

西安邮电大学

毕业论文

题目： IP 网络扫描技术
的研究与实践

学院： 通信与信息工程学院(人工智能学院)

专业： 通信工程

班级： 通工 1615

学生姓名： 杨海威

学号： 03161245

导师姓名： 贺伟 职称： 研究员

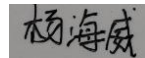
起止时间： 2020 年 3 月 5 日 至 2020 年 6 月 30 日

毕业设计（论文）承诺书

本人所提交的毕业论文《IP 网络扫描技术的研究与实践》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人深知本承诺书的法律责任，违规后果由本人承担。

签名：



日期：2020 年 3 月 5 日

西安邮电大学本科毕业设计(论文)选题审批表

申 报 人	贺伟	职 称	研究员	学 院	通信与信息工程学院 (人工智能学院)			
题目名称	IP 网络扫描技术的研究与实践							
题目来源	科研				教学		其它	√
题目类型	硬件 设计		软件 设计	√	论文		艺术 作品	
题目性质	应用研究		√		理论研究			
题目 简述	<p>(为什么申报该课题)</p> <p>网络扫描技术是 IP 网络进行网络管理和安全性能评估分析的基础技术, 本题研究当前 IP 网络中基于 ICMP/TCP/UDP 协议进行目标主机扫描的基本原理、技术和工具。要求搭建 IP 网络目标主机扫描分析的实验环境, 并基于 socket 接口或者 Wincap 开源库等常用的网络编程接口, 自主设计并实现一个综合扫描分析应用程序, 该程序能在实际网络环境下, 根据用户需求扫描指定网段主机的存在性、TCP/UDP 端口的开闭性, 并提供基本的输入输出界面, 按需进行协议、端口、IP 地址为分类的统计分析, 最后完成程序测试和毕业论文。</p>							
对学 生知 识与 能力 要求	<p>1) 有较好的 TCP/IP 网络基础知识。</p> <p>2) 熟悉 socket/Wincap 接口库以及操作系统。</p> <p>3) 有一定的 C 语言编程基础。</p> <p>4) 乐意动手、具备勇于克服困难的精神和合作精神。</p>							

<p>具体 任务 以及 预期 目标</p>	<p>(应完成的具体工作 , 预期目标和成果形式)</p> <p>1) 独立进行 IP 网络扫描技术和网络编程技术资料的搜集分析、了解本课题涉及的相关内容对社会、安全、法律等方面的影响, 制定工作计划、完成开题报告。</p> <p>2) 根据工作计划, 综合考虑时间、经济、实验室设备限制等约束条件, 提出可行的网络综合扫描实验方案。</p> <p>3) 独立完成网络扫描分析程序实验环境、及实验原型系统设计。</p> <p>4) 完成基于 Winsock/Wincap 接口之一应用系统设计实现, 能根据软件交付测试要求, 设计测试方案, 评价测试其有效性。</p> <p>5) 了解通信领域的知识产权保护及工程职业规范, 能在毕设和论文撰写过程中自觉遵守、履行责任。按照撰写规范和质量要求, 完成毕业论文撰写、验收及答辩。</p>		
<p>时间 进度</p>	<p>2020-3-16 至 2020-03-23: 研究和学习网络扫描的基础理论和网络编程接口, 完成工作计划和开题报告。</p> <p>2020-03-24 至 2020-04-19: 设计搭建可行的流量分析应用开发实验环境, 并完成实验原型需求分析。</p> <p>2020-04-20 至 2020-05-19: 选择 Socket/Wincap 接口之一, 完成编码实现实验原型。</p> <p>2020-05-20 至 2020-05-31: 测试优化实验系统, 进行几种扫描方式性能分析, 并开始论文写作。</p> <p>2020-06-01 至 2020-06-11 : 完成论文写作以及答辩。</p>		
<p>系 (教研室) 主任</p> <p>签字</p>	<p>年 月 日</p>	<p>主管院长</p> <p>签字</p>	<p>年 月 日</p>

西安邮电大学本科毕业设计（论文）开题报告

学生姓名	杨海威	学号	03161245	专业班级	通工 1615
指导教师	贺伟	题目	IP 网络扫描技术的研究与实践		

选题目的（为什么选该课题）

近年来，随着互联网的蓬勃发展，网络与人们的生活愈加紧密相关，互联网+的模式在各行各业开始得到运用，网络安全也开始被人们重视起来。IP 网络扫描是一种基于 ICMP/TCP/UDP 等协议进行目标主机扫描，以判断指定网段主机的存在性、端口的开闭性等信息的技术，通过对这些信息的统计分析可以提前暴露目标用户主机的潜在风险，为相关人员规避漏洞和管理主机提供技术支持。之所以选择这个题目，一方面是因为 IP 网络扫描具有早发现网络上潜在威胁，早处理以避免经济损失的优势；另一方面是可以通过本次毕设整合大学时期学习的通信、网络、编程等各方面的知识，为以后的学习工作打下良好的基础。

前期基础（已学课程、掌握的工具、资料积累、软硬件条件等）

- 1、熟悉计算机网络，了解常用的网络安全知识，对 TCP/IP 等相关协议有深刻的认识。
- 2、了解 linux、docker、云服务器，曾经自建网站，熟悉 C/S 等常见网站架构。
- 3、熟悉 c、java，目前正在学习 go，有网络编程相关经验，希望通过毕设提高自己的 go 工程能力。
- 4、已查阅相关完档对整体设计有一定思路，收集整理了相关资料。

【参考文献】

- [1] 李瑞民. 网络扫描技术揭秘：原理、实践与扫描器的实现[M]. 北京：电子工业出版社，2012
- [2] 凯文 R. 福尔（Kevin R. Fall）. TCP/IP 详解卷 1：协议原书第 2 版[M]. 北京：机械工业出版社，2016
- [3] 任波. 计算机网络安全与漏洞扫描技术的应用研究[J]. 信息与电脑（理论版），2019，24
- [4] 陈家东. 网络安全扫描系统实现技术研究[D]. 武汉：华中科技大学，2007

要研究和解决的问题（做什么）

- 1、提前查阅收集网络安全及 TCP/IP 资料，学习 go 网络编程。了解网络扫描实现的原理及模拟环境的搭建方案，制定工作计划。
- 2、根据工作计划，综合考虑时间、技术等各方面因素，提出可行的实验方案。
- 3、独立完成局域网模拟环境的搭建，完成 go 扫描器的编程，进行测试。
- 4、根据要求设计相关方案测试扫描器的功能，提升其性能和可靠性，完成实验并整理实验数据。
- 5、按照撰写规范和质量要求，完成毕业论文撰写、验收及答辩。

工作思路和方案（怎么做）

工作思路：

首先查阅网络安全和 TCP/IP 相关知识，了解 IP 网络扫描器的基本原理。目前打算采用 docker 模拟局域网环境，阿里云服务器模拟域名扫描，然后是本机扫描。编程采用 go，通过 go 的 http 库实现相关功能，完成一个 go 二进制文件，可以在 mac、linux、windows 环境运行。由于 go 没有官方的 GUI 库用于桌面图形展示，个人考虑使用 web 页面或者 fyne 展示。最后完成试验，记录实验数据，开始毕业论文的编写。

工作方案：

2019-3-4 至 2019-03-19： 研究和学习网络安全、TCP/IP 知识，完成工作计划和开题报告。

2019-03-20 至 2019-04-19： 完成 docker 局域网、阿里云域名的环境搭建，查看 fyne 的官方文档并了解基本用法，使用 go 完成 IP 网络扫描器。

2019-04-20 至 2019-05-19： 进行联调，测试代码是否有 bug，提高其性能及可靠性，记录实验数据

2019-05-20 至 2019-05-31： 根据实验数据和相关资料编写论文。

2019-06-01 至 2019-06-11： 完成论文写作以及答辩。

指导教师意见

签字

年 月 日

西安邮电大学毕业设计 (论文)成绩评定表

学生姓名	杨海威	性别	男	学号	03161245	专业 班级	通工 1615
课题名称	IP 网络扫描技术的研究与实践						
指导教师 意见	(从开题论证、论文内容、撰写规范性、学习态度、创新等方面进行考核)						
评阅 (验收) 意见	(从选题、开题论证、论文内容、撰写规范性、创新和预期成果等方面进行考核)						
答辩 小组 意见	(从准备、陈述、回答、仪表等方面进行考核)						
评分比例	指导教师评分 20 (%) 评阅 (验收) 评分 40 (%) 答辩小组评分 40 (%)						
学生总评 成绩	百分制成绩				等级制成绩		
答辩委 员会意 见	毕业论文(设计)最终成绩(等级) : 学院答辩委员会主任(签字、学院盖章) : 年 月 日						

目 录

第一章 概述.....	1
1.1 课题研究背景	1
1.2 系统研发主要内容、意义及目标.....	1
1.3 系统开发语言环境简介.....	1
第二章 相关技术介绍.....	3
2.1 网络扫描技术	3
2.2 TCP/IP 协议简介	4
2.3 go 协程简介	6
第三章 系统需求分析.....	7
3.1 系统可行性分析	8
3.2 系统业务需求分析	8
3.3 系统环境需求分析	8
第四章 系统设计及实现	10
4.1 系统总体流程	10
4.2 系统各功能模块	11
第五章 测试与结果分析	20
5.1 测试实例.....	20
5.2 测试结果与分析	22
结论	23
致谢	24
参考文献.....	25

