

摘要

互联网飞速发展几十年，已经成为了一个新世界，而这个新世界的发展必然要有新的治理办法。如今越来越多的工业设备、家庭设备接入到这个庞大的网络世界中，但是由于硬件、软件或者管理员后期维护等造成的一些安全缺陷使系统容易遭受到不法分子的攻击和利用，同时给社会带来直接或间接的损害，甚至可能会威胁到国家安全。而且，目前互联网上大部分设备都是基于 IPv4 协议下，且大部分运行中的硬件和软件都是已知的。因此，我们可以通过此网络空间搜索引擎对暴露在互联网上的设备进行扫描，并且对软件服务的类型进行识别，然后将识别后的结果写入数据库，并且提供给网络安全人员进行搜索。网络安全人员则可以通过本系统，可以查看数据库中存在的信息，并且列出某 IP 地址在数据库中的所有历史扫描结果，而且还能通过该引擎发现某些已知的安全漏洞。如果某系统出现了新的漏洞，通过引擎可以知道有多少系统正在遭受该漏洞的威胁。在本系统中,笔者优化了端口扫描和 IP 地址存储数字化，实现了中文检索，提供了一个相对比较有好的搜索界面，使网络安全人员更容易使用。

关键词：网络空间搜索引擎；shodan；分布式网络扫描；物联网搜索引擎； zoomeye

ABSTRACT

The rapid development of the Internet for decades has become a new world, and it will inevitably need some new ways of governance. Nowadays more and more industrial equipment and home equipment are connected to this huge network world, but some security flaws caused by hardware, software, or administrators' post-maintenance make the system vulnerable to be attacked and exploited by criminals, which may directly or indirectly cause damage to society, and even threaten national security. Moreover, since most of the devices on the Internet are currently based on the IPv4 protocol, and most of the running hardware and software are known, we can scan the devices exposed on the Internet through this cyberspace search engine. And identify the type of software service, then write the identified results to the database and provide them to the network security personnel for searching. Network security personnel can use this system to view the information existing in the database, and list all historical scan results of an IP address in the database, and find some known security vulnerabilities. In other words, if a new vulnerability occurs in a system, the engine can know how many systems are being threatened by the vulnerability. In this system, the author optimizes port scanning and IP address storage digitization, realizes Chinese search, and provides a relatively good search interface, making network security personnel easier to use.

Key Works: cyberspace search; shodan; network scan; IoT security; zoomeye

目录

摘要.....	I
ABSTRACT.....	II
目录.....	III
第 1 章 绪 论.....	1
1.1 课题研究背景及意义.....	1
1.2 与普通搜索引擎区别.....	1
1.2.1 普通搜索引擎.....	1
1.2.2 网络空间搜索引擎.....	2
1.3 国内外研究现状.....	2
1.4 研究内容.....	3
第 2 章 系统开发工具及技术.....	4
2.1 开发环境说明.....	4
2.2 系统关键技术分析.....	4
2.2.1 端口扫描.....	4
2.2.2 服务识别.....	5
2.2.3 全文搜索.....	6
2.2.4 Django MTV 模式.....	7
2.2.5 消息队列.....	7
第 3 章 系统分析.....	8
3.1 可行性分析.....	8
3.1.1 社会可行性.....	8
3.1.2 技术可行性.....	8
3.1.3 运用可行性.....	8
3.2. 功能需求分析.....	8
3.2.1 搜索模块.....	9
3.2.2 任务管理模块.....	9
3.2.3 扫描模块.....	9
3.3 非功能性需求.....	10
第 4 章 系统设计.....	11
4.1 系统框架设计.....	11
4.2 系统功能模块设计.....	12

4.2.1 搜索模块	14
4.2.2 扫描模块	14
4.3 系统数据库设计	15
第 5 章 系统详细设计与实现	17
5.1 项目结构	17
5.2 管理员模块	17
5.2.1 登录功能	17
5.2.3 添加端口任务	18
5.2.3 添加服务扫描任务	18
5.2.4 清空任务	19
5.2.5 更新统计数据	19
5.3 搜索模块	20
5.3.3 首页展示	20
5.3.1 搜索功能	20
5.3.2 单 IP 结果显示	21
5.4 扫描模块	22
5.4.1 端口扫描	22
5.4.2 服务扫描	22
5.4.3 数据同步	23
5.4.3 索引建立	23
第 6 章 系统测试	24
6.1 测试环境	24
6.2 搜索模块测试	24
6.3 添加端口扫描任务模块测试	25
6.4 添加服务扫描任务模块测试	25
6.4 统计数据更新	26
第 7 章 结论	27
参考文献	28
致谢	29

第1章 绪论

1.1 课题研究背景及意义

随着互联网的迅速发展及普及，互联网已经成了人们生活中不可分割的一部分。但互联网也面临着相当严峻的网络安全问题，2018年四月二十日习近平总书记在全国网络安全和信息化工作会议上强调：“没有网络安全就没有国家安全，就没有经济社会稳定运行，广大人民群众利益也难以得到保障。要树立正确的网络安全观，加强信息基础设施网络安全防护，加强网络安全信息统筹机制、手段、平台建设，加强网络安全事件应急指挥能力建设，积极发展网络安全产业，做到关口前移，防患于未然。要落实关键信息基础设施防护责任，行业、企业作为关键信息基础设施运营者承担主体防护责任，主管部门履行好监管责任。要依法严厉打击网络黑客、电信网络诈骗、侵犯公民个人隐私等违法犯罪行为，切断网络犯罪利益链条，持续形成高压态势，维护人民群众合法权益^[1]”。网络空间是广大人民群众的精神家园，网络空间的安全关乎国家长治久安和最广大人民群众的根本利益。

