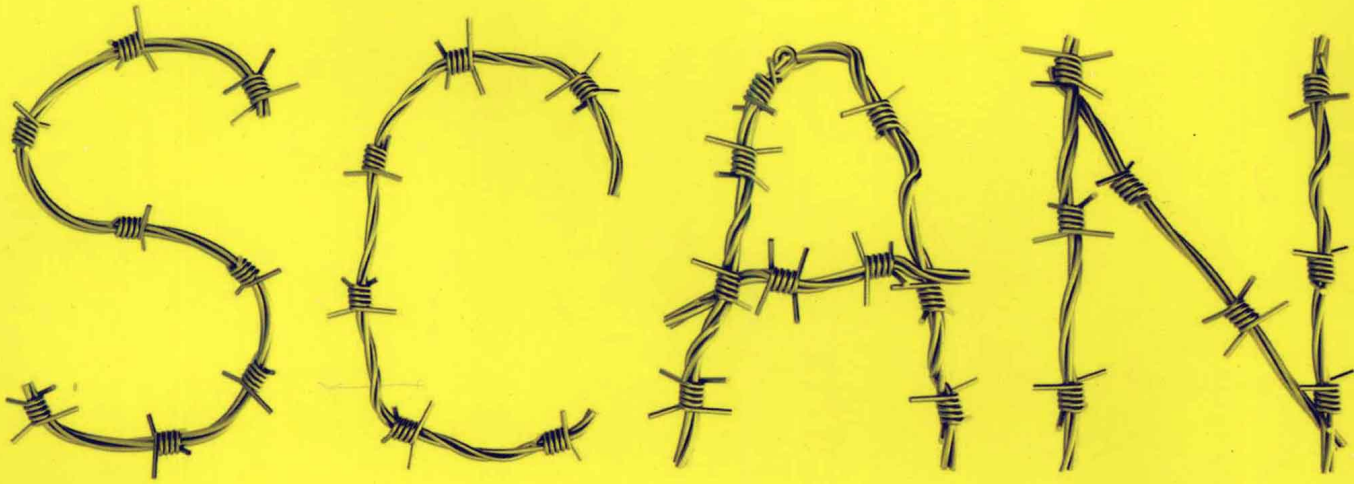
HZ BOOKS
华章科技

资深网络安全专家十余年研究与实践经验的结晶

重点突出，针对性地讲解了核心网络扫描技术的原理与最佳实践

实战性强，各种主流扫描器的设计方法和原理尽含其中，附有完整源代码可供参考

SECURITY



NETWORK SCANNING TECHNOLOGY UNLEASHED

PRINCIPLE, PRACTICE AND IMPLEMENTATION OF NETWORK SCANNER

网络扫描技术揭秘

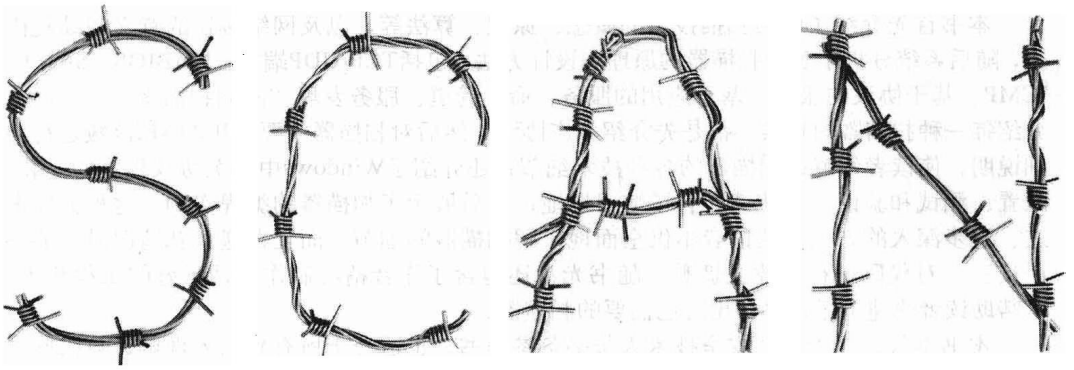
原理、实践与扫描器的实现

李瑞民 著

CD-ROM

机械工业出版社
China Machine Press

安全技術大系
SECURITY



NETWORK SCANNING TECHNOLOGY UNLEASHED

PRINCIPLE, PRACTICE AND IMPLEMENTATION OF NETWORK SCANNER

网络扫描技术揭秘

原理、实践与扫描器的实现

李瑞民 著

机械工业出版社
China Machine Press

本书系统地介绍网络扫描器的概念、原理与设计方法，饱含作者十几年来在网络技术应用实践中不断总结的经验与技巧。作者从网络协议这样的基本概念开始，细致深入地分析了网络扫描器的原理，并用自己制作的大量工程代码，揭示了网络扫描器的实现方法与最佳实践。

本书首先介绍了网络扫描技术的概念、原理、算法等，以及网络协议的意义与编程概述，随后系统分析了各种扫描器的原理与设计方法，包括TCP/UDP端口、NetBIOS、SNMP、ICMP、基于协议的服务、基于应用的服务、命名管道、服务发现、漏洞扫描器等。书中在介绍每一种扫描器的时候，都是先介绍相应协议，然后对扫描器中要使用的API函数进行详细说明，使读者知道该扫描器的各种技术细节；还介绍了Windows中相关协议程序的安装、配置、测试和验证等，使读者有了演习场地；最后展示了扫描器的编程实例。这种循序渐进、逐步深入的方式，使读者不仅全面地了解扫描器的细节，而且在遇到新情况时，能举一反三，对代码进行修改或调整。随书光盘还包含了作者精心制作与调试好的工程代码，可帮助读者快速上手，设计出自己需要的扫描器。