摘 要

21 世纪随着互联网技术的蓬勃发展,网络与人们的联系越来越紧密,互联网+的模式在各行各业开始得到运用,与此同时网络安全问题也开始被人们重视起来。IP 网络扫描是一种基于 ICMP/TCP/UDP 等协议进行目标主机扫描,以判断指定网段主机的存在性、端口的开闭性等信息的技术,通过对这些信息的统计分析可以提前暴露目标用户主机的潜在风险,为相关人员规避漏洞和管理主机提供技术支持。

本文分析了 IP 网络扫描技术的原理与常见的实现方式,采用较为新颖的 goLang 作为编程语言,完成了一个高性能的网络扫描器,能够对指定域名、网段、端口进行扫描,并将扫描结果保存到本地以供管理员使用。该扫描器得益于 Go 语言的 goroutine,即协程特性在性能上有较大提升,能够支持 Mac、Win 和 Linux 多个平台,有较大的实际使用价值。

关键词: IP 网络扫描; 网络安全; 规避漏洞; go 语言; 协程技术; 跨平台

ABSTRACT

In recent years, with the vigorous development of Internet technology, the network has become more closely related to people's lives. The Internet + model has been used in all walks of life, and network security has begun to be valued by people. IP network scanning is a technology that scans target hosts based on ICMP / TCP / UDP and other protocols to determine the existence of specified network segment hosts and the opening and closing of ports. Through statistical analysis of these information, targets can be exposed in advance. The potential risks of the user host provide technical support for related personnel to avoid vulnerabilities and manage the host.

This article analyzes the principles and common implementation methods of IP network scanning technology, uses a relatively new goLang as a programming language, completes a high-performance network scanner, can scan the specified domain name, network segment, port, and scan the results. Save to local for administrator use. The scanner benefits from the goroutine of the go language, that is, the coroutine feature has a big improvement in performance, can support multiple platforms of Mac, Win and Linux, and has a large practical use value.

Key words: IP Network Scanning; Network Security; Avoid Vulnerabilities; Go Language; Coroutine Technology; Cross-platform

引 言

随着当今互联网技术的发展，人们生活从每一个方面都融入了互联网的脉络中，Internet 成为社会经济发展最重要的推手。当今世界在网络的连接下已变成一个地球村，各行各业通过网络更加高效的完成资金与技术的流通，产生更大的经济效益。从来没有过这样一个事物能影响人们生活的方方面面，各种各样的信息都在网络上流通，网络安全在今天决定着一个企业能否成功，此时保证网络信息安全就是重中之重。

IP 网络扫描技术是一种基于 TCP/IP 等协议对目标主机进行扫描，从而发现目标主机是否有安全漏洞的一种技术。通过判断指定网段主机的存在性、端口的开闭性对其安全性作出评估，对这些信息的统计分析可以提前暴露目标用户主机的潜在风险，为相关人员规避漏洞和管理主机提供技术支持，保障企业网络不受攻击正常运行，创造更大的经济效益。

因为网络安全在当前社会经济活动有着巨大的价值，只有深入研究网络可能产生的安全隐患，分析其产生的原因，探究原理总结出特点，对此进行有效的保护，才能降低系统受到攻击造成信息泄漏的概率以减少损失。由于这个原因，一款高性能，跨平台的 IP 网络端口扫描器有着很大的应用价值，本文就详细介绍了该扫描器的设计原理及具体实现，为企业网络安全保驾护航。

第一章 概述

1.1 课题研究背景

21 世纪互联网技术蓬勃发展，在各行各业与人们的生活交织在一起，各种互联网+模式层出不穷。信息通过网络高效的收集流通，成为社会经济生产最重要的元素，对于个人、企业和国家来说，信息安全是决定事业成败的关键。

近年来，各种网络安全问题频发，小到个人隐私泄漏，大到某些国家监控全球，人们对网络安全愈发重视。如何防范网络攻击、杜绝信息被窃取成为人们研究的热门话题。

为了避免信息泄漏造成的损失，早发现早解决是最好的办法。所以我分析了 IP 网络扫描技术的原理与常见的实现方式，采用较为新颖的 GoLang 作为编程语言，完成了一个高性能的网络扫描器，能够对指定域名、网段、端口进行扫描，并将扫描结果保存到本地以供管理员使用，提前发现漏洞，提前解决，避免可能产生的经济损失。该扫描器利用 Go 协程特性，在性能上有明显的优势，可以跨平台在 Mac、Linux、Win 等多种环境运行，在实际生产环境中较大的应用价值。

1.2 系统研发主要内容、意义及目标

由课题研究背景可以看出，IP 网络扫描技术在网络安全这一方面有着重大意义。因此我们需要深入认识理解扫描技术的原理及特点，分析其优缺点，选用恰当的扫描技术尽可能提高扫描的速度和可靠性，及时发现系统的安全漏洞，快速处理加以修复，避免可能存在的安全风险，保障网络安全。

本系统旨在完成一个高性能、跨平台的 IP 网络扫描器，能够快速准确的发现指定域名、网段、端口的暴露情况，及时反映给管理员，方便管理员瞬速定位问题，及时处理，减少安全风险。主要实现以下功能：

- 1、扫描特定域名判断其是否存在
- 2、扫描特定网段范围主机的工作状态
- 3、识别目标主机的端口状态
- 4、保存风险信息，分析目标主机弱点
- 5、生成扫描结果日志以供分析

1.3 系统开发语言环境简介

本系统在 Mac 平台开发，采用谷歌开发的 Golang 作为开发语言，JetBrains 开发的 GoLand 作为编辑器，实现了一款高性能、跨平台的 IP 网络扫描器，有效减少了网络安全风险。

Go 语言是 Google 开发的一种静态强类型、编译型、并发型，并具有垃圾回收功能的编程语言，它用批判吸收的眼光，融合 C 语言、Java 等众家之长，将简

洁、高效演绎得淋漓尽致，它具有以下几个优点：

- 1、自动垃圾回收降低了开发难度
- 2、更丰富的内置类型
- 3、支持函数多返回值
- 4、简洁的类型和“非侵入式”接口
- 5、并发编程更轻盈更安全

编辑器采用 GoLand，它是由 JetBrains 公司推出的 Go 语言集成开发环境，非常适合 Go 语言开发者。它能够实时检测代码错误并给出修改意见，自动格式化和导包，智能代码提示，快速创建方法和查看 godoc，极大提高开发者的使用效率，是 Go 语言开发的一款利器。通过 GoLand 可以创建简洁、可靠、强壮的代码，方便整个开发过程。

第二章 相关技术介绍

2.1 网络扫描技术

2.1.1 网络扫描技术的原理

网络扫描技术核心思想是利用目标主机对探测数据包的响应情况来判断是否存在安全风险。当扫描器发送特定的探测数据包时，此时目标主机会返回一系列响应包，据此我们可以得知目标主机是否存在，它的相关端口和对应进程服务是打开还是关闭，通过分析这些信息就会发现潜在的漏洞，针对性的进行处理避免网络安全问题的发生。

2.1.2 常见扫描技术

1、ICMP 协议（主机扫描）

网际控制报文协议（Internet Control Message Protocol），属于 IP 协议，在 IP 协议栈中必须实现。主机扫描主要是确定目标主机是否存在，特点是不保证可靠性，不用来反映 ICMP 报文的传输情况。

2、TCP 协议（端口扫描）

TCP（Transmission Control Protocol）属于传输层，是一种面向连接、可靠交付、点对点的报文段。当发送信息时首先建立连接，然后开始传送数据，传送完毕后需要关闭连接，一般被称为三次握手，四次挥手，利用标志位完成端口扫描的功能。

当我们通过主机扫描确定了目标主机存在可以访问后，就可以使用端口扫描技术，利用 TCP 协议确定目标主机的开放端口，获取目标主机的服务信息，分析漏洞情况。

3、开放扫描技术：主要是 TCP Connect 扫描

开放扫描技术是利用通过 socket 编程实现，利用 connect()函数与目标主机建立连接，完成一次 TCP 协议的三次握手。调用该函数时，如果目标主机的端口打开状态，则 socket()函数可以成功返回，否则这个端口不可用，即没有提供服务。这种扫描方式的优势是不需要任何用户权限，扫描可靠性高，结果稳定。但由于会在目标主机留下记录，服务器会存在大量的错误连接日志，极易被发现，不够隐蔽，容易被定位和屏蔽。

4、半开放扫描技术

TCP SYN 扫描，这种扫描方式直接发送 SYN 包给对应的目标主机，如果目标主机端口没有开启，就会响应 RST 包；如果目标主机端口处于打开状态，就会响应 ACK 包和 SYN 包，再向目标主机发送一个停止建立连接的 RST 包标志。这种扫描方式没有建立 TCP 整个连接，一般被称为半连接扫描。

