网络安全

安全技巧（五）

小未 主编

目　录

网络漏洞扫描器的设计与实现

漏洞扫描器是一种自动检测远程或本地主机安全性弱点的程序。通过使用漏洞扫描器，系统管理员能够发现所维护的 Web 服务器的各种 TCP 端口的分配、提供的服务、Web 服务软件版本和这些服务及软件呈现在 Internet 上的安全漏洞。从而在计算机网络系统安全保卫战中做到“有的放矢”，及时修补漏洞，构筑坚固的安全长城。

1．引言

随着科学技术的飞速发展，21 世纪的地球人已经生活在信息时代。20 世纪人类两大科学技术成果--计算机技术和网络技术，均已深入到人类社会的各个领域，Internet 把"地球村"的居民紧密联系在一起，"天涯若比邻"已然成为现实。互联网之所以能这样迅速蔓延，被世人接受，是因为它具备特有的信息资源。无论对商人、学者，还是对社会生活中的普通老百姓，只要你进入网络的世界，就能找到其隐藏的奥妙，就能得到你所需要的价值，而这其中种种的人类社会活动，它们的影响又是相互的。近年来 Internet 的迅速发展，给人们的日常生活带来了全新的感受，"网络生存"已经成为时尚，同时人类社会诸如政治、科研、经济、军事等各种活动对信息网络的依赖程度已经越来越强，"网络经济"时代已初露端倪。

然而，网络技术的发展在给我们带来便利的同时也带来了巨大的安全隐患，尤其是 Internet 和 Intranet 的飞

速发展对网络安全提出了前所未有的挑战。技术是一把双刃剑，不法分子试图不断利用新的技术伺机攻入他人的网络系统，而肩负保护网络安全重任的系统管理员则要利用最新的网络技术来防范各种各样的非法网络入侵行为。事实已经表明，随着互连网的日趋普及，在互连网上的犯罪活动也越来越多，特别是 Internet 大范围的开放以及金融领域网络的接入，使得越来越多的系统遭到入侵攻击的威胁。但是，不管入侵者是从外部还是从内部攻击某一网络系统，攻击机会都是通过挖掘操作系统和应用服务程序的弱点或者缺陷来实现的，1988 年的"蠕虫事件"就是一个很好的实例。目前，对付破坏系统企图的理想方法是建立一个完全安全的没有漏洞的系统。但从实际上看，这根本是不可能的。美国 Wisconsin 大学的 Miller 给出一份有关现今流行操作系统和应用程序的研究报告，指出软件中不可能没有漏洞和缺陷。因此，一个实用的方法是，建立比较容易实现的安全系统，同时按照一定的安全策略建立相应的安全辅助系统，漏洞扫描器就是这样一类系统。就目前系统的安全状况而言，系统中存在着一定的漏洞，因此也就存在着潜在的安全威胁，但是，如果我们能够根据具体的应用环境，尽可能地早地通过网络扫描来发现这些漏洞，并及时采取适当的处理措施进行修补，就可以有效地阻止入侵事件的发生。因此，网络扫描非常重要和必要。

2．漏洞扫描器概述

漏洞扫描器是一种自动检测远程或本地主机安全性弱点的程序。通过使用漏洞扫描器，系统管理员能够发现所维护的 Web 服务器的各种 TCP 端口的分配、提供的服务、Web 服务软件版本和这些服务及软件呈现在 In

ternet 上的安全漏洞。从而在计算机网络系统安全保卫战中做到"有的放矢"，及时修补漏洞，构筑坚固的安全长城。

按常规标准，可以将漏洞扫描器分为两种类型：主机漏洞扫描器（HostScanner）和网络漏洞扫描器（NetworkScanner）。主机漏洞扫描器是指在系统本地运行检测系统漏洞的程序，如著名的 COPS、tripewire、tiger 等自由软件。网络漏洞扫描器是指基于 Internet 远程检测目标网络和主机系统漏洞的程序，如 Satan、ISSInternetScanner 等。