本书不仅是网管员和安全技术人员必备参考书，也适合于所有想深入理解计算机网络原理、全面了解网络扫描技术的学生、教师以及安全技术爱好者。

图书在版编目（CIP）数据

网络扫描技术揭秘：原理、实践与扫描器的实现/李瑞民著. —北京：机械工业出版社，2012.1

ISBN 978-7-111-36532-7

Ⅰ.网…　Ⅱ.李…　Ⅲ.计算机网络—安全技术　Ⅳ.TP393.08

中国版本图书馆CIP数据核字（2011）第240809号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：吴　怡
北京京北印刷有限公司印刷
2012年1月第1版第1次印刷
186mm×240mm · 32印张
标准书号：ISBN 978-7-111-36532-7
　　ISBN 978-7-89433-197-7（光盘）
定价：79.00元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991，88361066
购书热线：（010）68326294，88379649，68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

计算机网络的普及，使人们越来越关注网络的安全。而网络安全又取决于网络管理员和网络用户对网络的了解，这种了解不仅包括如何使用网络应用程序，还包括网络的配置甚至网络的技术细节。

网络中，协议是任意两台主机之间进行通信的前提和始终贯彻的原则。因此本书首先在第1章中对网络扫描的概念、原理、目的、算法、历史进行了综述，然后在第2章着眼于现在网络，特别是互联网中网络协议的意义，以及如何通过协议进行编程，由浅入深，逐步分析了网络协议的各项技术细节。

网络服务是一台主机对外开放的最终目的，而端口又是服务的对外窗口。因此以前很多人对网络扫描的理解就是端口扫描。端口直接与服务对应的关系，使得端口越来越成为人们关注的目标，打开一种服务就意味着打开了一个或几个端口，同时也打开了一扇“大门”，因此在第3章中，对端口扫描进行了全面详细的论述。

协议不同，对端口的扫描也不同，因此在随后的第4、5、6章中，分别介绍了基于NetBIOS、SNMP、ICMP扫描的协议、原理、扫描器实现方法等细节。

网络服务必然存在端口，但由于端口本身不具有强制性，所以即使扫描到端口，也不意味着必然对应某一服务，因此端口扫描只能作为初判，服务扫描才是真正判断对方是否提供某服务的最终方式。在第7章中，详细地介绍了完全基于服务的扫描方式，主要有WWW服务、FTP服务、Telnet服务、E-mail服务等应用层服务。这些服务极具扫描的普遍性，可以说，有了针对这些服务的扫描器设计框架和设计思路，任何一个新协议、新服务都可以用此方法做出相应的扫描器。在第8章中，介绍了另一种基于应用的扫描方式，在这里，除了WWW服务、FTP服务之外，还包括了对网络资源的扫描。这种扫描由于使用了应用层的API，所以程序简捷、透明性强，但对底层的控制不多。

在第9、10两章中，介绍了几种特殊应用领域的扫描方式：命名管道扫描器和服务发现扫描器。这两种扫描器是较新的技术，并且可以使网络扫描从仅限于安全领域内的应用，转换到生活领域内的应用。

第11章的漏洞扫描器则是对网络中存在的漏洞进行扫描，从而发现漏洞以及漏洞的位置。这种扫描方式具有很大的实用性，但这类漏洞扫描器的编制有一定的难度，且具有较强的时效性。

第12章介绍扫描防范技术，通过这些技术，可以检测或防止别人对主机本身或网络进行

扫描。

为了统一形式，书中在介绍各扫描器时都先介绍协议，使读者对该协议有个初步的了解。然后对其API，特别是扫描器中要使用的API函数进行详细介绍，使读者知道该扫描器要实现的各种技术细节。再针对Windows中该协议程序的安装、配置、测试和验证进行详述，使读者有演习场地。最后是扫描器的编程实例。采用这种循序渐进、逐步深入的方式，读者不仅能深入全面地了解扫描器细节，也能在遇到新情况时举一反三，对代码进行修改或调整。

在全书的组织上，虽然按技术人为地分了很多章，但实际上各章内容则是你中有我，我中有你，相互解释，相互引用，建议读者先读第1、2章，第3～10章则可以按需要酌情选读，最后再看第11、12章。

纵观这些扫描器系统及扫描监测、防护系统，无论是“攻”，还是“防”，都不取决于程序的设计程度，而是取决于使用程序的人。对于黑客，会利用攻方系统刺探、入侵别人的系统，用防方系统保护自己不被发现或刺探；而对于网络普通用户或网络管理员，则要通过攻方系统扫描自身缺点，防患于未然，同时利用防方系统及时发现别人的扫描操作。再安全的系统，都有漏洞，再全面的技术，都有不足，真正的网络安全，需要丰富的安全常识和强烈的安全意识。

全书中的说明和编程实例均以Microsoft Windows XP（SP3）作为默认的操作系统、以Microsoft Visual C++ 6.0（SP6）作为默认的开发环境。