TCP 间接扫描，这种扫描会构造一个虚假 IP 代替自己的真实 IP，用这种方式欺骗目标主机。为了得到真实的扫描结果，防止被本机发送的响应信息影响，要

持续的监控目标主机的活动，观察对应的 IP 序列号如何增长，发现其规律来判断端口的开闭。

5、隐蔽扫描

隐蔽扫描主要有 TCP FIN 扫描、TCP Xmas 扫描、TCP Null 扫描、TCP ftp proxy 扫描、分段扫描方式，一般综合使用多种扫描方式，在类 unix 平台更多使用间接扫描 + (TCP FIN 或 TCP Xmas 或 TCP Null) + 分段扫描的方式，这种扫描方式能够穿越防火墙而不被发现。

2.2 TCP/IP 协议简介

TCP/IP (Transmission Control Protocol/Internet Protocol, 传输控制协议/网际协议) 是计算机网络中最重要的协议簇，在网络传输中十分重要，它主要包括 TCP、UDP、IP 等。

TCP 即传输控制协议，是一种可靠交付、面向连接、基于字节流的网络协议，而 UDP 是一种无连接、尽最大努力交付的全报文，它们的特点区别如下

TCP	UDP
20字节	8字节
报文段	全报文
面向连接	无连接
可靠交付	尽最大努力交付
点对点	多对多 (1)
面向字节流	没有拥塞控制

图 2.1 TCP 与 UDP 区别

IP 协议属于网络层，主要功能是寻址和路由、传递服务、数据包的分片和重组。其数据包结构如下：



图 2.2 IP 数据包结构

TCP 协议建立连接和终止连接的过程是三次握手和四次挥手，在这个过程中完成信息的传输，建立连接过程如图 2.3。

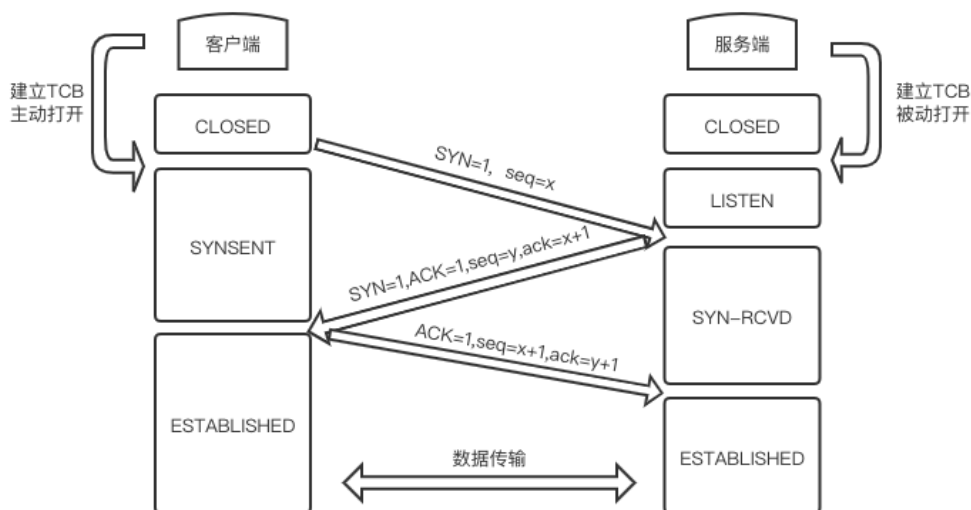


图 2.3 建立连接：三次握手

建立 TCP 连接时，客户端和服务端总共完成 3 次请求：

- 1、客户端发送初始序号 x 和 $\text{syn}=1$ 请求标志。
- 2、服务器发送请求标志 syn ，发送确认标志 ACK ，发送自己的序号 $\text{seq}=y$ ，发送客户端的确认序号 $\text{ack}=x+1$ 。
- 3、客户端发送 ACK 确认号，发送自己的序号 $\text{seq}=x+1$ ，发送对方的确认号 $\text{ack}=y+1$ 。

关闭连接过程如图 2.4。

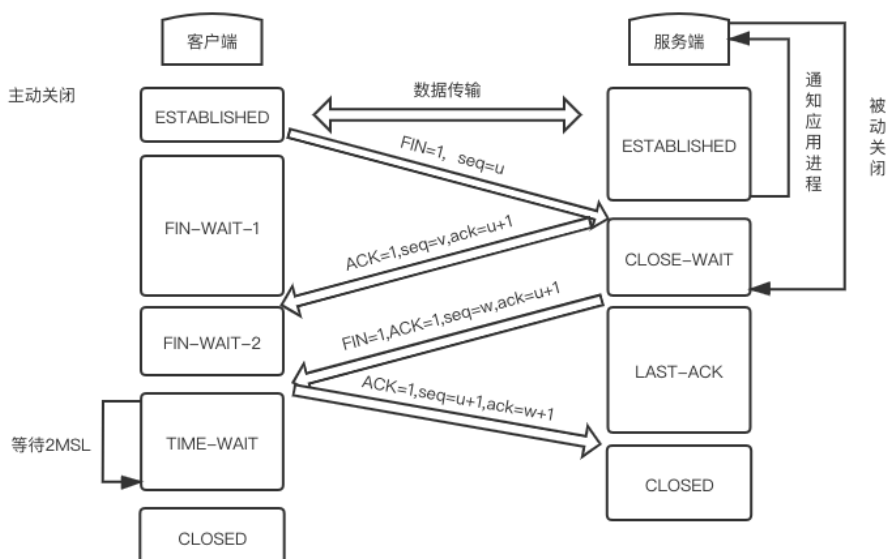


图 2.4 关闭连接：四次挥手

关闭连接共分以下 4 步：

- 1、客户端请求断开 `FIN,seq=u`
- 2、服务器确认客户端的断开请求 `ACK,ack=u+1,seq=v`
- 3、服务器请求断开 `FIN,seq=w,ACK,ack=u+1`
- 4、客户端确认服务器的断开 `ACK,ack=w+1,seq=u+1`

2.3 go 协程简介

Go 语言最大的特色就是从语言层面支持并发（Goroutine），每一个 go 程序至少开辟一个协程，goroutine 是 Go 语言最基本的执行单元。我们知道一个进程由多个线程执行，而一个线程又可以运行多个协程。在一个线程内的协程睡眠时它的运行权会直接交给其它协程，不会一直占用线程的资源，提高 CPU 的使用效率。

协程有三个状态，分别是休眠态、就绪态和运行态。在一个线程中，存在多个协程，但同时只有一个协程处于运行态，当该协程不需要使用时，其它处于就绪态的协程就会转变为运行态。休眠态的协程会在一些条件发生时，如 IO 操作的完成、睡眠时间的结束等才会激活，具备运行能力。

Go 语言的协程实现了一种非抢占式的调度，在程序运行时，调度器会采用 work-stealing 算法。这种算法当一个线程上不存在协程或者都处于休眠状态时，就会寻找其它线程，在等待队列上获取一些协程使用，是每个线程都有工作，使系统最大效率运转，提高 CPU 的使用效率。

第三章 系统需求分析

3.1 系统可行性分析

可行性分析是一个项目设立之初最重要的事情，只有经过可行性分析才能决定是否开启本项目。在开始一个项目时，我们需要客观评价其花费的时间、人力、物力和最终效果等，必须确保我们的投入在指定时间会有所回报，达到预期效果。可行性的分析是为了确保我们在较为合理的投入会在规定时间内能够获得预期结果，以此判断这个项目是否值得研发。

3.1.1 经济可行性

如今网络发展十分迅速，各大高校企业都有自己的服务器或使用云服务器，在内网部署自己的项目，也都有相应的运维工作人员维护，所以一款高性能的 IP 网络扫描器有着巨大的运用价值，可以代替人工提高效率，节省资源，快速准确发现潜在的风险使系统更加安全。

3.1.2 技术可行性

本人熟悉 C、JAVA、Go 等编程语言，对网络知识十分了解，有完成网络扫描器的技术储备，能够独立完成开发整个项目。

编程语言采用 Go，尽管可借鉴的一些实现方式基本都是采用 C 或者 JAVA 完成，但由于我最近在学习 Go 语言，想尝试使用 Go 完成整个系统。Go 语言出生名门，具有高性能、编译快、标准库丰富、并发编程容易等优势，可以打包成 2 进制文件跨平台执行，很适合网络扫描器的技术要求。

开发环境是 GoLand，它是 JetBrains 新开发的 Go 语言编辑器，帮助 Go 开发着更加方便、高效、舒适的编写程序。它支持各种插件，可以完成其它工具的扩展，能够一键编程打包发布。它十分智能，能够自动识别错误，提示相关信息和代码完成，支持格式化、模块化、自动导航，方便大型项目的编写和后期的完善，是 Go 语言开发人员最好的助手。

