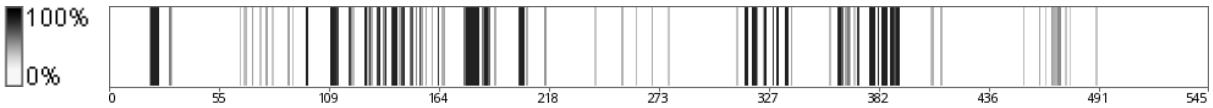
PaperPass旗舰版检测报告
简明打印版

比对结果(相似度):

总　体：24% (总体相似度是指本地库、互联网的综合对比结果)
本地库：13% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库：10% (期刊库相似度是指论文与学术期刊库的比对结果)
学位库：10% (学位库相似度是指论文与学位论文库的比对结果)
会议库： 3% (会议库相似度是指论文与会议论文库的比对结果)
图书库： 7% (图书库相似度是指论文与图书库的比对结果)
互联网：17% (互联网相似度是指论文与互联网资源的比对结果)

报告编号：5EBE466C8A35EB2YN
检测版本：旗舰版
论文题目：IP网络扫描技术的研究与实践
论文作者：杨海威
论文字数：15578字符(不计空格)
段落个数：294
句子个数：545 句
提交时间：2020-5-15 15:36:12
比对范围：学术期刊、学位论文、会议论文、书籍数据 、互联网资源
查询真伪：http://www.paperpass.com/check

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

1.相似度： 3% 篇名：《网络漏洞扫描技术研究》
来源：学术期刊 《福建电脑》 2008年5期
2.相似度： 3% 篇名：《网络安全漏洞扫描系统的设计与实现》
来源：学位论文 国防科学技术大学 2007
3.相似度： 2% 篇名：《一种分布式脆弱性检测技术的研究》
来源：学位论文 北京邮电大学 2010
4.相似度： 2% 篇名：《分布式漏洞扫描系统的设计与实现》
来源：学位论文 北京邮电大学 2005
5.相似度： 1% 篇名：《网络安全技术》
来源：书籍数据 西安电子科技大学出版社 2007-2-1
6.相似度： 1% 篇名：《网络漏洞扫描技术研究》
来源：会议论文 2010-09-01
7.相似度： 1% 篇名：《信息对抗理论与方法》
来源：书籍数据 武汉大学出版社 2008-08-01
8.相似度： 1% 篇名：《端口服务及版本探测的研究》
来源：学术期刊 《科技创业月刊》 2007年7期
9.相似度： 1% 篇名：《网络信息对抗》
来源：书籍数据 北京邮电大学出版社 2011-01-01

10.相似度： 1% 篇名：《基于网络的漏洞扫描系统的设计与实现》
来源：学位论文 北京邮电大学 2010
11.相似度： 1% 篇名：《计算机网络信息发现技术研究》
来源：学位论文 哈尔滨工业大学 2005
12.相似度： 1% 篇名：《计算机网络与信息安全》
来源：书籍数据 哈尔滨工业大学出版社 2008-09-01
13.相似度： 1% 篇名：《基于信息融合的网络安全评估方法研究》
来源：学位论文 西安电子科技大学 2009
14.相似度： 1% 篇名：《浅析网络安全扫描技术》
来源：学术期刊 《科技情报开发与经济》 2005年1期
15.相似度： 1% 篇名：《利用Matlab处理Excel文件》
来源：学术期刊 《科技情报开发与经济》 2005年1期
16.相似度： 1% 篇名：《基于TCP协议的端口扫描技术》
来源：学术期刊 《电脑开发与应用》 2011年1期
17.相似度： 1% 篇名：《网络漏洞扫描原理分析》
来源：学术期刊 《福建电脑》 2009年9期
18.相似度： 1% 篇名：《网络扫描技术实现及其在网络安全中的应用》
来源：学术期刊 《计算机应用研究》 2004年2期
19.相似度： 1% 篇名：《远程操作系统探测与防护技术研究》
来源：学位论文 上海交通大学 2004