在网络安全工作人员进行网络安全工作时，经常需要对网络服务器设备经常扫描，甚至有时候需要一些端口的历史数据进行分析。如：在企业威胁情报分析人员对某个IP进行威胁情报分析时需要对服务器的历史数据进行分析，来判断目标IP端口的历史变化是否为恶意IP，以便加快工作进程。当然网络空间搜索引擎的用途远远不止这些。本文的设计意在研究网络空间引擎的原理，与同仁交流一下对于网络安全的一些经验，贡献自己一份微薄的力量，为网络安全行业添砖加瓦。

1.2 与普通搜索引擎区别

由于网络空间搜索引擎与普通搜索引擎与网络空间搜索引擎面向的用户群里不一样，所以其工作方式也有很大的区别。普通搜索引擎作为普通用户人群所以功能简单，但是由于网络空间搜索引擎面向的是网络安全人员，所以会有一些特定的搜索语法。下面是对普通搜索引擎与网络空间搜索引擎的对比。

1.2.1 普通搜索引擎

普通搜索引擎在人们日常生活中使用频率非常高如：百度、谷歌、必应等，可以说搜索引擎是网民们进入互联网这个大世界的主入口。少了搜索引擎网民们就只能在互联网世界里乱撞。普通搜索引擎主要是通过爬虫程序将种子链接的网页内容读取出来，然后分析出新的链接进入队列工爬虫继续爬取。从大体结构上来看普通搜索引擎主要分为如下三个部分：

- 1) 信息采集(信息采集工作主要是通过超链接不断获取新的网页信息)。

- 2) 索引建立(主要通过全文索引技术对信息采集的结果进行索引建立)。
- 3) 搜索服务(给用户一个搜索的入口，通过关键字进行检索)。

1.2.2 网络空间搜索引擎

网络空间搜索引擎主要是通过扫描互联网上所有公网 IPv4 地址的全部端口进行扫描，然后将开放的端口通过服务识别程序进行服务识别(包括：操作系统、应用服务等)，将扫描的结果信息存入数据库，将不同的字段建立索引供用户搜索。网络空间搜索引擎大体结构分为如下四个部分：

- 1) 端口扫描
- 2) 服务识别
- 3) 索引建立
- 4) 搜索服务

1.3 国内外研究现状

目前市面上优秀的闭源网络空间搜索引擎有很多如：国外的 Shodan、Censys、国内的 ZoomEye(钟馗之眼)、Fofa(佛法搜)等。对比目前市面上的网络空间搜索引擎都大同小异，主要是对网络上开放的网络设备进行扫描，对结果入库建立索引。都可以对服务信息、端口等信息进行检索。下面是对目前市面上几个主流网络空间搜索引擎的说明与对比^[2]。

Shodan: 全球最早的开放式网络空间搜索引擎，主要针对 IPv4 地址下的服务器、网络摄像头、交换机路由器等基础网络设施进行扫描。提供了基于 HTTP 协议的 API 接口。

Censys: 该项目起源主要用于学术研究，由密歇根大学和 Rapid7 公司共同完成。该项目全部免费，censys 不仅支持 IPv4 扫描还支持对域名的证书进行扫描。但是在数据可视化上用户体验不佳。

Zoomeye: 该项目由北京知道创宇发出，支持 IPv4 和域名扫描，可以对网页内容进行搜索支持中文。是国内首个网络空间搜索引擎其对影响范围大。属于商业项目，对免费用户在搜索上有诸多限制。

Fofa: Fofa 是网络空间搜索引擎中的后起之秀，基础功能上与前面相同。还提供了用户自己提交插件扫描的功能，在此基础上对界面进行细致的优化大大提升了用户体验友好度。

这三款网络空间搜索引擎虽然都提供了检索功能但是系统的灵活性不够，不能根据用户的自身需求进行扫描。除 censys 之外其余三款对于免费用户在进行查询时有诸多的限制。

1.4 研究内容

本论文意在研究网络空间搜索引擎的原理，提升网络空间搜索引擎在端口扫描和服务识别上面的工作效率，并在此基础给用户设计一个相对友好的交互界面。设计并实现其分布式端口扫描、分布式服务扫描、内容检索的核心部件。在完成每一个小功能是对其模块进行单元测试并且在完成本论文既定功能是对整个网络空间搜索引擎进行基础测试，系统上线时所使用的软件尽量使用最新版或稳定版以减少已知漏洞对系统可能造成的威胁，从而保障网络空间搜索引擎上线运行时的稳定性与安全性。

在本论文第二章对本项目所使用到的关键技术进行了一一的罗列。第三章从经济可行、性技术可行性、等方面进行了详细的分析。并对本项目的功能需求进行了详细的划分和介绍。第四章对整个项目的框架设计做了概要的说明。第五章对整个系统的详细设计与实现过程做了详细说明展示。第六章说明了整个系统的测试工作的过程和结果。第七章是对本系统的所有设计与开发工作的总结。

第 2 章 系统开发工具及技术

2.1 开发环境说明

本系统整个开发过程中业务代码主要使用 Python3 进行编写，python3 是一门高级解释型语言，具有跨平台可移植的特性，语法简单明了可以快速上手。网页交互上主要使用 HTML5+CSS3+JavaScript 进行编写。

网站部分在 windows10 系统下进行开发，开发 IDE 使用的 Sublime Text。开发语言安装 Python3.5.4，并且通过 Python 自带的 pip 工具安装如下库：Django==2.1.7、pymysql==0.9.3、pika==0.13.0、IPy==0.83。

扫描部分运行在 Linux 系统下，开发 IDE 使用的 Jupyter notebook，Jupyter 是一个基于 Web 进行交互的编辑器支持多种语。因为扫描部分主要是网络 IO 操作，所以在 Linux 系统下运行效率要远优于 Windows 系统。开发语言安装 Python3.5.4 并且安装如下库：pika==0.13.0、pymongo、pymysql==0.9.3、IPy==0.83、qqwry-python3、libnmap^[3]。