综上所述，系统在软件、硬件、技术能力都已达到要求，具备完成所需的全部要素，在技术上是可行的。

3.1.3 操作可行性

本系统在 Linux、Mac、Win 上都能运行，不必担心不同公司所在环境。才用命令行的方式符合运维人员的操作习惯，指令简单，逻辑清晰，运行十分快速方便。

3.2 系统业务需求分析

通过对选题进行充分调研，对企业所面临的问题有了更充分的认知，本系统

主要是扫描局域网环境内一段网段范围内目标主机来判断端口开闭的情况。

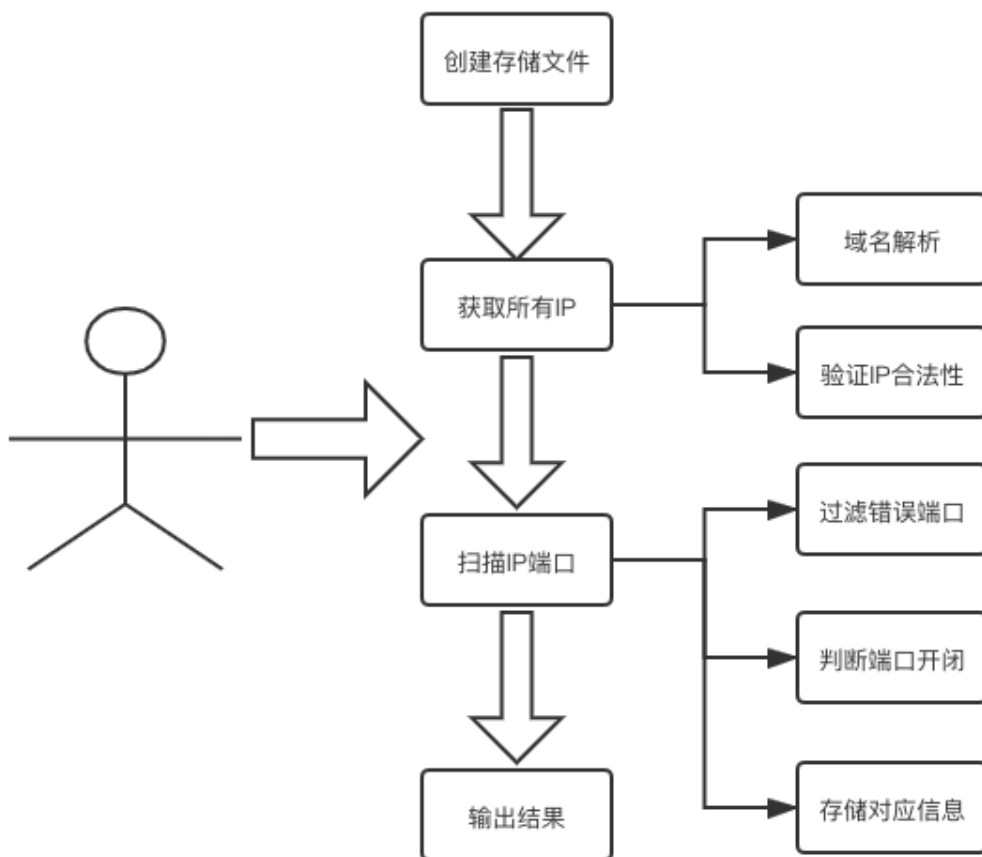


图 3.1 系统核心功能

以上就是系统的核心功能，主要是存储信息到日志中，获取所有 IP，扫描 IP 对应的端口，显示结果。在设计系统时需要满足以下四条原则：

准确性：扫描结果要十分准确。

实用性：使用方便、操作简单。

高效性：性能优良，扫描结果快速。

扩展性：模块功能划分清晰，便于后续维护扩展。

3.3 系统环境需求分析

3.3.1 运行环境

实例是 ecs.c6.large2 核 4GB，计算型 c6 I/O 优化实例，操作系统 Linux、Mac 或者 Windows 64 位，系统盘是高效云盘 40GB，固定带宽 1Mbps。已装上 MySQL、Redis 等相关软件占用端口，快照组配置完毕。

3.3.2 开发环境

电脑配置：MacBook Pro (15-inch, 2019) i9 DDR4 16GB 内存 512GB 硬盘

操作系统：MacOS 10.14.6

编程语言：Go 1.13

编辑环境：GoLand 2019.3

第四章 系统设计及实现

4.1 系统总体流程

系统总共分为四大模块，每个模块实现对应功能，主要分为信息交互模块、获取 IP 模块、端口扫描模块、信息存储模块，基本结构如图 4.1 所示：

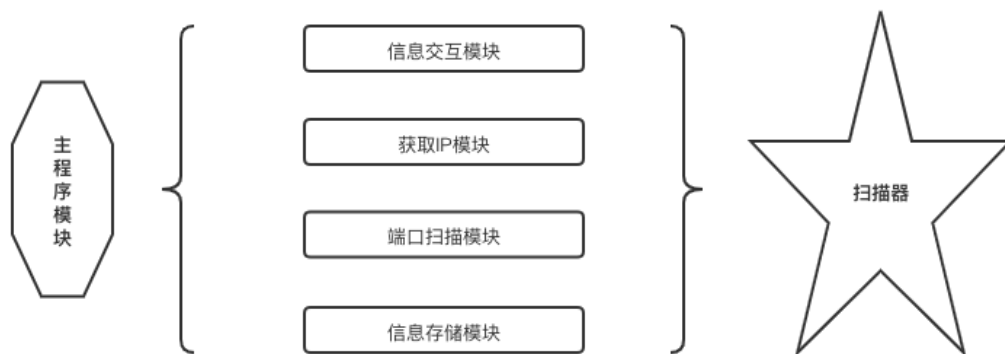


图 4.1 程序模块图

主模块完成整个程序的调度，每个模块实现自己的功能，是程序低耦合高扩展，主要包括以下功能：

- 1、信息交互模块：完成人机交互，获取网段、端口和设置等信息，并在程序完成后展示效果。
- 2、获取 IP 模块：获取所有 IP，完成域名解析、IP 合法验证等功能。
- 3、端口扫描模块：判断端口是否合法和开闭，过滤重复端口。
- 4、信息存储模块：完成 IP 及端口相关信息的存储，便于管理员查询使用规避风险。

系统总体流程主要分为获取输入信息、获取所有 IP、判断目标主机端口开闭、存储 IP 端口信息和显示结果，具体流程为：

- 1、提示帮助信息：包括 IP 地址、端口号范围、日志地址、超时时长、协程数等信息，通过提示信息是管理员更加方便快捷进行操作，如果不输入，则启用默认值。
- 2、创建目录和日志：根据输入参数在对应地址创建日志，以供扫描结果的存储，方便展示。
- 3、初始化：初始化结构体和相应参数。
- 4、获取所有 IP：第一步判断 IP 参数是否是域名，如果是则将域名解析成对应 IP 地址存储；第二步根据输入的网段获取全部 IP 存放在 slice 中；第三步验证 IP 是否合法，过滤掉不合适的 IP。

- 5、扫描 IP 端口：第一步是过滤错误的不在范围内的端口；第二步是判断该端口的开闭，即对应服务是否开启；第三步是将获取的 IP 和端口开闭信息存储在日志中以供查询。
- 6、显示结果：控制台显示扫描器扫描结果以供管理员使用。

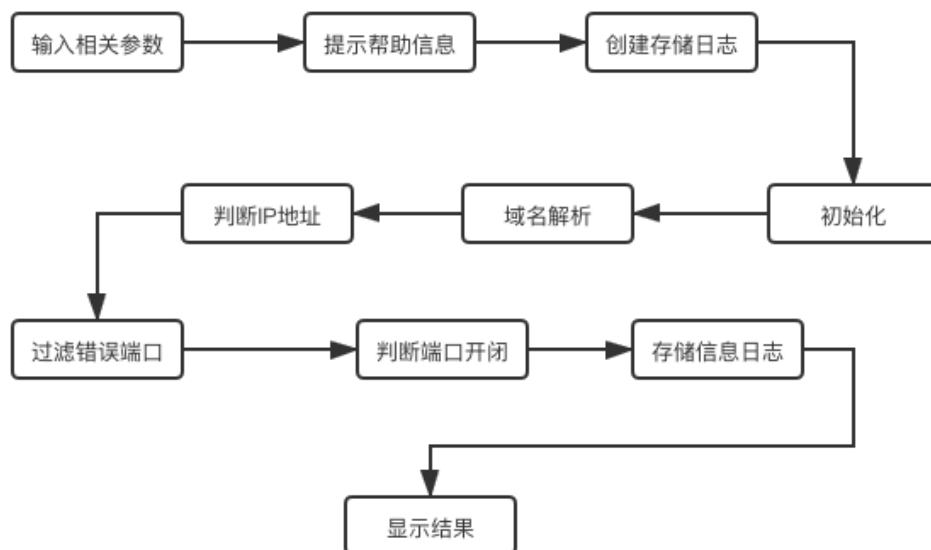


图 4.2 总体流程图

4.2 系统各功能模块

上述简要介绍了 IP 网络扫描器的基本模块和整体流程, 在这里我将详细介绍整个系统的设计思路, 各模块实现的细节及原理, 和采用的数据结构以及 Go 协程的使用, 使大家对本系统有更清晰的认识。受限于篇幅, 部分细节不会讲解, 我只会对每一部分重点内容进行讲解。