互联网相似资源列表：

1.相似度： 4% 标题：《网络扫描技术总结_whatday的专栏-CSDN...》
https://blog.csdn.net/whatday/article/details/85198323
2.相似度： 3% 标题：《网络扫描技术总结_amosilin的博客-CSD...》
https://blog.csdn.net/amosilin/article/details/51084804
3.相似度： 3% 标题：《《快学 Go 语言》第 11 课 —— 千军万马...》
https://blog.csdn.net/shellquery/article/details/100892903
4.相似度： 3% 标题：《Go学习笔记-协程与通道 - 简书》
https://www.jianshu.com/p/bda3d33d531d
5.相似度： 3% 标题：《《快学 Go 语言》第 11 课 —— 千军万马...》
https://blog.csdn.net/codehole_/article/details/100892469
6.相似度： 2% 标题：《goroutine原理分析 - 安得情怀似旧时》
https://blog.csdn.net/KentZhang_/article/details/83926533
7.相似度： 2% 标题：《goroutine原理分析 - 程序员大本营》
https://www.pianshen.com/article/773274695/
8.相似度： 2% 标题：《goroutine原理分析_安得情怀似旧时-CS...》
http://m.blog.csdn.net/blog/KentZhang_/83926533
9.相似度： 2% 标题：《go语言之行--golang核武器gorouti...》
https://blog.csdn.net/u014230625/article/details/86520568
10.相似度： 2% 标题：《理解Go协程与并发_Linux编程_Linux公...》
https://www.linuxidc.com/Linux/2019-08/160189.htm
11.相似度： 2% 标题：《理解Go协程与并发 - weixin_30371...》
https://blog.csdn.net/weixin_30371875/article/details/101657777
12.相似度： 2% 标题：《理解Go协程与并发 - 飞鸿影 - 博客园》
https://www.cnblogs.com/52fhy/p/11369028.html
13.相似度： 2% 标题：《Go语言学习-goroutine - 匠心独运 ...》
https://blog.csdn.net/johnWcheung/article/details/95767812
14.相似度： 2% 标题：《GO的并发之道-Goroutine调度原理&Ch...》
https://www.cnblogs.com/X-knight/p/11363730.html
15.相似度： 1% 标题：《计算机网络 - 网络层_网络_NayelyA的博...》
https://blog.csdn.net/weixin_40992982/article/details/105866856
16.相似度： 1% 标题：《「计算机网络」网络层协议总结_网络协议_Code...》
https://blog.csdn.net/weixin_43359179/article/details/105922444

17.相似度： 1% 标题：《goroutine与调度器 - Tony的专栏...》
https://blog.csdn.net/zdq0394123/article/details/17038631
18.相似度： 1% 标题：《浅谈goroutine - 简书》
https://www.jianshu.com/p/7ebf732b6e1f
19.相似度： 1% 标题：《goroutine与调度器 - davygeek...》
https://www.cnblogs.com/davygeek/p/7644625.html
20.相似度： 1% 标题：《Goroutine的调度_Mr.Phoebe的专...》
https://blog.csdn.net/u013007900/article/details/89116780
21.相似度： 1% 标题：《计算机网络-应用层 - Goinhn - CSD...》
https://blog.csdn.net/Goinhn/article/details/89717100
22.相似度： 1% 标题：《利率市场化对我国货币政策传导机制影响研究-经济1...》
https://max.book118.com/html/2018/0811/8041103123001117.shtm
23.相似度： 1% 标题：《Goland 使用_Goland，IDE，Go_...》
https://blog.csdn.net/william_n/article/details/105888166
24.相似度： 1% 标题：《IP 地址编址方式（分类、子网划分、无分类） -...》
https://blog.csdn.net/Zhxin606a/article/details/89389132
25.相似度： 1% 标题：《JetBrains GoLand 2019 fo...》
https://www.jianshu.com/p/5ee77626f821