本文针对目前 TCP/IP 网络和各种网络主机的安全现状，设计并实现了一个网络漏洞扫描器，在实际使用中取得了很好的效果。

3．网络漏洞扫描器的设计

3.1 网络漏洞扫描器的总体结构

我们设计的漏洞扫描器基于浏览器/服务器（B/S）结构，整个扫描器实现于一个 Linux、UNIX 和 Windows 操作系统相混合的 TCP/IP 网络环境中，其总体结构如图 1 所示，其中运行 Linux 的工作站作为发起扫描的主机（称为扫描主机），在其上运行扫描模块和控制平台，并建有漏洞库。扫描模块直接从扫描主机上通过网络以其他机器为对象（称为目标主机，其上运行的操作系统可以是 UNIX、Linux、Windows2000/NT 等）进行扫描。而控制平台则提供一个人机交互的界面。

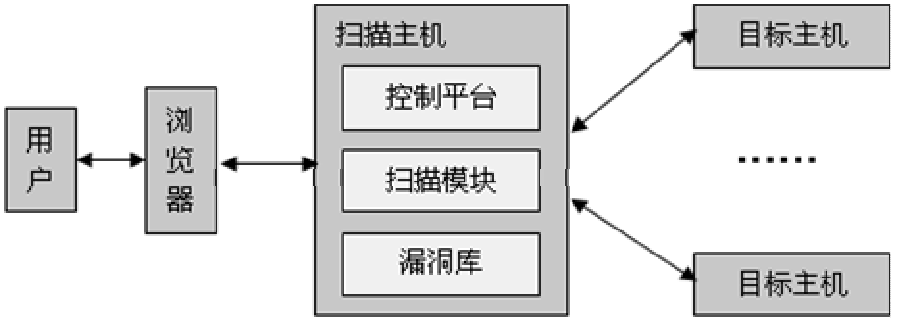

图 1 网络漏洞扫描器的总体结构

3.2 网络漏洞扫描器的扫描原理和工作原理

网络漏洞扫描器通过远程检测目标主机 TCP/IP 不同端口的服务，记录目标给予的回答。通过这种方法，可以搜集到很多目标主机的各种信息（例如：是否能用匿名登陆，是否有可写的 FTP 目录，是否能用 Telnet，httpd 是否是用 root 在运行）。在获得目标主机 TCP/IP 端口和其对应的网络访问服务的相关信息后，与网络漏洞扫描系统提供的漏洞库进行匹配，如果满足匹配条件，则视为漏洞存在。此外，通过模拟黑客的进攻手法，对目标主机系统进行攻击性的安全漏洞扫描，如测试弱势口令等，也是扫描模块的实现方法之一。如果模拟攻击成功，则视为漏洞存在。

在匹配原理上，该网络漏洞扫描器采用的是基于规则的匹配技术，即根据安全专家对网络系统安全漏洞、黑客攻击案例的分析和系统管理员关于网络系统安全配置的实际经验，形成一套标准的系统漏洞库，然后再在此基础之上构成相应的匹配规则，由程序自动进行系统漏洞扫描的分析工作。

所谓基于规则是基于一套由专家经验事先定义的规则的匹配系统。例如，在对 TCP80 端口的扫描中，如果

发现/cgi-bin/phf 或/cgi-bin/Count.cgi，根据专家经验以及CGI 程序的共享性和标准化，可以推知该 WWW 服务存在两个 CGI 漏洞。同时应当说明的是，基于规则的匹配系统也有其局限性，因为作为这类系统的基础的推理规则一般都是根据已知的安全漏洞进行安排和策划的，而对网络系统的很多危险的威胁是来自未知的安全漏洞，这一点和 PC 杀毒很相似。