4.2.1 提示信息

提示信息是操作人员初次接触该程序时使用, 主要帮助管理员输入正确参数使系统正常工作。如果系统没有设置, 则会采用默认参数, 也能正常运行。它会在程序一开始加载就执行, 有 ip, port, path, timeout, process, h 这 6 个参数。

```

var (
    startTime = time.Now()
    ip        = flag.String(name: "ip", value: "127.0.0.1", usage: "ip地址 例如:-ip=192.168.0.1-255")
    port      = flag.String(name: "p", value: "80-1000", usage: "端口号范围 例如:-p=80,81,88-1000")
    path      = flag.String(name: "path", value: "log", usage: "日志地址 例如:-path=log")
    timeout   = flag.Int(name: "t", value: 200, usage: "超时时长(毫秒) 例如:-t=200")
    process   = flag.Int(name: "n", value: 100, usage: "协程数 例如:-n=10")
    h         = flag.Bool(name: "h", value: false, usage: "帮助信息")
)
  
```

图 4.3 提示信息

ip: 输入域名或者 IP 网段 c, 在获取所有 IP 模块会得到一个切片存储所有 IP

port: 输入特定端口或者端口范围

path: 日志存储的范围

timeout: 程序出现 bug 时的超时时长

process: 主机开辟的协程数

h: 提示信息

4.2.2 数据结构

该系统数据结构较为简单, 在程序一开始运行时 flag.Parse()函数会加载提示信息中对应的参数直接传入程序, 不需要太过复杂的程序。

//ip 扫描

```
type ScanIp struct {
```

```
    debug    bool
```

```
    timeout int
```

```
    process int
```

```
}
```

整个程序的核心功能有以下几个函数:

//获取 scanIP

```
func NewScanIp(timeout int,process int,debug bool) *ScanIp
```

//创建日志目录

```
func Mkdir(path string)
```

//记录日志

```
func (s *ScanIp) sendLog(str string)
```

//获取所有 ip

```
func (s *ScanIp) GetAllIp(ip string) ([]string, error)
```

//获取所有端口

```
func (s *ScanIp) getAllPort(port string) ([]int, error)
```

//获取开放端口号

```
func (s *ScanIp) GetIpOpenPort(ip string, port string) []int
```

//端口合法性过滤

```
func (s *ScanIp) filterPort(str string) (int, error)
```

//查看端口号是否打开

```
func (s *ScanIp) isOpen(ip string, port int) bool
```

//数组去重

```
func (s *ScanIp) arrayUnique(arr []int) []int
```

4.2.3 主程序模块

该模块是程序的调度中心，主要负责初始化和各种配置、扫描 ip 存入日志、处理异常、调度其它模块。

- 1、调用 `flag.Parse()` 函数，加载管理员输入的配置参数并判断是否正确。
- 2、创建日志存储目录并初始化
- 3、扫描所有 IP 并将获取的信息存储到日志，处理 panic
- 4、展示信息

主模块逻辑简单清晰，运行逻辑如图 4.4 所示：

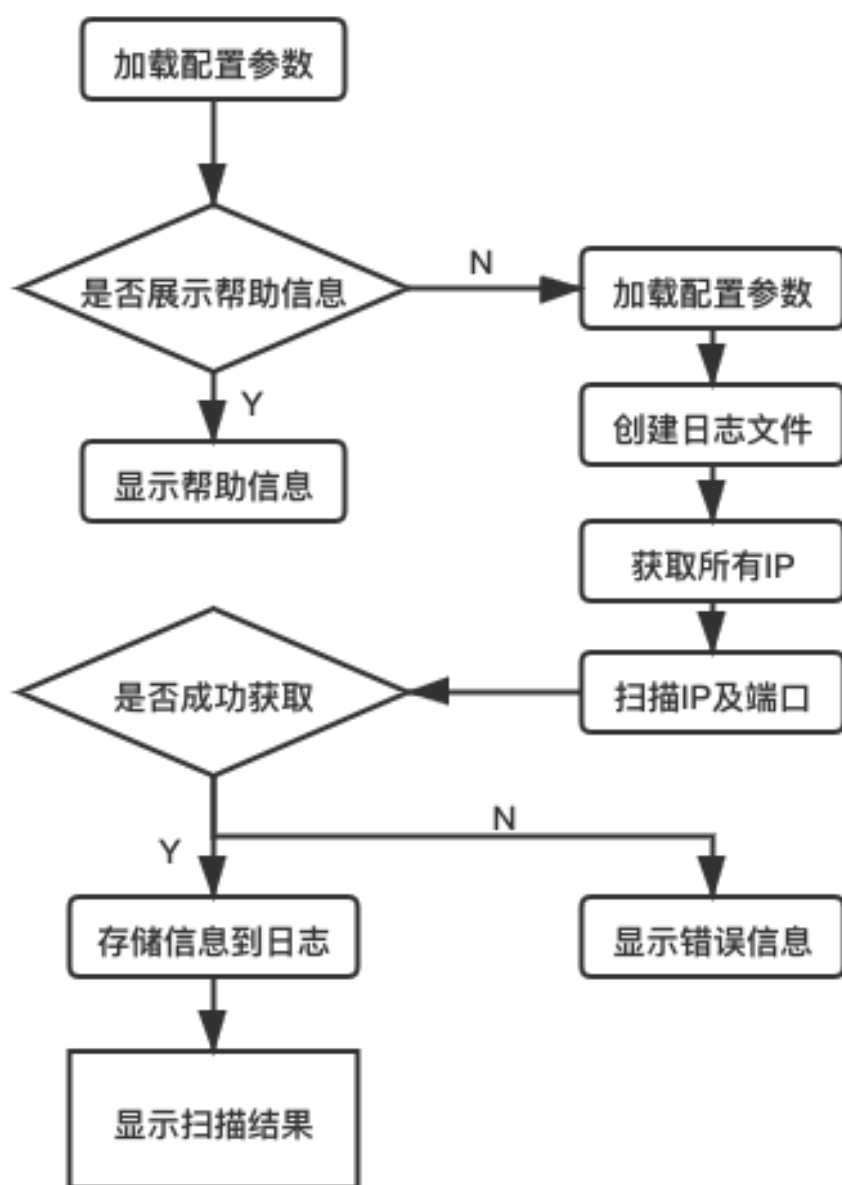


图 4.4 主模块逻辑图

4.2.4 获取 IP 模块

本模块主要是获取所有域名（DNS）解析出来的 IP 地址或者 IP 网段范围内的所有 IP，以供下一阶段端口扫描使用。

DNS 实际上是一个分布式数据库，在这个数据库中保存 IP 和主机名之间的对应关系，它有多个数据库，每个数据库只保存自己对应的数据以供查询使用。域名具有层次结构，从上到下依次为：根域名、顶级域名、二级域名。DNS 大多数情况下使用 UDP 进行传输，默认端口号为 53。当返回字节大于 512 或区域传送时使用 TCP 传输，端口号也为 53。由于一般使用 UDP 传输，域名解析器和服务器都必须自己采取措施处理超时和重传情况，从而保证系统的可靠性。整个解析过程如下：