2.2 系统关键技术分析

本小结主要对系统中的核心技术做一些简要讲解，并对其优点做了分析和对比。以便读者能更好的了解本系统的技术选型思路。

2.2.1 端口扫描

尝试在一台机器上连接到远程计算机上一系列的端口通常称为端口扫描。端口扫描作为整个扫描模块的开始，之后服务识别模块数据的唯一来源。端口扫描是对暴露在公网 IPv4 的所有端口(1~65535)进行扫描。

此功能的技术难点在面向整个互联网的 IPv4 地址进行扫描，如此大规模的扫描任务，对端口扫描的效率性能也是非常高的。为了将端口扫描的效率达到最佳，扫描部分主要使用 masscan^[4]程序作为支持。Masscan 是一种基于无状态扫描技术的工具，无状态扫描技术相比于传统基于 TCP 全连接端口扫描耗时更短、而且准确率出入不大。将传统基于 TCP 全连接扫描的 nmap 工具与无状态扫描技术的工具 Zmap 进行对比结果如图 2.1 是系统性能对比图。

网段	扫描工具	结果数量 / 个	耗时 / ms
144.76.0.0 / 24 单端口	本系统	90	11 091
	nmap	90	64 810
144.76.8.0 / 23 单端口	本系统	169	11 289
	nmap	169	206 990
144.76.0.0 / 24 多端口	本系统	295	26 584
	nmap	306	431 580
144.76.8.0 / 23 多端口	本系统	488	26 585
	nmap	496	799 280
144.76.0.0 / 24 多端口、	本系统	760	27 675
144.76.8.0 / 23 多端口	nmap	782	2 102 950

图 2.1 系统性能对比图^[5]

2.2.2 服务识别

服务识别就是对开放端口上运行的应用程序或服务进行识别，识别出端口对应的应用程序能大大提高网络安全工作者的工作效率。在服务识别上使用 nmap^[6]工具进行辅助，nmap 是一款强大的网络扫描工具它可以侦测出操作系统与设备的型号信息，还可以自定义脚本等多种扫描方式。下表 2.1 是常见端口对应的协议。

强化了 HTTP 协议扫描，因为在所有应用程序网络协议中 HTTP 作为使用在为广泛的一种网络协议，其下面包含了很多种应用程序，所以在 nmap 扫描的基础上本系统还增强了 HTTP 协议扫描，使用 GET 请求获取 HTTP 状态码、HTTP body 以及返回为 HTTP head。表 2.1 是常用协议表。

表 2.1 常用协议表

端口号	协议
21	FTP(文件传输协议)
22	SSH(安全登录)
23	Telnet(远程登录服务)
25	SMTP(简单邮件传输协议)
53	DNS(域名系统)
80	HTTP(超文本传输协议)
443	SSL(通过 SSL 可以使 HTTP 传输更安全)
1433	MSSQL(微软数据库)
3306	MySQL(数据库系统)
3389	RDP(远程桌面控制服务)
8080	HTTP(常用于超文本传输协议)

2.2.3 全文搜索

本系统使用全文检索对系统的核心数据做组织和管理，使用全文搜索引擎可以在数据量比较大时对非格式化的数据进行搜索。比较成熟的全文检索方案如：ElasticSearch^[7](以下简称 ES)。ES 是基于 Lucene 的分布式全文搜索引擎，ES 可以在大规模的数据集中对用的请求做出快速的响应。ES 有强大的数据检索能力主要有以下特点：

- 1) 支持分布式文件存储、和实时数据搜索。
- 2) 有丰富的客户端(C#、Java、Python 等)还有 HTTP Rest 风格的接口。
- 3) 横向可扩展，只需要增加一台服务器做好配置，启动就可以进入集群状态。
- 4) 可以对系统添加分词。

本系统使用的 ES6.6.1 版，在 ES 现有功能的基础上使用 IK 插件优化了全文检索功能对中文的支持。IK 分词插件^[8]可以自定义字典，是用户在搜索时有更好的用户体验，能更准确的返回搜索结果。

2.2.4 Django MTV 模式

Django 是一个基于 Python 语言开发的 web 框架，此框架采用了 MTV 模式，期本质上与 MVC 是一样的，也是为了各部件之间保持松耦合的关系，只是定义上有些许的不同。

M(Model)模型：主要负责对象与数据库之间的映射(ORM)

T(Template)模板：负责展示页面给用户

V(View)视图：负责业务逻辑，并在需要的时候调用 Model 和 Template。

除了上述之外，Django 框架还有强大的路由机制。它可以将用户请求的 URL 分发给不同的视图处理。如图 2.2 所示 Django 工作模型图。

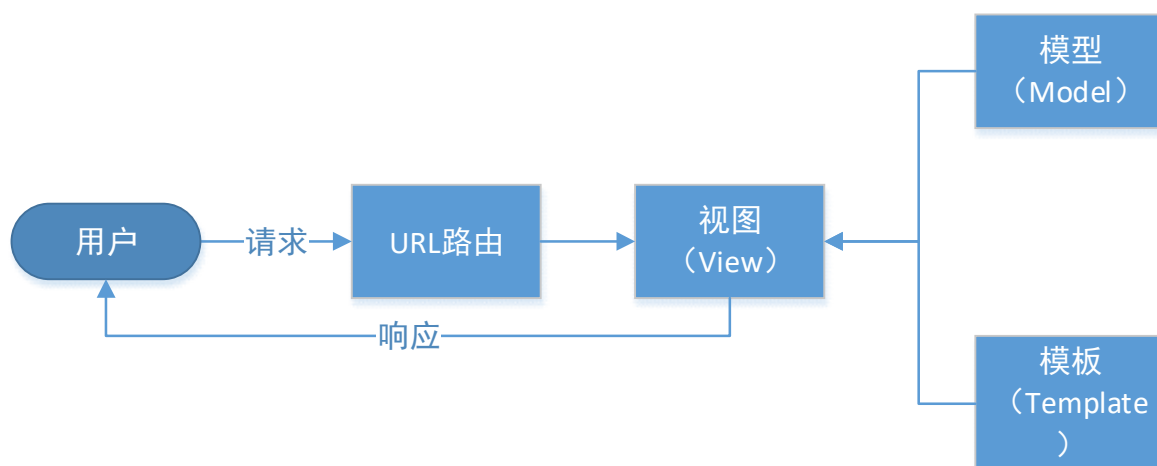


图 2.2 Django 工作模型图

2.2.5 消息队列

本系统使用 RabbitMQ 消息队列中间件来完成，任务发布与端口扫描和服务扫描模块之间的对接，使其达到网站与扫描模块之间的解耦。RabbitMQ 相比于其它的队列中间件，稳定性更好，能持久化存储。使用 RabbitMQ 可以将任务平均分摊到每一个扫描节点上。

