CERT 脆弱性数据库，1995年建立，包括脆弱性提示(Notes)数据库和脆弱性建议(Advisory)数据库两部分。

另一个值得一提的是通用脆弱性及暴露数据库 CVE，CVE 不是一个独立建立的脆弱性数据库，而是一个字典，将各种脆弱性数据库连接到一起。CVE 目前包含1604个正式条目和1796个候选条目。CVE 最大贡献是提供了一种统一的脆弱性命名方法，虽然命名方法不是基于脆弱性分类，而是基于脆弱性的发现时间顺序排队。现在通过回答一个简单的问题：“产品可以覆盖到多少个 CVE?”，就可以对各种安全产品进行横向比较了。

5 典型的软件脆弱性分析

操作系统和网络协议栈中一些安全漏洞不断被挖掘出来。无论是 UNIX 还是 NT，都有其脆弱性，都存在安全漏洞。Windows NT 通常被认为是更不安全的操作系统。已经发现的部分安全漏洞如：

·安全帐户管理(SAM)数据库可以复制；

·利用工具可以获得 Administrator 级别的访问权；

·在注册对话框中显示最近一次注册的用户名；

·WindowsNT 和 Windows95的客户可以保存口令于文件中，以便快速缓冲；

·文件句柄可能从内存中被读取到，然后用来访问文件，而无需授权；

·通过 FTP 有可能进行无授权的文件访问；

·任何用户可以通过命令行方式，远程查询任何一台 NT 服务器上的已注册的用户名；

·用 Ping 命令可能使一台 NT 机器“自杀身亡”。

UNIX/LIUNX 系统的脆弱性和网络协议栈的脆弱性也很多，最典型的几种如下：

5.1 缓冲区溢出脆弱性

缓冲区溢出是向一个缓冲区填充超过它处理能力的数据所造成的结果。缓冲区溢出脆弱性是最为常见的一种脆弱性，被发现和利用的缓冲区溢出脆弱性呈现上升趋势，它有很多种变型。

fingerd 缓冲区溢出脆弱性是缓冲区溢出脆弱性最典型的例子[3]。Finger 是 UNIX 类系统中常用的一个实用程序，通过它可以获得远程用户名。用户通过 fingerd 向远程用户发出请求，远程用户通过 finger 返回最多512个字符长的远程用户名，由于 fingerd 没有检查返回长度，返回区直接从堆栈分配，堆栈的下一个单元是 fingerd 的返回地址。黑客返回一个大于512字符的字符串，其中512个字符后紧接一个黑客程序的入口地址，则 fingerd 将按照黑客的返回串修改堆栈，将原返回

地址修改为黑客程序的入口地址。当 fingerd 返回时，将直接执行黑客程序。这种脆弱性一般是由于参数和返回值的合法性检查不充分造成的。

5.2 文件访问中竞争条件脆弱性

在 UNIX 操作系统中，进程只有通过系统调用才能访问文件，UNIX 操作系统提供了两种不同的寻址方式：文件路径名和文件描述符。由于 UNIX 文件系统是一个树型结构，采用文件路径名寻址方式需要对文件至少寻址一次，因此它是一种间接方式。而采用文件描述符寻址方式是直接方式。

当两个进程同时对一个文件进行访问时，如果有一个采用文件路径名方式来引用文件，就有可能无法保证操作的原子性，从而导致竞争条件产生[6]。

5.3 TCP 序列号预测脆弱性

标准的 TCP 连接建立序列采用的是三方握手机制。握手的双方通过序列号来判断是否是等待的数据包，而该序列号可以被黑客诊听并计算出来，所以，黑客可以伪造回答包。

5.4 IP 碎片脆弱性

有关 IP 碎片的脆弱性[9,10]较多，如微小碎片脆弱性就是由于碎片太小，要用第二个碎片才能够装下全部的 TCP 包头，而一般的过滤规则不对碎片包处理，那么，过滤器规则对第二个碎片中的 TCP 头的字段就不起作用了。

又如重叠碎片脆弱性[3]，由于允许新的碎片与先前收到的碎片中有重复的地方，当碎片重发后，如果重复的部分有不一致的地方，而在目的地，碎片到达的顺序不同，从而导致重组后数据包的内容不同。

又如 Linux 中泪滴脆弱性是一个在 Linux 的 IP 碎片重组程序中存在的脆弱性。在重组某些有重叠部分的碎片时，可能会导致系统崩溃。

5.5 delivermail 附加到文件脆弱性

本脆弱性存在于 BSD 操作系统中，当用户运行程序 delivermail 发送消息时，可以不发送给指定的用户，而是发送到某个文件，则邮件信息附加到指定文件尾，非法用户可以利用此脆弱性，在/etc/passwd 文件中增加任意条目，非法创建超级用户。