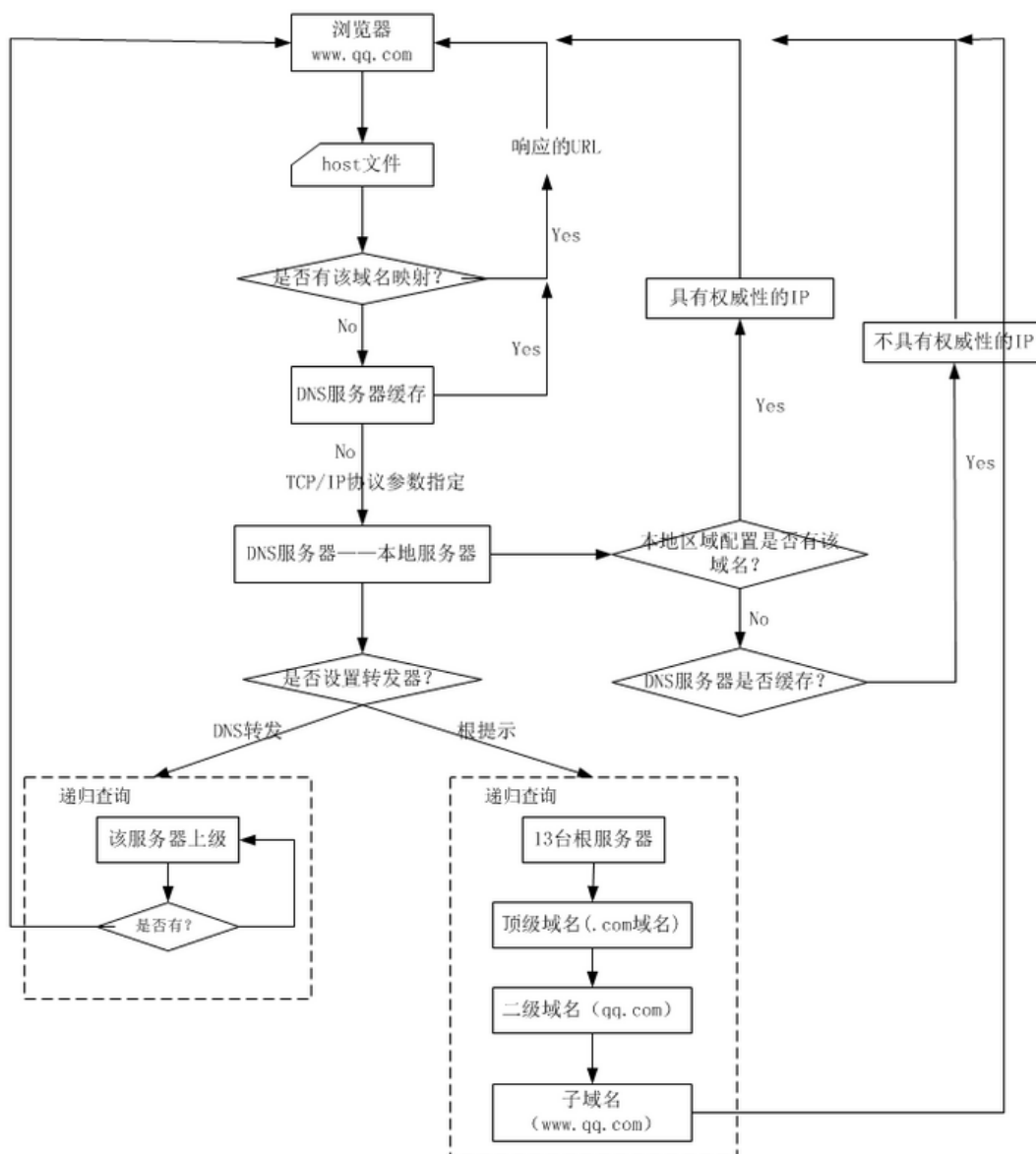


图 4.5 DNS 解析过程

通过 DNS 解析可以将域名解析为对应的 IP 地址，IP 地址由网络号和主机号两部分组成，各自都有固定的长度。

IP 地址 ::= { < 网络号 >, < 主机号 > }

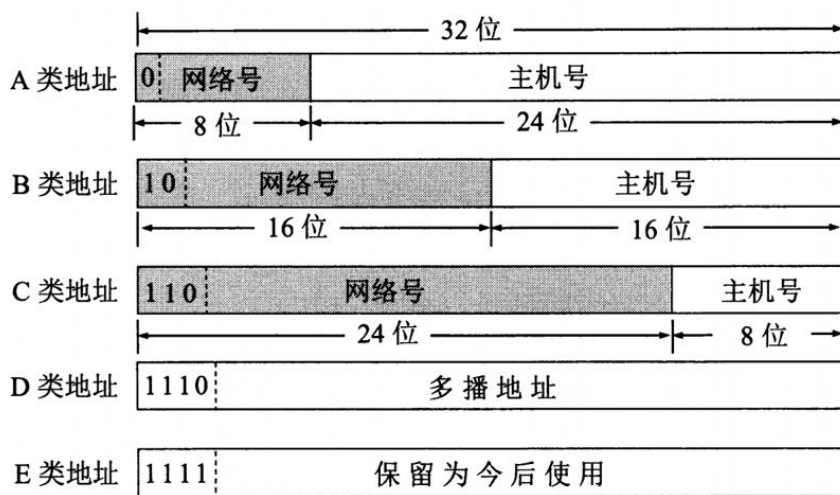


图 4.6 IP 地址中的网络号字段和主机号字段

获取 IP 地址后要把拿到的 IP 进行验证，在这里我们首先要明确子网划分的概念，我们需要把主机号的一部分字段拿出，用作子网号，然后可以把这两级 IP 地址划分成三级 IP 地址。

IP 地址 ::= { < 网络号 >, < 子网号 >, < 主机号 > }

首先我们要配置子网掩码才能使用子网。假设有一个 B 类地址，它的子网是两个比特，255.255.5.5 是默认子网掩码，经过计算它的子网掩码就是为 11111111 11111111 11000000 00000000，换算后就是 255.255.192.0。

验证完 IP 后再通过地址解析协议 ARP 获取对应的 mac 地址，即目标主机。方便下一阶段扫描目标主机所有端口。

主要实现代码如下：

```
//分割 IP 网段
ipTmp := strings.Split(ip, "-")
//域名解析
firstIp, err := net.ResolveIPAddr("ip", ipTmp[0])
net.ParseIP(firstIp.String())
//获取所有 IP
for i := 1; i < totalIp; i++ {
    ips = append(ips, fmt.Sprintf("%s.%s.%s.%d", ipTmp[0], ipTmp[1],
    ipTmp[2], startIp+i))
}
```

在该模块主要完成 DNS 解析、网段划分, IP 验证, 获取所有 IP 这 4 个功能, 通过本模块即可收集到所有目标主机, 方便扫描器扫描端口。

4.2.5 端口扫描模块

端口扫描模块是本系统最重要的模块, 在这一模块中完成了对目标主机端口的扫描, 通过判断端口的开闭来确定对应服务是否运行, 从而确定主机是否存在网络安全问题, 核心流程如图 4.7:

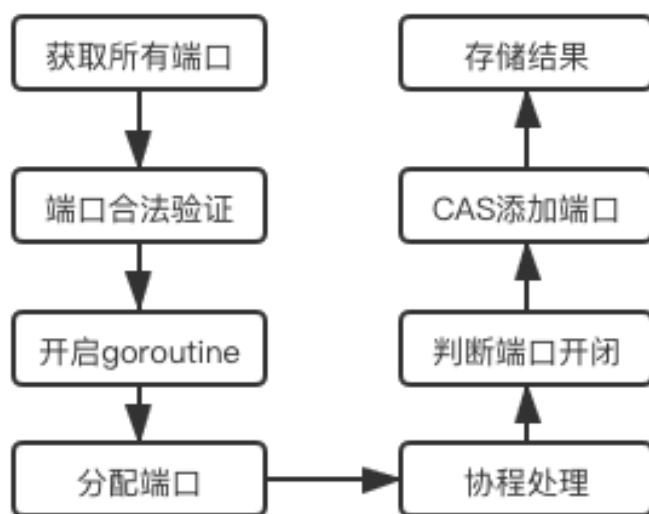


图 4.7 端口扫描流程

在整个过程中最重要的是 `net.DialTimeout` 和 `goroutine` 的使用, `net.DialTimeout` 函数是利用 ICMP 协议与 TCP 协议实现。

ICMP 协议是一种封装在数据报中的低层协议, 它可以更加高效的转发网络中的 IP 数据包, 由差错报告报文和询问报文两部分组成, 能够有效提高交付成功的几率, 报文类型如图 4.8:

ICMP 报文种类	类型的值	ICMP 报文的类型
差错报告报文	3	终点不可达
	11	时间超过
	12	参数问题
	5	改变路由(Redirect)
询问报文	8 或 0	回送(Echo)请求或回答
	13 或 14	时间戳(Timestamp)请求或回答

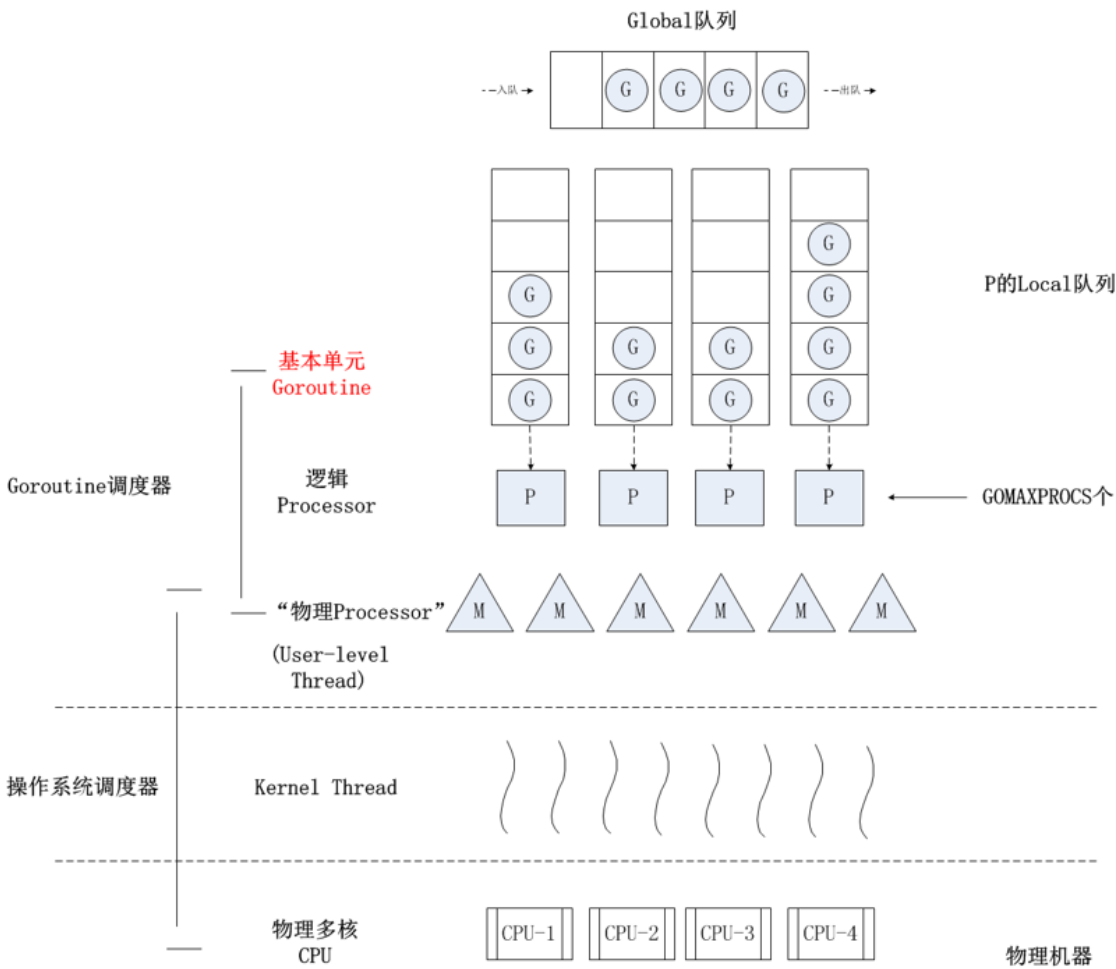
图 4.8 常见 ICMP 报文类型

例如平时经常使用的 ping 命令，就是向目标主机发送 ICMP 数据包，等待目标主机的响应。如果能收到恢复，则表明目标主机可以访问，否则表明目标主机不可达或者已经被对方系统过滤掉。