本书附带光盘，内容包括本书所有程序的源码工程文件，以及个别程序所需的必要支持文件。

声明

目　录

前　言

第 1 章　绪论 / 1

1.1　网络安全的概念 / 1

1.2　网络扫描的概念 / 2

1.2.1　服务和端口 / 2

1.2.2　网络扫描 / 4

1.3　网络扫描原理概述 / 5

1.4　扫描编程与客户端编程的区别 / 5

1.5　网络扫描的目的 / 5

1.6　网络扫描算法 / 6

1.6.1　非顺序扫描 / 6

1.6.2　高速扫描 / 8

1.6.3　分布式扫描 / 8

1.6.4　服务扫描 / 8

1.6.5　指纹识别算法 / 8

1.6.6　漏洞扫描 / 9

1.6.7　间接扫描 / 9

1.6.8　秘密扫描 / 9

1.6.9　认证扫描 / 10

1.6.10　代理扫描 / 10

1.6.11　手工扫描 / 10

1.6.12　被动扫描 / 10

1.7　网络扫描器的分类 / 11

1.8　网络扫描技术的发展史 / 12

1.8.1　手工扫描阶段 / 12

1.8.2　使用通用扫描器阶段 / 13

1.8.3　设计专用扫描器阶段 / 14

1.9　扫描器的限制 / 14

1.10　当前网络常见的漏洞 / 14

1.10.1　DOS和DDOS / 15

1.10.2　缓冲区溢出 / 15

1.10.3　注入式攻击 / 17

1.10.4　明文传输 / 17

1.10.5　简单密码 / 18

第 2 章　网络协议和网络编程例程 / 19

2.1　常用的网络编程 / 19

2.1.1　TCP/IP协议编程 / 20

2.1.2　NetBIOS/NetBEUI协议编程 / 41

2.1.3　Win Inet高层编程 / 47

2.1.4　命名管道和邮槽高层编程 / 48

2.2　扫描器中公用编程示例 / 49

2.2.1　CTreeCtrl控件的应用 / 49

2.2.2　CListCtrl控件的应用 / 51

2.2.3　INI文件的操作 / 53

2.2.4　数据库ADO的简单应用 / 56

2.2.5　IP格式的互换 / 59

2.2.6　Windows操作系统类型的判断 / 62

2.2.7 多线程的局限性和使用方式 / 63
2.2.8 VC++下Windows Socket的使用 / 66
2.2.9 网卡的混杂模式 / 69
2.3 嵌入外部程序 / 71
2.3.1 可执行外部程序的几个函数 / 71
2.3.2 编程实例：使用重定向接收外部程序运行结果 / 76
2.3.3 编程实例：使用管道接收外部程序运行结果 / 80
第 3 章 TCP/UDP端口扫描器的设计 / 85
3.1 端口扫描的概念 / 85
3.1.1 端口的概念 / 85
3.1.2 端口扫描原理 / 87
3.2 端口扫描技术 / 87
3.2.1 网络通信实例分析 / 87
3.2.2 TCP扫描 / 90
3.2.3 UDP扫描 / 92
3.3 手工扫描 / 93
3.3.1 检测单主机单端口开与否 / 93
3.3.2 检测单主机单端口是否有相应服务 / 94
3.3.3 检测多主机或多端口 / 95
3.4 编程实例：TCP端口扫描器 / 98
3.4.1 程序主界面 / 99
3.4.2 程序代码 / 100
3.5 编程实例：UDP端口扫描器 / 112
3.5.1 程序主界面 / 112
3.5.2 程序代码 / 113
第 4 章 NetBIOS扫描器的设计 / 120
4.1 NetBIOS协议的使用 / 120
4.1.1 查看和修改NetBIOS配置 / 120
4.1.2 查看NetBIOS配置的命令 / 122
4.2 IP和主机名的互换 / 127
4.2.1 主机名转IP地址 / 127
4.2.2 IP地址转主机名 / 127
4.3 MAC地址的读取 / 128
4.4 本地域名、子网掩码、网卡类型的读取 / 129
4.5 用户名、共享目录、组列表的读取 / 134
4.5.1 Unicode编程与ANSI之间的互换 / 134
4.5.2 用户名列表的读取 / 137
4.5.3 共享目录的读取 / 150
4.5.4 组列表的读取 / 156
4.5.5 远端主机时间的读取 / 159
4.5.6 远端服务支持类型的读取 / 161
4.5.7 主机信息的读取 / 163
4.6 NetBIOS的安全性 / 166
4.7 编程实例：反“IP欺骗”——MAC地址扫描器的设计 / 169
4.7.1 反“IP欺骗”的原理 / 169
4.7.2 MAC地址扫描器的主界面 / 170
4.7.3 程序代码 / 170
4.8 编程实例：NetBIOS的通用扫描器 / 176
4.8.1 程序主界面 / 176
4.8.2 程序代码 / 177