实现一个基于规则的匹配系统本质上是一个知识工程问题，而且其功能应当能够随着经验的积累而利用，其自学习能力能够进行规则的扩充和修正，即是系统漏洞库的扩充和修正。当然这样的能力目前还需要在专家的指导和参与下才能实现。但是，也应该看到，受漏洞库覆盖范围的限制，部分系统漏洞也可能不会触发任何一个规则，从而不被检测到。

整个网络扫描器的工作原理是：当用户通过控制平台发出了扫描命令之后，控制平台即向扫描模块发出相应的扫描请求，扫描模块在接到请求之后立即启动相应的子功能模块，对被扫描主机进行扫描。通过对从被扫描主机返回的信息进行分析判断，扫描模块将扫描结果返回给控制平台，再由控制平台最终呈现给用户。

3.3CGI 的应用

整个漏洞扫描系统利用了浏览器/服务器（B/S）架构，目的是为了消除由于操作系统平台的不同而给程序的运行带来的差异，还为了能利用 HTML 提供的一系列功能，如超文本功能、灵活的版面编辑功能来构建一个美观灵活的人机接口。在该网络漏洞扫描器的实现中，我们通过 CGI 技术来连接前台的浏览器和后台的扫描程序。

CGI 是通用网关接口，作为一种规范，它允许 Web 服务器执行其他程序并将它们的输出以相应的方式储存在发给浏览器的文本、图形和音频中。CGI 程序能够提供从简单的表单处理到复杂的数据库查询等各种功能，这大大增强了 Web 的动态处理能力和交互能力。服务器和 CGI 程序相结合能够扩充和自定义 WorldWideWeb 的能力。

CGI 过程的主要步骤如下：

浏览器将 URL 的第一部分解码并联系服务器；

浏览器将 URL 的其余部分提供给服务器；

服务器将 URL 转换成路径和文件名；

服务器意识到 URL 指向一个程序，而非一个静态的文件；

服务器准备环境变量，执行 CGI 程序；

程序执行，读取环境变量和 STDIN；

程序为将来的内容向 STDOUT 发送正确的 MIME 头信息；

程序向 STDOUT 发送其输出的其余部分，然后终止；

服务器发现程序终止，关闭与浏览器的连接；

浏览器从程序中显示输出。

STDIN 和 STDOUT 是标准输入和标准输出的助记符。对 Web 服务器，STDOUT 送至 CGI 程序的 STDIN，程序的 STDOUT 反馈回服务器的 STDIN。在激活具有 POST 方法的 CGI 程序时，服务器使用它的 STDOUT；对于 GET 方法，服务器不使用 STDOUT。两种情况下，服务器都要求 CGI 程序通过 STDOUT 返回信息。在我们的程序中选择了 POST 方法。

3．网络漏洞扫描器的实现

3.1 扫描模块的实现

整个网络漏洞扫描器的核心部分是扫描模块，它是由很多子模块组成的，其结构如图 2 所示。

3.1.1 基本信息探测子模块的实现

在设计时加入该模块的目的是在调用扫描主模块之前探测主机是否在线，以避免不必要的空扫描。该模块的实现原理和常用的 ping 命令相似，方法是向目标主机发送 ICMP 回显报文请求，根据返回值来分析判断主机是否在线。所有安装了 TCP/IP 协议的在线网络主机，都会对这样的 ICMP 回显报文请求给与答复。虽然现在有些主机装了个人防火墙，可以屏蔽掉这样的 ICMP 回显报文请求，但是我们这个扫描系统的对象是提供网络服务的网络主机，而这样的主机是不应该屏蔽掉 ICMP 回显报文请求的，因为这样会让一些用户误认为该主机不在线，从而丧失了作为网络服务器的意义。为了降低网络拥塞导致丢包的可能性，在实现中重复四次向目标主机发送 ICMP 回显请求包。