全文简明报告:

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

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： 2019年 11月16日 至 2020年 5月14日

毕业设计（论文）声明书

{56%：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，} {96%：论文中所引用他人的文献、数据、图件、资料均已明确标注；} {100%：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。}

{95%：本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。}

{100%：本人深知本声明书的法律责任，违规后果由本人承担。}

论文作者签名： 杨海威

日期： 2020 年 5 月 14日

摘 要

{61%：近年来，随着互联网技术的蓬勃发展，网络与人们的生活愈加紧密相关，} {40%：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。

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

{43%：随着互联网技术的发展，网络与人们的生产生活联系愈加紧密，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。 {43%：作为一个信息存储于流通的平台，网络的安全对企业能否成功有着决定性的作用，此时保证信息安全就是重中之重。}

{46%：IP网络扫描技术是一种基于 TCP/IP等协议对目标主机扫描，以发现目标主机是否有安全漏洞的技术。} 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， {53%：才能更好的完成网络安全防护工作，保证企业经济效益。}

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， {41%：本文就详细介绍了该扫描器的设计原理及具体过程，以保障网络的可靠安全。}

第一章 概述

1.1课题研究背景

{51%：21世纪互联网技术蓬勃发展，与人们的生活联系愈加紧密，各种互联网+概念层出不穷。} 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。 {40%：如何防范网络攻击、杜绝信息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， 在实际生产环境中有较大的应用价值。

1.2 系统研发主要内容、意义及目标

{51%：通过上述背景可知，IP网络扫描技术对于网络安全有着重大意义。} 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， {43%：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：

扫描特定域名判断其是否存在

扫描特定网段范围主机的工作状态

识别目标主机的端口状态

{90%：保存漏洞信息，分析系统脆弱点}

生成扫描结果日志以供分析

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：

自动垃圾回收降低了开发难度

更丰富的内置类型

支持函数多返回值

简洁的类型和“非侵入式”接口

并发编程更轻盈更安全

{96%：GoLand是一款专为Go开发人员构建的跨平台IDE，功能非常强大，拥有强大的代码洞察力，}　{93%：帮助所有开发人员即时错误检测和修复建议，快速和安全的重构，一步撤销，智能代码完成，}　{88%：死代码检测和文档提示，创建快速，高效，可靠的代码。}　{80%：GoLand代码可帮助我们快速切换到类型实现的阴影方法，实现，用法，声明或接口，更加高效的开发。}

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

{80%：网络扫描技术就是通过端口向指定网段主机的　TCP/IP　服务端口发送探测数据包，记录目标主机的响应情况。}　{53%：通过分析是否响应来判断相关端口、即进程是打开还是关闭，由此可以得知目标主机端口对应的服务情况，发现潜在的漏洞。}　{43%：IP扫描也可以通过截取网络传输过程中IP数据包来监视对应系统的运行情况，对相对应的数据进行分析，获取目标的网络安全漏洞。}

2.1.2常见扫描技术

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。

TCP 协议（端口扫描）

{92%：传输控制协议 TCP（Transmission Control Protocol），位于运输层，是一种面向连接的可靠服务，在传送数据之前必须先建立连接，数据传送结束后要释放连接。}{46%：扫描器主要利用TCP协议的三次握手和四次挥手、 在标志位实现端口扫描。}

{85%：当确定了目标主机可已访问后，就可以使用端口扫描技术，确定目标主机的开放端口，} {48%：包括各种网络协议和应用监听的端口，以此判断对应服务是否开启。}

开放扫描技术： 主要是TCP Connect扫描

{81%：TCP Connect扫描是通过调用socket函数connect()连接到目标主机上，实现一次完整的三次握手过程。} {90%：如果该端口处于监听状态，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。 {58%：它具有稳定可靠，不需要特殊的权限的优点，但同时由于扫描方式不够隐蔽，} {97%：服务器日志会记录下大量密集的连接和错误记录，容易被防火墙发现和屏蔽。}