第 5 章 SNMP扫描器的设计 / 186
5.1 SNMP协议 / 186
5.1.1 管理信息结构 / 187
5.1.2 管理信息库 / 187
5.1.3 通信协议 / 191
5.2 SNMP的API / 193
5.2.1 数据类型和常用结构 / 194
5.2.2 管理程序API / 197
5.3 SNMP安装和验证 / 204
5.4 编程实例：SNMP通用读设工具 / 207
5.4.1 程序主界面 / 208
5.4.2 程序代码 / 209
5.5 编程实例：基于SNMP的主机扫描器 / 213
5.5.1 程序主界面 / 214
5.5.2 程序代码 / 214
第 6 章 ICMP扫描器的设计 / 221
6.1 ICMP协议简介 / 222
6.2 ping与tracert命令简介 / 222
6.2.1 ping程序使用 / 222
6.2.2 tracert程序使用 / 224
6.3 ICMP通信实例分析 / 226
6.4 ICMP协议内容 / 227
6.4.1 目的不可达消息 / 227
6.4.2 超时消息 / 228
6.4.3 参数问题消息 / 229
6.4.4 源拥塞消息 / 229
6.4.5 重定向消息 / 230
6.4.6 回送请求或回送响应消息 / 231
6.4.7 时间戳请求和时间戳响应消息 / 231
6.4.8 信息请求或信息响应消息 / 232
6.5 ICMP扫描的安全性 / 233
6.6 编程实例：快速多IP的ICMP扫描器 / 234
6.6.1 程序主界面 / 234
6.6.2 程序原理 / 237
6.6.3 程序代码 / 238
第 7 章 基于协议的服务扫描器的设计 / 250
7.1 WWW服务扫描 / 251
7.1.1 WWW服务器架构 / 251
7.1.2 协议消息格式 / 254
7.1.3 WWW服务器的安装与配置 / 260
7.2 编程实例：WWW服务扫描器 / 264
7.2.1 扫描原理 / 265
7.2.2 程序主界面 / 266
7.2.3 程序代码 / 266
7.3 FTP服务扫描 / 272
7.3.1 FTP简介 / 272
7.3.2 FTP服务器的安装与配置 / 274
7.4 编程实例：FTP服务扫描器 / 278
7.4.1 程序主界面 / 278
7.4.2 程序代码 / 278
7.5 Telnet服务扫描 / 281
7.5.1 Telnet协议简介 / 281
7.5.2 Telnet的安装与配置 / 284
7.6 编程实例：Telnet服务扫描器 / 286
7.6.1 程序主界面 / 287
7.6.2 程序代码 / 287
7.7 Email服务扫描 / 291
7.7.1 电子邮件协议简介 / 291
7.7.2 电子邮件服务器的安装与配置 / 299

7.8 编程实例：Email服务扫描器 / 306
7.8.1 程序主界面 / 306
7.8.2 程序代码 / 307
第 8 章 基于应用的服务扫描器的设计 / 314
8.1 Win Inet编程接口 / 314
8.1.1 CInternetSession类 / 315
8.1.2 CInternetConnection类 / 322
8.1.3 CHttpConnection类 / 323
8.1.4 CFtpConnection类 / 324
8.1.5 CInternetFile类 / 237
8.1.6 CInternetException类 / 329
8.2 编程实例：基于应用的WWW服务扫描器 / 329
8.3 编程实例：基于应用的FTP服务扫描器 / 330
8.4 网络资源协议 / 332
8.4.1 NETRESOURCE结构 / 332
8.4.2 WNetOpenEnum函数 / 333
8.4.3 WNetEnumResource 函数 / 334
8.4.4 WNetCloseEnum函数 / 335
8.5 编程实例：网络资源扫描器 / 336
8.5.1 程序主界面 / 336
8.5.2 程序代码 / 337
第 9 章 命名管道扫描器的设计 / 341
9.1 命名管道 / 341
9.2 命名管道API / 342
9.2.1 命名管道的UNC格式 / 342
9.2.2 命名管道编程的API / 342
9.3 命名管道编程示例 / 349
9.3.1 命名管道服务器端 / 349
9.3.2 命名管道客户端 / 350
9.4 邮槽 / 352
9.4.1 邮槽的UNC格式 / 352
9.4.2 邮槽编程的API / 352
9.5 邮槽编程示例 / 354
9.5.1 邮槽服务器端编程 / 354
9.5.2 邮槽客户端编程 / 355
9.6 编程实例：SQL Server命名管道扫描器的设计 / 356
9.6.1 Microsoft SQL Server 简介 / 356
9.6.2 程序主界面 / 359
9.6.3 程序代码 / 360
第10章 服务发现扫描器的设计 / 364
10.1 服务发现简介 / 364
10.2 UPnP协议 / 365
10.2.1 寻址 / 367
10.2.2 发现 / 367
10.2.3 描述 / 368
10.2.4 控制 / 369
10.2.5 事件 / 369
10.2.6 展示 / 370
10.3 XML协议 / 371
10.4 SSDP协议分析实例 / 373
10.4.1 设备类型 / 374
10.4.2 协议消息格式 / 377
10.5 编程实例：服务发现扫描器 / 381
10.5.1 程序主界面 / 382
10.5.2 程序代码 / 383
第11章 漏洞扫描器的设计 / 395
11.1 注入式漏洞扫描器 / 395
11.1.1 SQL注入式攻击原理 / 396
11.1.2 注入式攻击的局限性 / 398