第3章 系统分析

3.1 可行性分析

本章节从社会可行性、技术可行性、功能性需求和非功能性需求四个方面做了全面的剖析。

3.1.1 社会可行性

互联网影响着整个社会的发展，互联网已经成为了事故多发地如：2016年的徐玉玉事件^[9]、2017年的Wannacry勒索病毒^[10]等。所以研究网络安全相关的工具与对策及治理办法成为了一个迫在眉睫的任务，近几年很多网络安全相关的企业机构相继崛起，让更多的网络安全人士看到了希望发展的希望。“十三五”规划第六篇指出强化信息安全保障。统筹网络安全和信息化发展，完善国家网络安全保障体系，强化重要信息系统和数据资源保护，提高网络治理能力，保障国家信息安全^[11]。从社会发展现状和国家政策支持来看，此项目对网络安全行业的发展是有一定意义的。

3.1.2 技术可行性

系统开发的整个开发过程使用当前较稳定的 Sublime Text 代码编辑器进行开发，整个系统使用了作者熟悉的 Python3 作为主要开发语言，大大降低了技术风险。用 Django 对网站进行架构，Django 框架有完整的路由系统，还自带 RBAC 权限模块，还有强大的 ORM 功能等众多优点，使网站的结构清晰明了。行业内很多技术人员使用 Django 框架开发，贡献了许多的开发文档，大大降低了本项目的技术风险。使用了稳定性高的 MySQL 作为整个系统的主要存储，采用了 RabbitMQ 消息队列中间件和 MongoDB 解决了分布式扫描的难点。RabbitMQ 相较于其它消息队列优点在于其极强的稳定性。使用了 ElasticSearch 全文搜索引擎作为搜索部分的核心，其优势在于其在行业内搜索领域极高的声誉，并且 ES 还支持分布式存储搜索。整个系统满足了安全稳定，易扩展的要求。

3.1.3 运用可行性

网络空间搜索引擎网站部分采用 B/S 结构，前端采用了 HTML5、CSS3、JavaScript 进行编码设计，用户只需要在浏览器上进行操作，使用了 Bootstrap 框架进行开发大大提高了用户友好度。后端代码使用了 Django 框架进行功能整合，是代码结构简单明了使整个系统有很高的扩展性。整个系统可以运行在 Linux 系统上使整个系统的稳定性更强。

3.2. 功能需求分析

整个系统分为三大模块。具体内容如下：

3.2.1 搜索模块

- 1) 搜索功能说明：普通用户进入首页，输入查询条件由系统对用户输入的信息进行处理，然后返回对应的结果到首页。用户可以输入系统指定的语法对某个字段进行精准的搜索。
- 2) 搜索结果：系统返回搜索结果后，用户可以点击某个 IP 进行单独的结果显示，也可以点击 IP 地址跳转到该 IP 对应端口的首页。
- 3) 单独 IP 结果显示：对某个 IP 地址的信息进行单独页面的展示，包括：ip、端口、服务类型、产品名等。
- 4) 网站在搜索结果页可以展示出搜索引擎中的前十条统计数据，包括服务统计、端口统计。

3.2.2 任务管理模块

- 1) 登录：用户在登录界面输入用户名和密码，传到后台通过系统验证用户是不是管理员，是管理员则执行进入后台管理界面。
- 2) 添加端口扫描任务：管理员登录后可以进入任务管理界面，添加 IP 信息进入队列系统，输入的 IP 地址格式可以为：192.168.1.1 或 192.168.1.1-192.168.1.255 或 192.168.1.1/24。并且在提交成功后显示提交成功。
- 3) 添加服务扫描任务：管理员登录后可以进入任务管理界面，添加服务扫描任务。输入格式：ip:port（192.168.1.1:8080）每行一条
- 4) 删除任务：管理员登录后可以对队列系统的信息进行清空。选择知道的队列名，然后点击提交即可清空队列任务。
- 5) 统计展示：管理员登录后可以查看系统数据库所有统计后的服务名称及数量还有端口号及数量。
- 6) 统计信息更新：管理员登录后可以对服务名及端口号的统计信息进行更新统计。

3.2.3 扫描模块

- 1) 端口扫描：端口扫描脚本从队列中间件读取到一个 IP 地址后，对该 IP 地址下面 1-65535 端口进行扫描，并在扫描后将开放的端口以(192.168.1.1:80)的结构写入队列中间件。
- 2) 服务扫描：服务扫描脚本通过队列中间件读取一条端口(192.168.1.1:80)数据然后通过 python 自带的模块执行 shell 命令调用 nmap。
- 3) 数据转存：管理员在服务器通过命令行的方式可以加 mongoDB 中的 Nmap 扫描数据转存到 MySQL。

3.3 非功能性需求

- 1) 单机端口扫描速度：单个服务器(CPU1 核、内存 2G、带宽 1Mbps)，每小时至少对 10 万个端口进行扫描。
- 2) 单机服务识别速度：单个服务器(CPU1 核、内存 2G、带宽 1Mbps)，每小时至少对 100 个开放的端口进行服务识别。
- 3) 稳定性：系统在某个模块宕机后任然能继续运行。