半开放扫描技术

TCP SYN扫描

{82%：该扫描方式是向目标主机发送SYN包，如果应答是RST包，那么端口是关闭的；}{96%：如果应答中包含SYN和ACK包，说明目标端口属于监听状态，然后传送一个RST包给目标主机从而停止建立连接。} {97%：由于SYN扫描事，全连接尚未建立，所以这种技术通常被称为半连接扫描。}

TCP间接扫描

{47%：该方式模拟第三方IP欺骗目标主机来隐藏真实IP。} {79%：由于本机会对目标主机发送应答信息，所以必须监控目标主机的IP行为，从而获得原始的扫描结果。} {96%：本机通过伪造第三方主机IP地址向目标主机发起SYN扫描，并通过观察其IP序列号的增长规律获取端口的状态。}

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， {64%：一般是几种扫描机制的综合使用，类unix下一般采用间接扫描 +（TCP FIN或TCP Xmas 或TCP Null）+ 分段扫描的方式，}{75%：具有隐蔽性好，可穿越防火墙的特点。}

2.2 TCP/IP协议简介

{60%：TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/网际协议）是指能够在多个不同网络间实现信息传输的协议簇，} 它主要包括TCP、UDP、IP等。

{82%：TCP即传输控制协议，是一种面向连接的、可靠的、基于字节流的通信协议，}{50%：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}

图2.1 TCP与UDP区别

{51%：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

{49%：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

{86%：建立TCP连接时，需要客户端和服务器共发送3个包：}

1、客户端发送初始序号x和syn=1请求标志。

{47%：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seq=y，发送客户端的确认序号ack=x+1。}

{50%：3、客户端发送ACK确认号，发送自己的序号seq=x+1，发送对方的确认号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

{66%：Go语言作为并发编程语言最大的特色是协程和管道。} {100%：一个进程内部可以运行多个线程，而每个线程又可以运行很多协程。} {100%：线程要负责对协程进行调度，保证每个协程都有机会得到执行。} {100%：当一个协程睡眠时，它要将线程的运行权让给其它的协程来运行，而不能持续霸占这个线程。} {100%：同一个线程内部最多只会有一个协程正在运行。}

{100%：协程可以简化为三个状态，运行态、就绪态和休眠态。} {100%：同一个线程中最多只会存在一个处于运行态的协程，就绪态的协程是指那些具备了运行能力但是还没有得到运行机会的协程，} {100%：它们随时会被调度到运行态，休眠态的协程还不具备运行能力，它们是在等待某些条件的发生，} 比如 IO 操作的完成、睡眠时间的结束等。

{63%：Go 语言实现了一种非抢占式的调度，运行时调度器采用了 work-stealing 算法，} {95%：当某个线程空闲时，也就是该线程上所有的协程都在休眠（或者一个协程都没有），} {100%：它就会去其它线程的就绪队列上去偷一些协程来运行。} {71%：也就是说这些线程会主动找活干，在正常情况下，运行时会尽量平均分配工作任务，从而使CPU最

高效运转，提高了系统并发能力。}

第三章 系统需求分析

3.1系统可行性分析

{48%：可行性分析是一个项目设立之初最重要的事情，它是整个项目成败的关键。} 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。

{81%：使用GoLand作为开发环境，GoLand是JetBrAIns的新商业IDE，旨在为Go开发提供符合人体工程学的环境。} {100%：新的IDE通过特定于Go语言的编码辅助和工具集成扩展了IntelliJ平台。} {91%：编码协助IDE分析代码，查找符号之间的连接，提供代码完成，快速导航，巧妙的错误分析，} 格式化和重构，在代码的模块化与后期完善上十分方便。

{48%：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。 {55%：在设计系统时需要满足基本的设计原则：}