11.1.3 单机模式或C/S模式的攻击 / 398
11.1.4 B/S模式下扫描程序设计 / 401
11.2 主机弱密码扫描 / 412
11.2.1 网络连接的API / 412
11.2.2 密码穷举分析 / 416
11.2.3 程序主界面 / 418
11.2.4 程序代码 / 419
11.3 DOS/DDOS攻击 / 425
11.3.1 程序主界面 / 427
11.3.2 程序代码 / 427
11.4 明文密码嗅探 / 432
11.4.1 程序主界面 / 433
11.4.2 程序代码 / 434
11.5 端口对照 / 443
11.5.1 程序主界面 / 443
11.5.2 程序代码 / 445
第12章 扫描防范技术的研究 / 451
12.1 更换端口 / 452
12.2 预留陷阱技术 / 453
12.3 基于哨兵的端口扫描监测 / 454
12.3.1 程序主界面 / 455
12.3.2 程序代码 / 456
12.4 基于嗅探的端口扫描监测及DDOS拒绝服务监测 / 460
12.4.1 程序主界面 / 461
12.4.2 程序代码 / 462
12.5 实时监测本地所有TCP/UDP连接及端口 / 467
12.5.1 程序主界面 / 467
12.5.2 结构与函数API / 468
12.5.3 程序代码 / 471
12.6 如何关闭端口 / 478
12.6.1 FTP端口 / 478
12.6.2 WWW端口 / 480
12.6.3 Telnet端口 / 480
12.6.4 NetBIOS端口 / 481
附录A 本书容易混淆概念解析 / 482
附录B Windows Socket错误返回码 / 486
附录C Win Inet错误返回码 / 491
附录D HTTP错误返回码 / 493
参考文献 / 498
后记 / 499

第1章 绪 论

网络在人们生活中占据了越来越重要的位置，人们以各种方式访问网络，在享受网络带来的方便的同时，也越来越多地感受到网络安全所潜在的和已带来的问题。

作为一个用户，自己的主机是否安全；作为网络管理员，所管理和维护的网络是否安全，解决这些问题除了及时打上操作系统的最新补丁以及安装防火墙和防病毒软件之外，是否还有别的方法呢？答案很简单，就是以一个访问者的角度，全面审视一下自己的网络有什么，没有什么，还有哪些缺陷或漏洞，这正是网络扫描的主要功能之一。

网络扫描的使用者不同，扫描的目的和范围也不同，如前所述，用户和网络管理员所关注的扫描目的和范围就有很大的不同。扫描的目的不同，扫描的原理当然也不相同，同时，客观上由于扫描程序所在的主机、要扫描的主机不同，操作系统不同，配置不同，而使用的扫描器、扫描方式也会不尽相同。正因为此，可以发现网络扫描是一个综合的技术领域，当然，根据实际情况，具体问题可以具体分析；同时，也可以看出任何一项扫描技术都具有一定的局限性。

网络扫描不是自网络出现就专门设计的一种应用服务，而是当网络发展到一定阶段后，不同程序设计者根据不同的需求而产生的。同时，由于网络扫描不是一种通用的应用服务，所以并没有产生相应的行业标准。纵观整个网络扫描史，可以分为手工扫描、使用通用扫描工具、设计专用扫描工具三个阶段。

最后，本章对当前仍然广泛存在的几个与网络扫描有关的漏洞进行了简述。

1.1 网络安全的概念

网络安全是指网络中的硬件系统、软件系统及其系统中涉及的数据因受到预设的保护，而可以连续、可靠、正常地运行，并不因偶然事故或者恶意处理而遭受破坏、非法更改、信息泄漏。网络安全从其本质上来讲就是网络上的信息安全和服务安全，从广义来说，凡是涉及网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论都是网络安全的研究领域。网络安全涉及计算机技术、网络技术、通信技术等多种技术，是一门汇集信息论、密码学、数论等多种学科的综合性学科。

网络安全更多看重的是应用实践，因为实践才是检测网络安全的唯一标准。同时，由于网络安全涉及网络的所有方面，故网络安全的关键在于培养网络用户的安全意识，只有用户具备良好的安全意识，各网络安全工具才能更好地结合，产生最佳效果。

1.2 网络扫描的概念

网络扫描是根据对方服务所采用的协议，在一定时间内，通过自身系统对对方协议进行特定读取、猜想验证、恶意破坏，并将对方直接或间接的返回数据作为某指标的判断依据的一种行为。在上述定义中，网络扫描的概念包含以下意思：

- 网络扫描器几乎全部是客户端一方的程序，所针对的对象绝大多数是服务器方。
- 网络扫描通常是主动的行为，绝大多数网络扫描器的扫描行为都是在或希望在服务器不知情的情况下偷偷进行，通常在扫描器的设计中，扫描行为应尽可能地避免被服务器察觉。所以扫描器通常不会对被扫描的主机有过多的要求，只能主动适应服务器的各项要求。
- 网络扫描通常具有时限性。该时限虽然没有一个明确的界限，但一般来说都是接近扫描的最快速度。如果某个用户每隔几个小时访问一下公司的网站主页，则不能算是扫描。
- 扫描几乎都是要用工具进行，因为操作系统提供的程序并不都具有扫描的各项要求。
- 扫描的目的一般是，对预先的猜想进行验证或采集一些关心的数据。
- 不可回避的一点就是，网络扫描更多地被黑客用于选择攻击目标和实施攻击，并且由于扫描自身的特点，通常被认为是网络攻击的第一步。

1.2.1 服务和端口

生活中的“服务”是指为他人做事，并使他人从中受益的一种有偿或无偿的活动，该活动通常不以实物形式，而是以提供“活劳动”（指物质资料的生产过程中劳动者的脑力和体力的消耗过程）的形式满足他人某种特殊需要。