6 脆弱性分析和探测工具

现在，任何一个平台上都有几百个熟知的脆弱性，如果靠人工进行测试是一项极其繁琐的工作，而扫描程序[8]能轻易地解决这些问题。开发者利用可得到的常用攻击方法并把它们集成到整个程序中，这样，使用者就可以通过分析输出的结果发现系统的脆弱性。

扫描器是能够自动检测远程或本地主机安全脆弱性的程序。扫描器能够发现一个主机和网络，进而发现这台主机上有什么服务正在运行，最后通过测试这些服务，发现系统中存在的脆弱性。

目前，脆弱性扫描器采用的是基于特征的扫描方法。基于特征的扫描又称为基于知识的扫描或者违规扫描。这种方法依据具体特征库进行判断，主要判别所搜集到的数据特征是否在脆弱性数据库中出现，所以，关键在于脆弱性特征库的规模和完善程度，而将具体脆弱性抽象成特征，其收集和分析工作量非常大，而且，资源总是不够，永远在犯罪或者安全事件后面。

第二种是基于行为进行扫描。根据使用者的行为或资源使用状况来判断是否入侵，也被称为异常检测。它首先定义一组系统“正常”情况的阈值，如 CPU 利用率、内存利用率、文件校验和等，然后将系统运行时的数值与所定义的“正常”情况比较，得出是否有被攻击的迹象。这种检测方式的核心在于如何分析所在系统的运行情况，通用性较强。它甚至有可能检测出以前未出现过的攻击方法。但因为不可能对整个系统内的所有用户行为进行全面的描述，况且每个用户的行为是经常改变的，所以它的主要缺陷在于误检率很高，尤其在用户数目众多，或工作目的经常改变的环境中。其次由于统计简表要不断更新，入侵者如果知道某系统在检测器的监视之下，他们能慢慢地训练检测系统，以至于最初认为是异常的行为，经一段时间训练后也认为是正常的了。

结束语 对计算机系统安全的研究从70年代就已经开始，并且已经取得了许多成果，但对信息系统脆弱性的研究还是一项比较新的课题，国外一些大学和研究团体在脆弱性定义、脆弱性分类、脆弱性数据库的建立和维护、脆弱性扫描器等方面取得了一些进展。大多数的脆弱性分析都是基于已有的脆弱性数据库进行的，根据从各种渠道获得的脆弱性构成一个数据库，然后采用数据分析、数据统计、数据挖掘、机器学习等手段对脆弱性数据进行归类分析。

各个厂商也对自身系统中的各种脆弱性和不断翻新的黑客攻击手段不停地提供补丁程序。

但我们认为，所有这些都没有从根本解决脆弱性，因为脆弱性是一个系统性的问题，并不是一个随时都可以添加到系统中的基本功能部件，目前普遍采用的对安全性进行改进的脆弱性解决方案并不是一个好办法，相反，应该从一开始就将安全性设计到软件中，安全性就象容错，需要有效、仔细地规划和设计，并遍布到整个系统中，从操作系统和网络协议簇的设计、结构、算法、编码等核心位置出发，对软件系统的脆弱性进行研究。

参 考 文 献

1 Aslam T. A Taxonomy of Security Faults in the UNIX Operating System. Purdue University, 1995

2 Krsul I V. Software Vulnerability Analysis. Purdue University, 1998

3 Krsul I, Spafford E, Tripunitara M. Computer Vulnerability Analysis. Purdue University, 1998

4 Bishop M. A Taxonomy of UNIX System and Network Vulnerabilities, 1995

5 Bishop M, Bailey D. A Critical Analysis of Vulnerability: [Taxonomies. Tech. Rep. CSE-96-11]. Department of Computer Science at the University of California at Davis. Sep. 1996

6 Bishop M, Dilger M. Checking for Race Conditions in File Accesses, CSE-95-10, Sep. 1995

7 Schumacher M, Haul C, Hurler M. Data Mining in Vulnerability Database. March, 2000

8 Conry-Murray A. Vulnerability Assessment Tools, Network Magazine, Apr. 2001

9 Security Problems in the TCP/IP Protocol Suite, AT&T Bell Laboratories Murray Hill, New Jersey

10 Ziemba G, Reed D, Traina C P. RFC-1858 Security Considerations for IP Fragment Filtering, Network Working Group, Oct. 1995

11 GJB 2256-94：军用计算机安全术语. 1994