TCP 协议是面向连接的，提供可靠交付，有流量控制，拥塞控制，提供全双工通信，面向字节流的一种协议。在前面已经介绍了 TCP 协议三次握手、四次挥手的具体步骤，而端口开闭就是利用 TCP 的三次握手应答机制来判断对应服务是否运行的。

由于可能扫描的端口数量过多，再加上有很多目标主机，会存在性能瓶颈。在实现过程中，采用 Go 特有的协程来实现，高效的利用了系统资源，有效提高了程序性能。

goroutine 能拥有强大的并发实现是通过 GPM 调度模型实现，下面就来解释下 goroutine 的调度模型，如图 4.9 所示。



Goroutine调度原理图

图 4.9 Goroutine 原理图

Go 的调度器内部有四个重要的结构：M，P，S，Sched。

M: M 代表内核级线程，一个 M 就是一个线程，goroutine 就是跑在 M 之上的；M 是一个很大的结构，里面维护小对象内存 cache (mcache)、当前执行的

goroutine、随机数发生器等非常多的信息

G: 代表一个 goroutine，它有自己的栈，instruction pointer 和其他信息（正在等待的 channel 等等），用于调度。

P: P 全称是 Processor，处理器，它的主要用途就是用来执行 goroutine 的，所以它也维护了一个 goroutine 队列，里面存储了所有需要它来执行的 goroutine。

Sched: 代表调度器，它维护有存储 M 和 G 的队列以及调度器的一些状态信息等。

G 要想到 M 上执行，必须先绑定一个 P，然后 P 在 M 上执行，所以我说 P 是 G 和 M 的中间层，P 的数量决定了，同时最多有几个 G 在执行，P 数数量小于等于 CPU 的核数。P 可以控制整个程序的并发程度。

由 P 来完成一部分 M 的任务，之前是 M 从任务队列取任务，现在是 P 从任务队列取任务，放到自己的本地队列，当 M 上执行的 G 阻塞时，P 与 M 分离，这个阻塞的 G 仍然和 M 绑在一起继续阻塞等待系统调用返回。那么 P 就可以继续和其他的 M 结合，你看 M 和 G 就解耦了，此时，M 只执行任务，P 只分发任务，解耦了之前的 M 执行任务，又要管理任务的耦合。这时候，M 面对的不是 G 了，M 只需找到一个 P 去结合，然后执行 P 中的 G。

协程核心代码如下：

```

wg := sync.WaitGroup{}
for k, v := range all {
    wg.Add(1)
    go func(value []int, key int) {
        defer wg.Done()
        var tmpPorts []int
        for i := 0; i < len(value); i++ {
            opened := s.isOpen(ip, value[i])
            if opened {
                tmpPorts = append(tmpPorts, value[i])
            }
        }
        mutex.Lock()
        openPorts = append(openPorts, tmpPorts...)
        mutex.Unlock()
        if len(tmpPorts) > 0 {
            s.sendLog(fmt.Sprintf("%v 【%v】协程%v 执行完成，时长： %.3fs，
            开放端口： %v", time.Now().Format("2006-01-02 15:04:05"), ip, key,

```

```
time.Since(start).Seconds(), tmpPorts))  
    }  
    }(v, k)  
}  
wg.Wait()
```

4.2.6 信息存储模块

信息存储模块主要是完成整个模块信息的录入读取，方便管理员后台管理，对扫描信息进行分类整理，以供分析可能存在的网络安全风险，及时发现并解决掉。

工作流程主要是：

- 1、创建目录和日志
- 2、存储全部端口 IP 信息
- 3、读取端口信息，协程调用进行端口扫描
- 4、存储扫描结果

核心函数：

//创建日志目录

```
func Mkdir(path string)
```

//记录日志

```
func (s *ScanIp) sendLog(str string)
```

4.2.7 总结

本章详细介绍了整个系统的设计与实现，围绕信息交互模块、获取 IP 模块、端口扫描模块、信息存储模块讨论了实现原理及细节，给出了具体实现流程，方便读者参考。

第五章 测试与结果分析

5.1 测试实例

本章主要对已完成的 IP 扫描器各功能进行测试，根据具体的测试用例对实验结果进行分析，讨论系统的功能及不足。

在阿里云 ecs 下执行，帮助信息界面如下：



```

Welcome to Alibaba Cloud Elastic Compute Service !


[root@iZz612v4ylpks5Z ~]# cd document/
[root@iZz612v4ylpks5Z document]# ls
gittest  main  scanPort
[root@iZz612v4ylpks5Z document]# ./main -h
scanPort version: scanPort/1.10.0
Usage: scanPort [-h] [-ip ip地址] [-n 进程数] [-p 端口号范围] [-t 超时时长] [-p
ath 日志保存路径]

Options:
  -h      帮助信息
  -ip string
          ip地址 例如：-ip=192.168.0.1-255 或直接输入域名 yhw.com (default "127.0.0
.1")
  -n int
          协程数 例如：-n=10 (default 100)
  -p string
          端口号范围 例如：-p=80,81,88-1000 (default "80-1000")
  -path string
          日志地址 例如：-path=log (default "log")
  -t int
          超时时长(毫秒) 例如：-t=200 (default 200)
  
```

图 5.1 帮助信息

由图 5.1 可知，本系统可在 Linux 下输入提示参数显示帮助信息，方便管理人员执行后续操作，输入参数运行程序。

在 Mac 下执行，域名扫描测试，输入 `./scanPort -ip=www.ip.com -p=80-2000` 命令，结果如下：



```

yhw8930deMacBook-Pro:mac yhw8930$ ./scanPort -ip=www.ip.com -p=80-2000
===== Start 2020-05-13 21:40:01 ip:www.ip.com,port:80-2000 =====
2020-05-13 21:40:01 【127.0.0.1】 需要扫描端口总数:1921 个，总协程:100 个，每个协程处理:20 个
，超时时间:200毫秒
2020-05-13 21:40:04 【127.0.0.1】 协程1 执行完成，时长： 2.678s，开放端口： [80]
2020-05-13 21:40:04 【127.0.0.1】 协程46 执行完成，时长： 2.882s，开放端口： [980]
2020-05-13 21:40:04 【127.0.0.1】 协程47 执行完成，时长： 3.087s，开放端口： [1001]
2020-05-13 21:40:04 【127.0.0.1】 协程28 执行完成，时长： 3.088s，开放端口： [631]
2020-05-13 21:40:04 【127.0.0.1】 扫描结束，执行时长3.088s，所有开放的端口:[80 980 1001 631]
===== End 2020-05-13 21:40:05 总执行时长：4.49s =====
  
```

图 5.2 域名扫描测试

由图 5.2 可知：该域名为 `www.ip.com`，对应主机 IP 是 `127.0.0.1`（即本机），可以扫描到 80、631、980、1001 端口。

结果分析：本系统可以正确获取域名、端口范围参数执行，协程开辟正常，运行速度很快，可以完成指定域名一段端口范围的扫描。

在 Mac 下执行，IP 及特定端口扫描测试，输入 `./scanPort -ip=192.168.0.100 -p=1001,80-990 -path=ip -n=50`，结果如下：

```
yhw8930deMacBook-Pro:mac yhw8930$ ./scanPort -ip=192.168.0.100 -p=1001,80-990 -path=ip -n=50
===== Start 2020-05-13 22:06:22 ip:192.168.0.100,port:1001,80-990 =====
2020-05-13 22:06:22 【192.168.0.100】 需要扫描端口总数:912 个，总协程:50 个，每个协程处理:19
个，超时时间:200毫秒
2020-05-13 22:06:24 【192.168.0.100】 协程1 执行完成，时长: 1.883s，开放端口: [1001 80]
2020-05-13 22:06:24 【192.168.0.100】 协程48 执行完成，时长: 1.894s，开放端口: [980]
2020-05-13 22:06:24 【192.168.0.100】 扫描结束，执行时长1.895s，所有开放的端口:[1001 80 980]
===== End 2020-05-13 22:06:25 总执行时长: 2.90s =====
yhw8930deMacBook-Pro:mac yhw8930$ ls
ip          log          scanPort
yhw8930deMacBook-Pro:mac yhw8930$ cat ip/192.168.0.100_port.txt
2020-05-13 22:05:14 ip:192.168.0.100,开放端口:[1001 80]
2020-05-13 22:06:25 ip:192.168.0.100,开放端口:[1001 80 980]
```