该模块不只探测主机是否在线，而且能根据 ICMP 回显应答报文的 TTL（TTL 是位于 IP 首部中的生存时间字段）值来粗略分辨出目标主机操作系统，为下一步的扫描提供依据，特别是在扫描模块的调用和漏洞库的选择上。

该模块在实现中和其他模块不同的一个最大特点是：其他扫描模块是针对应用层的，用一般的套接字即能完成网络连接；而该模块是针对于网络层的，使用一种叫原始套接字的技术来实现。原始套接字（rawsocket）提供了一些使用 TCP 和 UDP 套接字不能实现的功能：

可以访问 ICMP 和 IGMP 等协议的数据包，可以读写内核不处理的 IP 数据包，可以创建自定义的 IP 数据首部。使用原始套接字可以编写基于 IP 协议的高层网络协议。

3.1.2 端口扫描子模块的实现

当基本信息探测子模块得知目标主机在线时，端口扫描子模块即被调用。该模块将根据传来的参数相应的扫描 TCP 的 1~1024 或者 1~65535 端口。扫描方式是利用 TCP 的完全连接方式，即利用 TCPconnect 扫描技术来设计扫描模块，这是最基本的 TCP 扫描。通常通过调用套接口函数 connect()连接到目标计算机上，完成一次完整的三次握手过程。如果端口处于侦听状态，那么 connect()就能成功返回。否则，这个端口不可用，即没有提供服务。这个技术的一个最大的优点是不需要任何权限。系统中的任何用户都有权利使用这个调用。另外的一个优点就是比其他扫描方式（如 SYN 扫描和 FIN 扫描等等）更稳定可靠。但这种方法的一个缺陷是：扫描方式不隐蔽。通常作为一个扫描器软件的应用，TCP 的 connect 会重复且大量地被集中使用，在被扫描的一端则会很容易发现这种扫描行为，目标计算机的 log 文件会显示一连串的连接和连接是否出错的服务消息，并且能很快地使它关闭；而且大多数防火墙也能屏蔽这种扫描，随着防火墙技术的快速发展，其他的一些曾经被认为是很隐蔽的扫描方式也可能被防火墙发现并屏蔽掉。所以相对而言，TCPconnect 扫描方式的这个缺陷已经被淡化了。而且，我们开发的扫描系统是从系统管理员的角度出发，因此上述的问题都是不存在的，除非他/她非法扫描他人网站主机。