第 4 章 系统设计

依照第 3 章中的需求分析，对系统进行相对应的设计。主要为系统框架的设计、系统功能模块的设计和数据库的设计。

4.1 系统框架设计

本论文中的网络空间搜索引擎采用 Python 语言进行编写。系统以高内聚低耦合的标准为指导，使整个系统高度模块化。将扫描模块从整个系统单独提取出来，端口扫描和服务扫描都可以作为独立的功能使用。系统数据流图如图 4.1 所示。

扫描部分采用 Masscan 和 Nmap 结合进行扫描。Masscan 做为端口扫描，Nmap 做服务扫描。使用 RabbitMQ 消息队列中间件作为任务添加和扫描之间的数据交换，使扫描模块独立出来可以实现多台机器同时扫描以提高扫描的效率。

网站采用 B/S 架构，包含两大部分：搜索和后台管理。整个网站部分使用 Django 框架做支撑，使用 ES 全文检索数据库进行全文索引。Python 的 elasticsearch 库可以对 ES 进行读写操作。

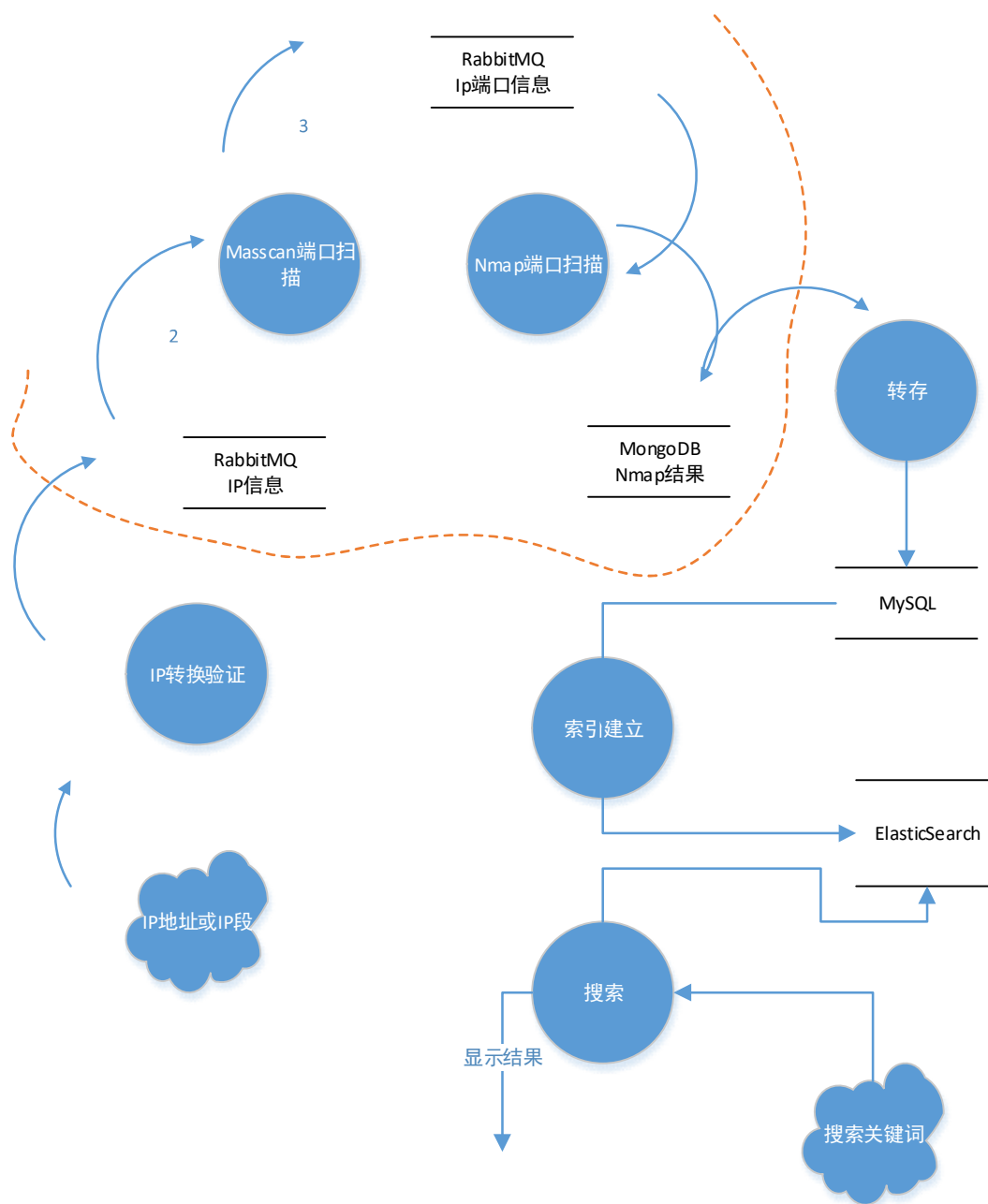


图 4.1 系统整体数据流图

4.2 系统功能模块设计

对于网络空间搜索引擎，整体采用 B/S 的模式与用户进行交互。用户不需要安装专门的软件即可直接使用本系统。本章节主要分两个模块对本系统的功能框架进行阐述。如图 4.2 系统功能模块图。