在网络中，“服务”是指某主机按预先定义的协议和一些国际标准、行业标准，向其他主机提供某种数据的支持，并且称服务提供者为“服务器”（Server），称服务请求者为“客户端”（Client）。与生活中的服务相比，网络上的服务更强调的是协议，即双方必须具有相同的协议，才能进行交流。

一台主机可以安装多个服务，这些服务可以是相同的服务，也可以是不同的服务。为了区分这些服务，引入“端口”（Port）这个概念，即每一个服务对应于一个或多个端口。端口具有独占性，一旦有服务占用了某个端口，则通常情况下，另外的服务不能再占用这个端口。

根据Berkeley套接字的约定，端口名称用一个2字节（16 位）的无符号整数来表示，范围为：0~65535，共65536个。其中，端口名称在0到1023之间的端口习惯上称为“熟知端口”（well-known port），主要用于一些公用的并得到国际组织IANA（The Internet Assigned Numbers Authority，互联网数字分配机构）公认的服务；端口名称在1024至49151之间的端口称为“登记端口”，主要用于服务类，而又不属于熟知端口的程序使用；端口名称在49152至65535之间的端口称为“临时端口”，是指任何程序都可以临时使用的端口。原则上，1024至65535之间的端口，只要不出现冲突，用户程序可以根据情况随时使用。

有关端口更详细的说明，参见3.1.1节“端口的概念”。

需要说明的是，由于习惯问题和历史原因，有很多术语产生了混淆。比如在计算机的整机设计和生产时，常常将主机区分为“服务器”（Server）和“工作站”（Work Station），二者从整体设计、部件生产、整体检测方式上都不同。比如服务器通常是无人值守，并需要每周7×24小时正常运转，所以优先考虑的是CPU性能、内存容量、网络吞吐量；而工作站则主要是由用户来操作的，因此更多关注的是显示效果、音频效果等方面。

这种和工作站所对应的、硬件上的服务器（下面称为主机）与上述中和客户端相对应的服务器（下面称为服务端或服务器端）没有必然的对应关系。即TCP/IP协议虽然规定了各个协议的实现细节，但却并没有硬性地规定应用程序应如何与协议软件进行交互。因此实际上所说的服务端和主机之间没有一个一一对应关系。主机是一个硬件的概念，是一台物理设备，而服务端是指一组软件系统，一台主机上可以装多个服务器软件来提供服务，而某个服务端也可以由几台计算机通过软件进行捆绑后实现，因此，可以看出，一个普通主机装上服务端软件，即可以作为一个服务端向客户端提供服务，一台服务器的主机向另一台服务端请求服务的时候，它是作为一个客户端的身份出现的。甚至一台主机，既可以是服务端，也可以作为一个客户端。

另一个软件和硬件名称出现混淆的就是“端口”。即使是在计算机的物理硬件上，也有两个混用的“端口”，一个是计算机连接其他的外部设备的外部物理接口，一般来说统称为端口，如计算机或交换机、路由器等物理设备面板上的RJ-11端口（接电话线的Modem口）、RJ-45端口（即以太网网口）、RS232端口（早期的串行设备接口）等；另一个是专指计算机上的RS232/RS482接口，并且都使用端口（Port）作为其名称。而本处所指的端口则是逻辑上的端口，是专指通过RJ-45以太网网口连接以后，利用协议进行区分的逻辑上的一个值。由此可见，访问网络上的一个指定的服务至少需要知道IP地址和端口两个要素，即

Socket地址=（IP地址：Port端口号）

服务器要想让客户端访问自身，必须同时将二者公布于众，缺一不可；否则客户端将不知道该到哪台主机或某台主机的哪个端口上访问服务，即：

连接=(Socket地址1，Socket地址2)=（IP地址1：Port端口1，IP地址2：Port端口2）

在对端口的状态描述中，各种称呼都有，有的用“开”和“不开”，有的用“激活”和“关闭”，有的用“开”和“关”，鉴于所表示的意义完全一致，因此，后续部分中，统一称为“开”和“关”。“开”即表示有对应的服务程序通过该端口向外界提供相应服务，只要外界使用满足这一端口的协议访问该端口，就可以得到相应的服务；而“关”则表示对应的服务程序没有安装或当前没有处于运行状态，即使在客户端运行相应访问请求的程序，仍无法得到结果。比如运行浏览器IE（Internet Explorer）并在地址栏输入一个不提供WWW服务的IP地址后，IE就会得到回复：访问出错。

1.2.2 网络扫描

扫描源于物理术语，是通过对一定范围内的光或电信号进行检测处理，然后以数值或图形方式进行展示的一个操作。网络扫描也一样，是通过对一定范围内的主机的某种属性进行试探性地连接和读取操作，最终将结果展示出来的一种操作。

端口具有独占性，一旦一个服务使用了某个端口，则另外的服务不能再使用这个端口。端口的占用原则是：先申请的先使用，后申请的在申请时报错。同时，上述的这种对应关系，只是一种约定，但任何操作系统都没有强制软件遵照执行，因此在使用时，存在如下几个情况：

- 某个主机不向外界提供WWW服务，所以该主机的80端口是空闲的。但该主机上的另一个不提供WWW服务的程序使用了80端口。因此，该主机80端口是对外打开的，但不提供WWW服务。
- 某主机虽然提供WWW服务，但该主机并不想让别的人都知道该主机提供该服务，于是该主机的管理员将该主机上的WWW服务的端口由默认的80端口，改为了其他的端口8000，并将该端口告诉了他允许访问的用户。在这种情况下，需要通过其他联系方式通知所有允许访问的主机。
- 某主机对外界想同时提供基于ASP的WWW服务和基于JSP的WWW服务，二者虽然同为WWW服务，但运行机制、配置等各不相同，并且没有一个通用的软件能同时提供，且二者占用了同一个端口80，于是该主机的管理员将ASP设定为80端口，而将JSP的端口设定为8080端口，并在双方主页上互相告知对方端口的存在。