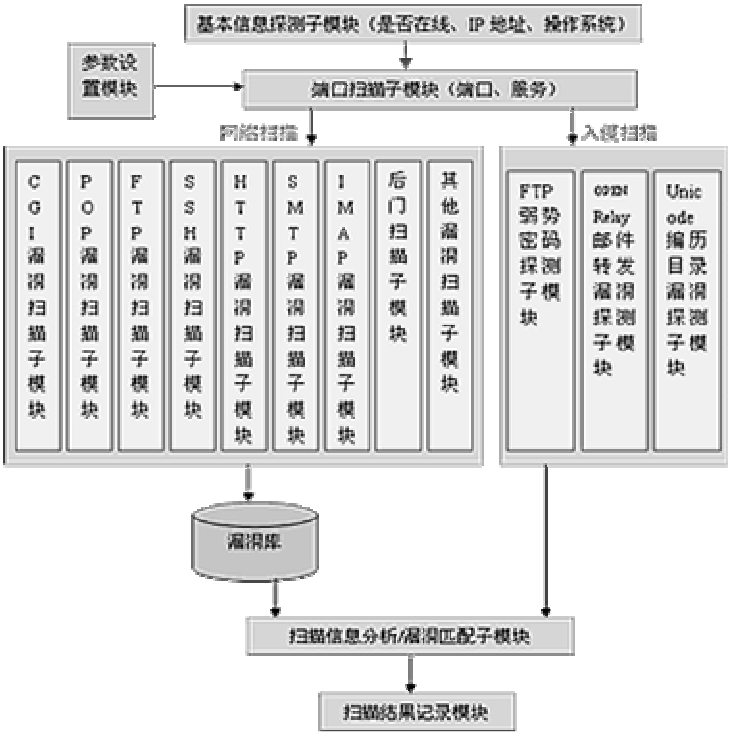

图 2 扫描模块的结构

网络扫描是个集中的、重复的行为，显而易见，它也是个比较耗资源的行为--不光是耗费扫描主机的资源，也耗费被扫描主机的资源；不光占用主机的资源，同时也占用网络相当多的资源。总体上看是个特别耗时的过程。在该网络扫描器的设计和实现的过程中，我们从两方面进行了优化：利用非阻塞连接技术和多进程技术。其结果是，首先，这两种技术的运用明显地加快了扫描的速度；而且，用多进程来实现高效率的利用了资源，从而达到了节省资源的功效。

网络漏洞扫描是建立在端口扫描的基础之上的。从黑客攻击行为的分析和收集的漏洞来看，绝大多数都是针对某一个网络服务,也就是针对某一个特定的端口的。

所以漏洞扫描也是以同样的思路来进行的。而如今大多数国内国外的扫描器把端口扫描和具体的漏洞扫描分开来，相互之间几乎没有什么联系：端口扫描的目的就是为了向用户报出当前所开的端口和网络服务，以及看是否有些特定的后门存在；漏洞扫描则完全是另外一个独立的流程，不管目标主机的相应端口及服务是否打开，都要做一系列的扫描。这样看起来好像扫得很全面很彻底，但很明显的一点是，如果在对目标主机毫无了解的情况下，比如说是一台最普通的、几乎没有提供任何网络服务的机器，此时对它也进行彻底的漏洞扫描可以说是意义甚微的，甚至可以说在某些时候会对系统及网络有些反面的影响。

而我们在设计实现该漏洞扫描系统时则从另一个角度出发，保证在达到同等目的的前提下，尽量少占用网络、主机以及时间资源，提高资源的利用率和扫描系统的效率。基本思想是：避免不必要的模块调用，根据不同的实际情况来调用相应的扫描子模块。

在实现中，所有子模块的运行都是和端口扫描的主流程同时进行的。在此，我们利用了多进程技术来实现并发。进程是具有一定功能的程序，是关于一个数据集合的一次运行活动，它是程序运行的基本单位。在传统的 UNIX 模型中，当一个进程需要由另一个实体执行某些操作时，该进程派生（fork）一个子进程，让子进程去进行处理。此时子进程与父进程是完全独立的两个运行实体，以这样的方式可以实现并行。现在在 UNIX/Linux 系统中也可以用线程来实现程序的并行执行。和线程比起来，fork 子进程存在以下两个问题：

fork 的代价是昂贵的：内存映像要从父进程拷贝到

子进程，所有描述字要在子进程中复制等等。目前的实现使用一种称为写时拷贝（copy-on-write）的技术，可避免父进程数据空间向子进程的拷贝，除非子进程需要自己的拷贝。尽管有这种优化技术，fork 仍然是很昂贵的。

fork 子进程执行后，需要用进程间通信（IPC）在父子进程之间传递信息。fork 之前的信息容易传递，因为子进程从一开始就有父进程数据空间以及所有描述字的拷贝。从子进程返回信息给父进程则需要做更多的工作。

在我们的扫描系统的设计中，所需要的并发执行模块数并不是很多，对于现有的硬件设备来讲，创建十来个子进程的代价根本算不上什么，这样 fork 子进程的第一种缺陷的影响几乎就不存在了；关于它的第二种问题其实对我们的实现也没有影响，因为这些扫描子模块之间以及子模块和主模块之间的交流除了共享文件之外，避免了其它的通信。而且从实现上来说，线程使用比较复杂，采用 fork 子进程会使开发和调试相对简单易行些。