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

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

{40%：系统采用模块编程，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：

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。

获取IP模块： 获取所有IP，完成域名解析、IP合法验证等功能。

端口扫描模块： 判断端口是否合法和开闭，过滤重复端口。

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：

提示帮助信息： {46%：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。

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。

初始化： 初始化结构体和相应参数。

获取所有IP： {44%：第一步判断是否是域名，如果是则将域名解析成为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。

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。

显示结果： {49%：控制台显示相关信息以供管理员使用。}

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。 {40%：若果没有设置则会采用默认值，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。

图4.3 提示信息

ip: 输入域名或者IP网段，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

{41%：整个主模块的逻辑十分清晰，下面是主模块的运行逻辑图：}

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。

{100%：DNS是一个分布式数据库，提供了主机名和 IP 地址之间相互转换的服务。}{100%：这里的分布式数据库是指，每个站点只保留它自己的那部分数据。} 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。 {100%：DNS 可以使用 UDP 或者 TCP 进行传输，使用的端口号都为 53。} {98%：大多数情况下 DNS 使用 UDP 进行传输，这就要求域名解析器和域名服务器都必须自己处理超时和重传来保

证可靠性。} {57%：在两种情况下会使用 TCP 进行传输，返回字节大于512或区域传送。} 整个解析过程如下：

图4.5 DNS解析过程

{52%：通过DNS解析可以将域名解析为对应的IP地址，IP地址由两部分组成，} {100%：网络号和主机号，其中不同分类具有不同的网络号长度，并且是固定的。}

IP 地址 : : = {< 网络号 >, < 主机号 >}

{84%：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，{98%：即通过在主机号字段中拿一部分作为子网号，把两级 IP 地址划分为三级 IP 地址。}

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

{100%：要使用子网，必须配置子网掩码。} {100%：一个 B 类地址的默认子网掩码为 255.255.0.0，如果 B 类地址的子网占两个比特，} 那么子网掩码为 11111111 11111111 11000000 00000000，也就是 255.255.192.0。

{50%：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])
//验证IP地址
net.ParseIP(firstIp.String())
//获取所有IP
for i : = 1; i < totalIp; i++ { ips = append(ips, fmt.Sprintf("%s.%s.%s.%d", ipTmp2[0], ipTmp2[1], ipTmp2[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

{43%：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，核心流程如下：

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。

{87%：ICMP协议能够更有效地转发 IP 数据报和提高交付成功的机会。} {100%：它封装在 IP 数据报中，但是不属于高层协议。} {71%：ICMP 报文分为差错报告报文和询问报文，如图4.8。}

图4.8 常见ICMP报文类型

{50%：例如常见的ping命令，就是向目标主机发送ICMP数据包，等待应答。} {62%：如果能收到，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。}

{44%：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。

{41%：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。

{83%：groutine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。}

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: {100%：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；} {99%：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、随机数发生器等等非常多的信息 G：} {100%：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: {99%：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: {100%：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。}

{100%：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，} {100%：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。

{100%：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从任务对列取任务，} {100%：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。} {77%：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，} {100%：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。} {100%：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

{41%：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，} {47%：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

{46%：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：

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：

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：

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：

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：

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： {44%：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，在企业的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

{50%：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，{40%：根据用户需求扫描指定域名、网段主机的存在性、TCP/UDP端口的开闭性，} 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：

{54%：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。}

{47%：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。}

{46%：对系统需求及实现可行性进行分析，确保能够顺利完成。}

{58%：设计并实现了IP网络扫描器的各项功能。}

{58%：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。

实现分布式扫描。 {40%：现在大型企业的系统都是分布式，本系统需要结合前沿的分布式技术进行分布式扫描以提高扫描效率。}

能够进行客观准确的安全评估。 {40%：系统应该根据相应的算法对扫描出来的漏洞进行整理分类和评估，标注风险等级并及时提示。}

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，眨眼将我已度过4年的本科生活。 这4年，是我不断成长、突破自我、走向成熟的砥砺行程。 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。

{41%：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] Musthaler，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