有了服务，就相当于打开了某一个或几个指定的端口，则客户端就可以通过Socket连接到服务器端的该端口，并获取服务。例如某一台计算机配置了WWW服务，该服务默认的端口是80，则在客户端上打开浏览器（如Microsoft 的Internet Explorer），输入该服务器的网址，就可以访问该服务器的WWW服务。

但在某些时候，扫描有三种应用：

1）我们不知道远端的服务器是否提供WWW服务，或者我虽然知道远端的服务器提供WWW服务，但该服务使用的不是默认的80端口，而是使用自己定义的端口，那么怎么知道对方是否提供WWW服务，并且该服务在哪个端口呢？该操作就叫WWW服务扫描。

2）我并不想知道远端的服务器是否有某一个具体的服务，我只想知道对方服务器都有哪些服务，可以通过扫描对方所有打开的端口实现，这时，该操作叫做端口扫描。

3）对于一批计算机，我想知道这些计算机是否提供某个服务，或这些计算机都打开了哪些端口，这种操作叫做批量扫描。

由此可见，所谓的网络扫描，就是一方通过某种协议，在目标主机不需知情的情况下，对其实施的一种获取想要的信息或通过读到的信息验证预想的行为过程。在此概念中有两个部分：

1）网络扫描中，目标主机可以知情，也可以不知情，因此这不是一种双方预商量的行为。

2）扫描的目的无论是读取信息，还是想验证某一个事先的预想，其目的都是作为下一步行动的参考。因此网络扫描往往不是一次单独的行动，之后通常会有下一步的操作。

1.3 网络扫描原理概述

根据扫描的概念可以发现，当一个主机向一个远端服务器的某个端口提出建立连接的请求时，如果对方有此项服务就会应答；如果对方未安装此项服务，即使向相应的端口发出请求，对方仍无应答。客户端向服务端发出请求的过程见图1.1。利用这个原理，如果对所有熟知端口或自己选定的某个范围内的熟知端口分别建立连接，并记录下远端服务器所给予的应答，通过查看记录就可以知道目标服务器上都安装了哪些服务，这个过程就叫做端口扫描，所使用的程序叫做扫描程序。通过端口扫描可以搜集到很多关于目标主机的很有参考价值的信息，例如，对方是否提供FTP服务、WWW服务或其他服务。

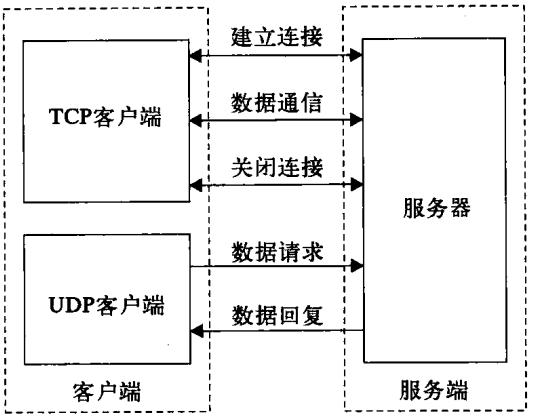


图1.1 网络TCP/UDP扫描示意图

1.4 扫描编程与客户端编程的区别

同样是网络编程，同样采用C/S（客户端/服务器）模式，在网络通信中的角色也相同，所以说网络扫描编程属于客户端编程的一种，但又不完全等同于客户端编程。为了区分，下面分别用扫描程序和客户端程序称呼二者。

与客户端程序相比，网络扫描程序具有一定的特殊性，主要体现在如下几个方面：

- 对服务端的要求不同。从实际的编程角度来看，如果是普通客户端程序的开发，客户端的开发者与服务器的开发者首先需要共同协商以决定通信时采用什么协议等细节；而扫描程序的开发者则需要通过已有的协议或猜测、试探等方式决定采用什么技术，故扫描程序对服务器端是没有要求的。同时，扫描程序的扫描结果也常具有不可预知性。
- 扫描具有全部或局部的“遍历”性，客户端具有针对性。普通客户端的开发者一旦确定了需求，剩下的就是按需求去实现各功能的细节；而扫描程序的开发者则通常通过对全部或局部进行功能遍历，以期验证自己的猜测或获得更多更详细的数据。
- 对服务器的服务支持程度不同。客户端连接服务器的目的是为了让客户端用户能远程地使用服务器所提供的各项功能，因此，通常客户端程序要支持所有的命令，有些命令哪怕只有极少数机会被用到，客户端也必须提供支持；而扫描程序则只需要支持和使用所需要的最少的几个命令。

1.5 网络扫描的目的

网络服务本身就是一种“广而告之”的通信方式，任何网络服务，如果提供者不将自身提供的服务告诉别人，别人当然也无所知晓，更谈不上使用。但实际应用中，则有各种可能情况，如网络服务的提供者只想让一部分人访问，而不是所有人都能访问；再如随着计算机