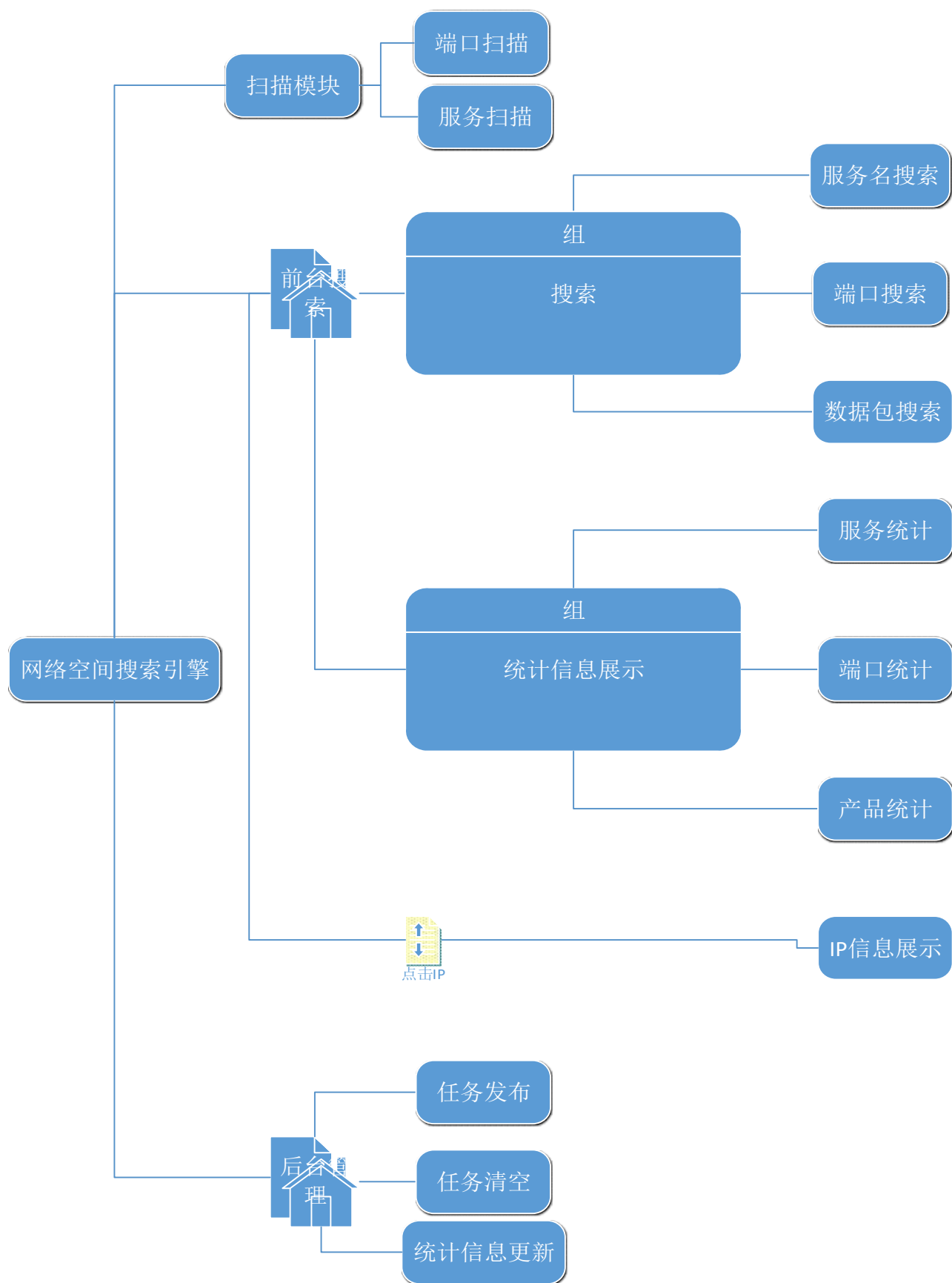


图 4.2 系统功能模块图

4.2.1 搜索模块

搜索模块主要包含：前台搜索和后台的任务发布

前台首页可以展示系统的一些数据统计结果如：服务名数量统计、端口数量统计等
搜索功能可能对服务名，端口，数据包内容等进行搜索

后台管理主要包含添加任务和清空任务队列

4.2.2 扫描模块

Masscan 扫描功能主要对 IP 的所有端口进行扫描，Masscan 扫描脚本启动时先连接 RabbitMQ 中间件，建立持久化连接后 RabbitMQ 会主推送 IP 给 Masscan 脚本然后执行任务。Masscan 将扫描后的结果存入 RabbitMQ 中间件

Nmap 扫描功能：主要对 IP 的端口进行服务的扫描识别。Nmap 脚本启动后与 RabbitMQ 建立持久化的连接，RabbitMQ 会及时推送 IP 及端口给 Nmap 扫描脚本然后开始执行扫描任务。Nmap 脚本对端口进行扫描后将结果存入 MongoDB 服务器。如图 4.3 扫描模块框架图。

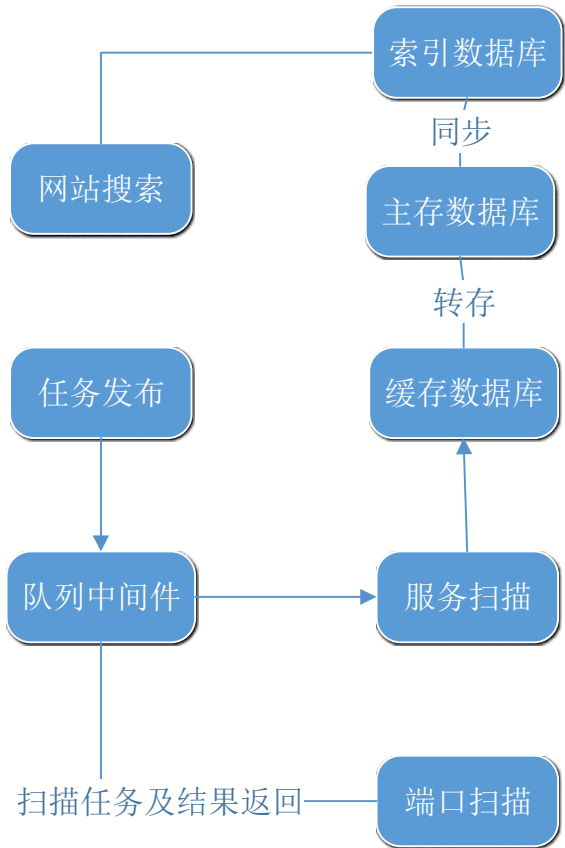


图 4.3 扫描模块框架图

4.3 系统数据库设计

1) nmap 表存储扫描后的结果。如表 4.1 所示

表 4.1 nmap 表结构

字段名	数据类型	说明
id	Int	主键自增
ip_id	Int	对应 iplocation 表 id
port	Int	端口号
protocol	Varchar	协议名
ser_name	Varchar	服务名
ser_product	Varchar	产品名
ostype	Varchar	系统类型
body	Longtext	网页主题内容
indate	Datetime	日期