图 5.3 IP 及特定端口扫描测试

由图 5.3 可知：输入 IP 地址为 192.168.0.100，端口为 1001 及 80-900 范围，日志地址为 `ip/192.168.0.100_port.txt`，协程数为 50，可以得到日志和 80、980、1001 端口。

结果分析：本系统可以完成指定 IP 地址及指定端口、端口范围的扫描，可以根据扫描目标自行设置协程数量，日志存储可以自行设定路径，日志可以展示扫描结果共供管理员分析。

在 Mac 下执行，特定网段扫描测试，输入 `./scanPort -ip=192.168.0.90-100 -p=80-2000`，截取部分结果如下：

```
2020-05-13 22:30:08 【192.168.0.99】 扫描结束，执行时长4.075s，所有开放的端口:[]
2020-05-13 22:30:09 【192.168.0.100】 需要扫描端口总数:1921 个，总协程:100 个，每个协程处理:2
0 个，超时时间:200毫秒
2020-05-13 22:30:12 【192.168.0.100】 协程46 执行完成，时长: 3.110s，开放端口: [980]
2020-05-13 22:30:12 【192.168.0.100】 协程1 执行完成，时长: 3.110s，开放端口: [80]
2020-05-13 22:30:12 【192.168.0.100】 协程47 执行完成，时长: 3.112s，开放端口: [1001]
2020-05-13 22:30:12 【192.168.0.100】 扫描结束，执行时长3.112s，所有开放的端口:[980 80 1001]
===== End 2020-05-13 22:30:13 总执行时长: 54.88s =====
```

图 5.4 特定网段扫描测试

由图 5.4 可知：输入 IP 地址范围是 192.168.0.90-100，端口范围是 80-2000，可以得到 192.168.0.100 主机存在，开放端口 80、980、1001。

结果分析：本系统可以完成局域网下特定网段主机的扫描，获取端口信息。

在 Mac 下执行，错误用例测试，输入 `./scanPort -ip=192.168.0.300 -p=80-2000`，结果如下：


```
yhw8930deMacBook-Pro:mac yhw8930$ ./scanPort -ip=192.168.0.300 -p=80-2000  
===== Start 2020-05-13 22:40:45 ip:192.168.0.300,port:80-2000 =====  
192.168.0.300域名解析失败lookup 192.168.0.300: no such host
```

图 5.5 错误用例测试

由图 5.5 可知：输入 IP 地址为 92.168.0.300，返回结果 no such host。

结果分析：本系统可以对输入的错误参数进行处理，给出错误提示。

5.2 测试结果与分析

从以上 5 个测试用例我们可以看出该扫描器可以完成对域名、IP、网段的端口开闭进行扫描，给出扫描结果，扫描速度较快，可以跨平台运行，在实际生产中有很大的应用价值。不足之处是只能获取对应的端口号，由于端口众多且部分服务企业可能会自己更改端口，或者把内网端口映射到外网做了修改，故本系统没有对端口号进行分析，只能由管理员自己根据常规情况来判断，可能会增大管理员的工作强度。

结 论

随着互联网的飞速发展，网络安全问题越来越被人们重视，网络安全扫描是检测网络安全状况的最好方式，它可以提前发现潜在的安全漏洞，使从业人员提前解决，避免可能产生的经济损失。

本文主要实现了一款高性能、跨平台的 IP 网络扫描器，该系统能在实际网络环境下，根据用户需求扫描指定域名网段，判断主机是否存在、TCP/UDP 端口是否开闭，并给出 IP 及端口信息供管理员使用，及时发现网络安全风险并解决。

主要有以下工作：

- 1、结合课题研究背景确定了系统研发的主要内容、意义和目标。
- 2、对所需要的相关技术做了详细介绍，主要是网络扫描技术原理、TCP/IP 协议和 go 协程。
- 3、对系统需求及实现可行性进行分析，确保能够顺利完成。
- 4、设计并实现了 IP 网络扫描器的各项功能。
- 5、对完成的系统进行功能测试，证明系统可用。

由于个人水平有限和时间较为紧张，系统设计还有一些不足之处，在后续的研究中需要进一步解决完善。

- 1、实现分布式扫描。现在大型企业的系统都是分布式，为了满足这种需要本系统要具备分布式扫描的能力，提高扫描效率。
- 2、能够进行客观准确的安全评估。系统应该对扫描出来的漏洞进行整理分类，通过相应的算法进行评估，标注风险等级并及时提示。
- 3、完成安全友好的界面。本系统纯命令行比较适合专业人员使用，需要实现 web 界面以供非专业人员使用，降低技术门槛。

致 谢

时间如白马过隙，眨眼间我已度过 4 年的本科生活。这 4 年，是我不断成长、突破自我、走向成熟的砥砺前行。在此，我向这 4 年来所有的老师、同学表示我最诚挚的感谢！

感谢通工 1608 班的同学，是你们让我在刚来到学校就打开了心扉，在一个陌生的城市陌生的大学开始一段崭新的旅程。

感谢通工 1615 班的学生，是你们让我不断进步、追求卓越，没有荒废大学的时光，我们在各自的路上成长，一起走向优秀。

感谢实验室的所有同学，你们为我指明了前进的方向，从大一到大四，我们一起抛洒汗水，学习进步，重复试验、参加比赛、聚会娱乐，让青春散发耀眼的光芒。

感谢 406 的室友，是你们营造了一个良好的学习生活环境，我们一起度过了 4 个春华秋实，愿我们友谊长存。

感谢 4 年来所有的任课老师，是你们认真负责的教学，让我收获了知识的果实，成为一名合格的通工人。

感谢导员，是你在一些琐事给我提供帮助，让我大学生活无忧，收获了两段充实的实习经历。

感谢毕设老师，在选题和毕设过程中提供的帮助，及时督促我完成毕设，顺利完成设计和论文。

特别感谢我的父母，在这 22 年的生活中，是你们无私的爱与付出才有了现在的我，我永远爱你们！

最后感谢自己，不负韶华，度过美好充实的大学生活！

参考文献

- [1] 李瑞民. 网络扫描技术揭秘: 原理、实践与扫描器的实现[M]. 北京: 电子工业出版社, 2012
- [2] 谢希仁. 计算机网络[M]. 北京: 电子工业出版社, 2017
- [3] 凯文 R. 福尔 (Kevin R. Fall). TCP/IP 详解卷 1: 协议原书第 2 版[M]. 北京: 机械工业出版社, 2016
- [4] 段广丽. 计算机网络安全漏洞防范探析[J]. 信息与电脑(理论版). 2017(08)
- [5] 任波. 计算机网络安全与漏洞扫描技术的应用研究[J]. 信息与电脑(理论版), 2019, 24
- [6] 张文海. 网络安全漏洞扫描技术研究[J]. 福建电脑, 2011, 10
- [7] 翟涵. 基于网络爬虫的 Web 安全扫描工具的设计与实现[D]. 北京: 北京邮电大学, 2018
- [8] 王俊. 面向 web 系统的安全信息搜集平台的设计与实现[D]. 北京: 北京邮电大学, 2017
- [9] 吴倩倩. 综合型漏洞扫描系统的研究与设计[D]. 北京: 华北电力大学, 2015
- [10] 陈家东. 网络安全扫描系统实现技术研究[D]. 武汉: 华中科技大学, 2007
- [11] 杨忠仪. 网络安全扫描系统关键技术的研究与实现[D]. 湖南: 国防科学技术大学, 2007
- [12] 张长智. 基于漏洞扫描技术的网络安全风险评估[D]. 四川: 电子科技大学, 2009
- [13] 杨博文. 网络漏洞扫描关键技术研究[D]. 四川: 电子科技大学, 2019
- [14] 龚小刚. 网络漏洞扫描技术研究[C]. 中国电子学会第十七届信息论学术年会, 2010
- [15] 胡惊涛, 李华波, 陈刚. 网络安全扫描技术研究[C]. 第十三届全国青年通信学术会议
- [16] Musthajer,Linda. Pwnie Express makes vulnerability scanning of remote sites as simple as plug-and-play[J]. Network World (Online). 2014
- [17] Teodor,Almroth, Jonas,Persson, Mats. A quantitative evaluation of vulnerability scanning[J]. EN. 2011 (4)
- [18] Y.SU,X.F.LI,S.F.WANG. Vulnerability Scanning System Used in the Internet of Things for Intelligent Devices[DB]. 2017
- [19] Li Junyi , Su Fei , Lin Zhaowen et al. The research and analysis of worm scanning strategies in IPv6 network[C]. 2011 13th Asia-Pacific Network Operations and Management Symposium, 2011
- [20] Shashi Shaw ; Prasenjit Choudhury. A new local area network attack through IP and MAC address spoofing[C]. 2015 International Conference on Advances in Computer Engineering and Applications, 2015
- [21] Rodney Rohrmann ; Mark W. Patton ; Hsinchun Chen. Anonymous port scanning: Performing network reconnaissance through Tor[C]. 2016 IEEE Conference on Intelligence and Security Informatics (ISI), 2016