系统越来越复杂，有时运行一项功能，会默认自动地打开某个服务。

不同的扫描目的，对扫描器的要求也不一样。有些用户只需要知道某个端口是开或是关，或者只想了解某一段端口内，有哪些是开着的，从而判断对方主机的大概作用。而有些用户则不但要知道某一端口是否开，而且还要知道对方所开的这个端口是否提供了默认情况下该端口所应该提供的服务。还有些用户需要知道若干的指定端口是否同时开或同时关，因为有些服务同时占用了几个端口，如NetBIOS服务同时占用了137、138、139三个端口，如果只是其中部分端口开着，不但表明对方没有提供NetBIOS服务，还表明对方有不符合标准的程序正在使用熟知端口。

扫描的目的一般是：

- 获取某范围内的端口某未知属性的状态。这种情况下，一般是不知道对方情况，只是想通过扫描进行查找。例如，通过扫描检测某个网段内都有哪些主机是开着的。
- 获取某已知用户的特定属性的状态。这种情况下，一般是有明确的目标，有明确要做的事，下面只是查找一下某些属性。例如，通过扫描检测指定的主机中哪些端口是开的。
- 采集数据。在明确扫描目的后，主动地采集对方主机的信息，以便进行下一步的操作。例如，没有预定目的地扫描指定的主机，判断该主机都有哪些可采集的数据。
- 验证属性。在明确扫描目标，并且知道对方具有某个属性的情况下，只是通过扫描验证一下自己的想象，然后判断下一步的操作。例如通过扫描指定的服务，验证对方是否是Windows类操作系统。
- 发现漏洞。通过漏洞扫描，主动发现对方系统中存在的漏洞。如扫描对方是否具有弱密码。

扫描的用户主要为三大类：网络管理员、黑客、普通用户，他们对网络使用的角度不同，所以扫描的目的也各不相同，见表1.1。

表1.1 端口扫描用户类别和扫描目的

用户类别	扫描目的
网络管理员	通过扫描自身网络，发现漏洞，从而关掉不用的端口，安装漏洞补丁程序，判断网内主机是否按要求操作，如学校的网管判断学生是否改了IP地址
黑客	利用扫描，采集数据、发现漏洞，为进行攻击做准备
普通用户	查找网内的可用资源，判断网内主机开机状况，判断服务器是否提供某一服务

1.6 网络扫描算法

在实际扫描器编写过程中，除了有各种技术的选择之外，还需要选择合适的扫描算法。使用哪些扫描算法，也完全取决于扫描的目的，因为这些算法有些可以提高扫描效率，有些可以增加扫描准确度或扫描隐蔽性，有些甚至可能牺牲某些优点而获得所需要的特性。

1.6.1 非顺序扫描

参考已有的扫描器，会发现几乎所有的扫描器都无一例外地使用增序扫描，即对所扫描

的端口自小到大依次扫描，殊不知，这一效果可以被对方的防火墙或IDS（Intrusion Detection System，入侵检测系统）作为判断正被扫描的特征。虽然通过多线程会使这一特征发生少量的变化，但从整体效果上看，仍然显示增序现象。

改变增序特征并不难，一般有如下几种非顺序扫描算法。

1. 逆序扫描算法

顾名思义，逆序扫描就是在扫描的时候，采用从大到小的逆序扫描方式。

2. 随机重排扫描算法

随机重排扫描即重新排列要扫描端口的顺序。在新排的顺序中，为了避免漏掉或重复使用某一端口，可以采用互换位置的方式进行。这个过程可以用一个数组和随机数产生函数rand来实现：

```
#define MAXPORTCOUNT 65536
WORD *NewSort(WORD wBegin,WORD wEnd)
{
WORD buff[MAXPORTCOUNT]={0};          //建立一个数组
WORD count=wEnd-wBegin,i,wTemp,wRand;
if (count<0) return NULL;             //判断范围的合理性
srand(time(NULL));                    //随机种子
for (i=0;i<=count;i++)                //先将所有值复制到数组中
    buff[i]=wBegin+i;
    for (i=0;i<=count;i++)
    {//顺序地让每一个端口值与另一个值对换
        wRand=rand()%count;           //读取一个随机位置
        //将当前位置的数据与随机位置的数据互换
        wTemp=buff[i];
        buff[i]=buff[wRand];
        buff[wRand]=wTemp;
    }
    return (WORD *)buff;
}
```

利用这段程序可以保证打乱后的端口顺序不被遗漏，也不会重复，而新的顺序完全是随机分布的。采用这个顺序进行扫描的时候，对方防火墙在监测到某个端口连接时，无法立即对下一个要连接的端口进行预测，从而使“基于通过连续端口被连接算法进行扫描判断”的方式失效（详见12.3节“基于哨兵的端口扫描监测”）。

3. 线程前加延时扫描算法

为了提高扫描速度，很多扫描算法都采用多线程扫描。在Windows中，同级别的扫描线程通过抢占方式获得CPU的优先使用权，各个线程理论上没有先后之别，但考虑到创建时总要有一定顺序，因此即使在运行中偶尔相邻的两个线程顺序会做出调整，整体上各线程之间也有先后的顺序。

一个简单的算法就是在每一个线程中，扫描函数开始之前挂起（Halt）一个随机的时间，这样会在不影响各线程创建时间的前提下，调整各线程中扫描的顺序。具体的方法就是在线程的开始加一句：

```
Sleep(rand()%5000);//假设每个线程挂起时间是5秒以内的一个随机数
```