2) iplocation 表存储 IP 地址的主表。如表 4.2 所示

表 4.2 IP 存储表

字段名	数据类型	说明
id	Int	主键自增
ip	Bigint	IP 地址，以数字类型存储
indate	Datetime	添加时间

3) ser_statistics(服务统计表)，统计服务信息。如表 4.3 所示

表 4.3 服务信息统计表

字段名	数据类型	说明
id	Int	主键编号自增
ser_name	Varchar	服务名
ser_count	Int	统计数量

4) port_statistics(端口统计表), 统计端口信息。如表 4.4 所示

表 4.4 端口信息统计表

字段名	数据类型	说明
id	Int	主键编号自增
port	Int	端口号
p_count	Int	端口数量统计

5) auth_user(用户表), 存储管理员账户的基本信息。如表 4.5 所示

表 4.4 端口信息统计表

字段名	数据类型	说明
id	Int	主键编号自增
password	Varchar	用户密码, 密文为 sha256
last_login	Datetime	最后登录时间
is_superuser	Tinyint	是否为超级用户(1 为超级用户)
username	Varchar	用户名
first_name	Varchar	第一次使用名称
last_name	Varchar	最后使用名称
email	Varchar	用户邮箱
is_staff	Tinyint	是否可以访问 admin 管理界面
is_active	Tinyint	是否已登录
date_joined	Datetime	添加时间

第 5 章 系统详细设计与实现

5.1 项目结构

本项目使用的 Python 版本为 Python 3.5.4。搜索模块使用的开发工具为 Sublime Text3，系统为 Windows10。扫描模块使用的开发工具为 jupyter，系统为 Linux。浏览器以 Chrome 内核为主，前端界面使用 HTML+CSS3+JavaScript、框架使用 Bootstrap 响应式布局框架，网站的后端使用 Django 作为支撑。使用到的数据库有 MySQL，MongoDB、ElasticSearch。使用到的中间件 RabbitMQ。

5.2 管理员模块

此模块主要是管理员登录之后才能操作的功能。

5.2.1 登录功能

登录功能主要是出于安全方面的考虑，防止恶意用户进入本系统随意添加或修改网站破坏本系统，所以管理员必须登录才能使用本系统的后台功能。此功能使用 Django 自带的权限认证机制，只需要调用 auth 模块^[12]，并恰当的使用其方法。用户从前台点击登录后，跳转到登录界面输入用户名和密码点击提交将表单通过 POST 提交到后台，Django 接收到用户名密码后从数据库中读出用户名看密码是否匹配。如果匹配则登录成功自动跳转至后台首页，否则提示用户失败。图 5.1 是登录界面的展示。



图 5.1 登录界面

5.2.3 添加端口任务

管理员通过此功能的界面可以对系统添加任务。管理员在界面输入 IP 或者 IP 段(格式: 139.199.187.177 或 139.199.187.1-139.199.187.254 或 CIDR139.199.187.1/24), 如果用户输入的是 IP 地址系统将直接写进 RabbitMQ 队列中间件的 work 队列, 如果用户输入的是 IP 段(139.199.187.1-139.199.187.254)系统将自动解析生成 IP 列表然后写入 work 队列, 如果用户输入的是 CIDR 格式 IP 段(139.199.187.1/24)系统也将自动解析然后写入 work 队列。系统会自动检查用户输入的数据是否符合要求, 如果不符合则弹框提示用户并且不写入 work 队列。图 5.2 是添加任务界面的展示。