3.1.3 入侵扫描的实现

从图 2 中可以看出，由于扫描方式的不同，可以将端口扫描子模块分为两大类：网络扫描和入侵扫描。与网络扫描不同，入侵扫描没有对应的漏洞库，本节简单介绍三类入侵扫描子模块的实现方式。

FTP 弱势密码探测子模块主要是对一些可能存在的用户名作弱势密码的探测。其实现方法主要就是模拟客户端的用户协议解释器的功能，和服务器端建立连接并发送一系列命令和处理相应的应答，以此模拟登录，从而判断是否存在有弱势密码的账号。

FTP 命令和应答在客户和服务器的控制连接上是以

NVTASCII 码形式传送的。这些命令都是 3 或 4 个字节的大写 ASCII 字符，其中一些带选项参数。从客户向服务器发送的 FTP 命令超过 30 种，如 ABOR（放弃先前的 FTP 命令和数据传输）、QUIT（从服务器上注销）、SYST（服务器返回系统类型）等。应答都是 ASCII 码形式的 3 位数字，并跟有报文选项。其原因是软件系统需要根据数字代码来决定如何应答，而选项是面向人工处理的。在我们的子模块中是根据应答中的数字代码来判断的，如 125（数据连接已经打开，传输开始）、331（用户名就绪，要求输入口令）、501（语法错误--无效参数）等。

OPENRelay 邮件转发漏洞是针对 SMTP 网络协议的。用 TCP 进行的邮件交换是由报文传送代理 MTA（MessageTransferAgent）完成的。最普通的 Unix 系统中的 MTA 是 Sendmail。客户端 MTA 通过访问服务器端的 25 号端口来和服务器端的 MTA 通信。两个 MTA 之间也用 NVTASCII 进行通信。客户向服务器发出命令，服务器用数字应答码和可选的可读字符串进行响应。这与 FTP 非常类似。关于服务器端的应答也与 FTP 非常类似，在此就不赘述了。

OPENRelay 邮件转发漏洞探测子模块也是通过发送一系列的命令和分析处理一系列的应答，以此模拟客户端 MTA 的功能来通过服务器连续发送一定数量的邮件，从而判断服务器是否没有屏蔽掉 OPENRelay 的功能。

Unicode 遍历目录漏洞探测子模块是针对 80 端口的 WWW 网络服务的。WWW 服务遵循的是 HTTP 协议，所以 HTTP 协议的一些特点决定了该子模块的实现。Un

icode 遍历目录漏洞的扫描是对 IIS 服务器软件的编码机制存在的漏洞进行具体的探测。不同的测试漏洞的特殊编码加上命令后，被封装在 HTTP 协议的请求中，向目标主机的 80 端口发送,并靠分析返回的状态码分析这种编码是否绕过了 IIS 的路径检查，可以执行相应的命令了。在 Unicode 遍历目录漏洞探测子模块的实现中，所采用的请求方法是 GET 方法。

3.1.4 扫描模块的流程

扫描模块在工作时，首先进行初始化，在初始化阶段，主要是读取所需的参数。比如从基本信息探测子模块得到操作系统类型，由此来决定在扫描中需要使用的漏洞库；还有一些用户自己配置的参数。除了读取参数外，还要建立一些文件以供以后使用。

初始化后，建立非阻塞 socket 并连接，然后根据得到的相关端口及对应的服务，来调用相应的漏洞扫描子模块（包括 CGI 漏洞扫描子模块、POP 漏洞扫描子模块、FTP 漏洞扫描子模块、SSH 漏洞扫描子模块、HTTP 漏洞扫描子模块、SMTP 漏洞扫描子模块和 IMAP 漏洞扫描子模块以及三种类型的入侵扫描子模块）,当所有的端口都已经扫描完以后，调用后门扫描子模块和其他漏洞扫描子模块。由于这两个扫描子模块并不是针对某一个端口或网络服务的，所以没有像其他的漏洞扫描子模块一样在端口扫描过程中被调用。但有一点是和其他子模块是一致的，即这两个子模块的调用形式也是通过创建子进程来完成的。

3.2 漏洞库的建立

一个网络漏洞扫描系统的灵魂就是它所使用的系统漏洞库，漏洞库信息的完整性和有效性决定了扫描系统

的功能，漏洞库的编制方式决定了匹配原则，以及漏洞库的修订、更新的性能，同时影响扫描系统的运行时间。

3.2.1 设计原则

通过比较分析和实验分析，在漏洞库的设计中，我们遵循了以下几条原则：

从漏洞库的简易性、有效性出发，选择以文本方式记录漏洞。这样易于用户自己对漏洞库进行添加配置。因此在我们提供漏洞库升级的基础上，又多出了一条途径更新漏洞库，以保持漏洞库的实时更新，也使得漏洞库可以根据不同的实际环境而具有相应的特色。

对每个存在安全隐患的网络服务建立对应的漏洞库文件。一般情况下一个网络服务对应两个漏洞库，一个是针对 UNIX 的漏洞库，另一个则是针对 WindowsNT/2000 的漏洞库。

对漏洞危险性进行分级。这有助于系统管理员了解漏洞的危险性，从而决定所要采取的措施。

提供漏洞危害性描述和建议的解决方案。有些扫描器，虽然扫描漏洞的功能强大，但不提供详细描述和解决方案（如著名的扫描器 ISSInternetScanner）。这往往导致系统管理员对检测出的漏洞没有足够的重视，或不了解它的危害也不知如何修补漏洞而暂置一旁。因此，漏洞扫描器要真正发挥它的预警及预防的作用，真正成为系统管理员的有效助手，就应当提供对漏洞的具体描述和有效的解决方案。

3.2.2 漏洞分级原则

在对黑客攻击行为分析的基础上，又借助了一些资深的系统管理员的经验，我们对漏洞进行了比较粗略的分级。将漏洞按其对目标主机的危险程度分为三级，即

A 级、B 级和 C 级。A 级漏洞是允许恶意入侵者访问并可能会破坏整个目标系统的漏洞，如允许远程用户未经授权访问的漏洞。A 级漏洞是威胁最大的一种漏洞，大多数 A 级漏洞是由于较差的系统管理或配置有误造成的。同时，几乎可以在不同的地方，在任意类型的远程访问软件中都可以找到这样的漏洞。如，FTP、gopher、Telnet、Sendmail、finger 等一些网络程序常存在一些严重的 A 级漏洞。B 级漏洞是允许本地用户提高访问权限，并可能允许其获得系统控制的漏洞。例如允许本地用户非法访问的漏洞。网络上大多数 B 级漏洞是由应用程序中的一些缺陷或代码错误引起的。Sendmail 和 Telnet 都是典型的例子。此外，因编程缺陷或程序设计语言的问题造成的缓冲区溢出问题也是一类典型的 B 级安全漏洞。C 级漏洞是任何允许用户中断、降低或阻碍系统操作的漏洞，如拒绝服务漏洞。最典型的一种拒绝服务攻击是 SYNFLOOD，即入侵者将大量的连接请求发往目标服务器，目标主机不得不处理这些"半开"的 SYN，然而并不能得到 ACK 回答，很快服务器将用完所有的内存而挂起，任何用户都不能再从服务器上获得服务。综上所述，对网络主机危害最严重的是 A 级漏洞，其次是 B 级漏洞，而 C 级漏洞是对系统正常工作进行干扰。

3.2.3 漏洞库的实现

通过大量的多方收集，主要是对 www.cert.org、www.aucert.com、www.securefocus.com 及中国绿色联盟等反黑权威网站漏洞信息进行分类整理，我们对存在漏洞的主要的网络服务逐一建立了三级漏洞描述文件，作为各个漏洞扫描子模块的访问库体。

在每个漏洞库文件中，每条漏洞信息占一行，行首