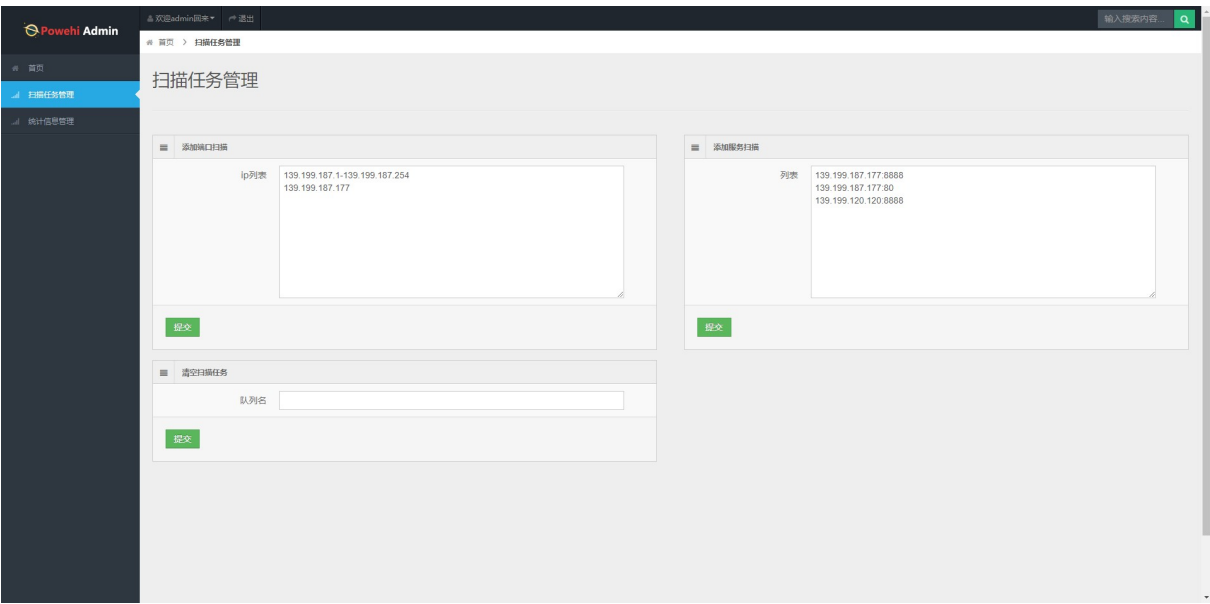


图 5.2 添加任务界面

5.2.3 添加服务扫描任务

管理员通过此功能的界面可以对系统添加服务扫描任务。管理员在界面输入格式: IP:port (139.199.187.177:8080), 如果用户输入的是扫描任务系统将直接写进 RabbitMQ 队列中间件的 ports 队列, 如果用户输入了不符合格式要求的数据, 将提示用户数据格式错误。

5.2.4 清空任务

管理员通过此功能可以对 RabbitMQ 中间件中的队列数据进行清空。用户通过界面输入队列的名称并且点击确定即可清空队列，如果成功则提示用户成功否则提示失败。图 5.3 是清空任务界面的展示。

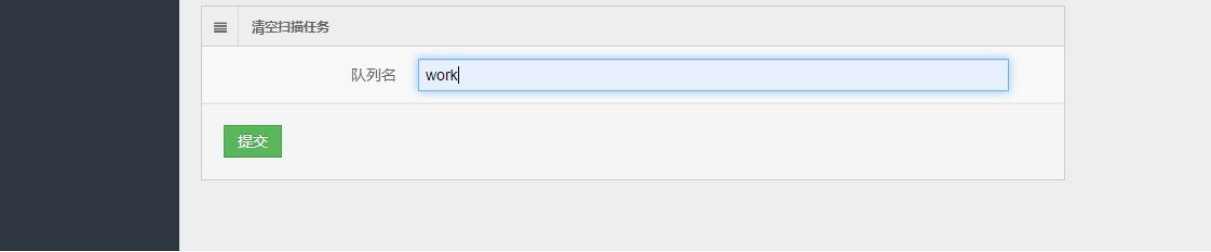
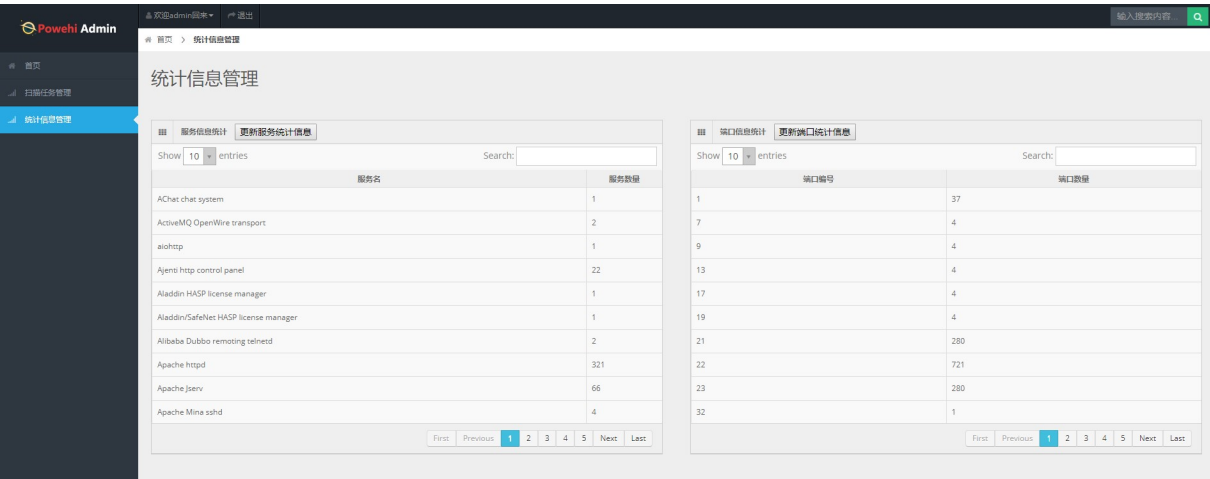


图 5.3 清空任务界面

5.2.5 更新统计数据

管理员通过此功能可以对统计数据更新，包括服务统计，端口统计。系统通过 mysql 的 group by 语句，将结果转存至对应的服务统计表和端口统计表。图 5.4 是查看和更新攻击数据界面的展示。图 5.4 是查看和更新统计数据界面。



服务名	服务数量
AChat chat system	1
ActiveMQ OpenWire transport	2
aliohttp	1
Ajenti http control panel	22
Aladdin HASP license manager	1
Aladdin/SafeNet HASP license manager	1
Alibaba Dubbo remoting telnetd	2
Apache Httpd	321
Apache Jserv	66
Apache Mina sshd	4

端口编号	端口数量
1	37
7	4
9	4
13	4
17	4
19	4
21	280
22	721
23	280
32	1

图 5.4 查看和更新统计数据界面

5.3 搜索模块

5.3.3 首页展示

用户打开网站首页会提示用户怎样进行搜索，列出所有搜索语法。

5.3.1 搜索功能

用户通过此功能可以对需要的关键字或者服务名或端口进行搜索，并查看其结果。每页显示十条信息，系统自动对搜索结果进行分页，点击下一页时进入下一页。具体实现原理，使用 ES 的 scroll 功能，ES 系统会将搜索结果的文档 id 进行缓存，然后返回一个 scroll_id 给用户，用户进行翻页操作是只需要将 scroll_id 带入查询即可返回 10 条新结果给用户。界面如图 5.5 所示。

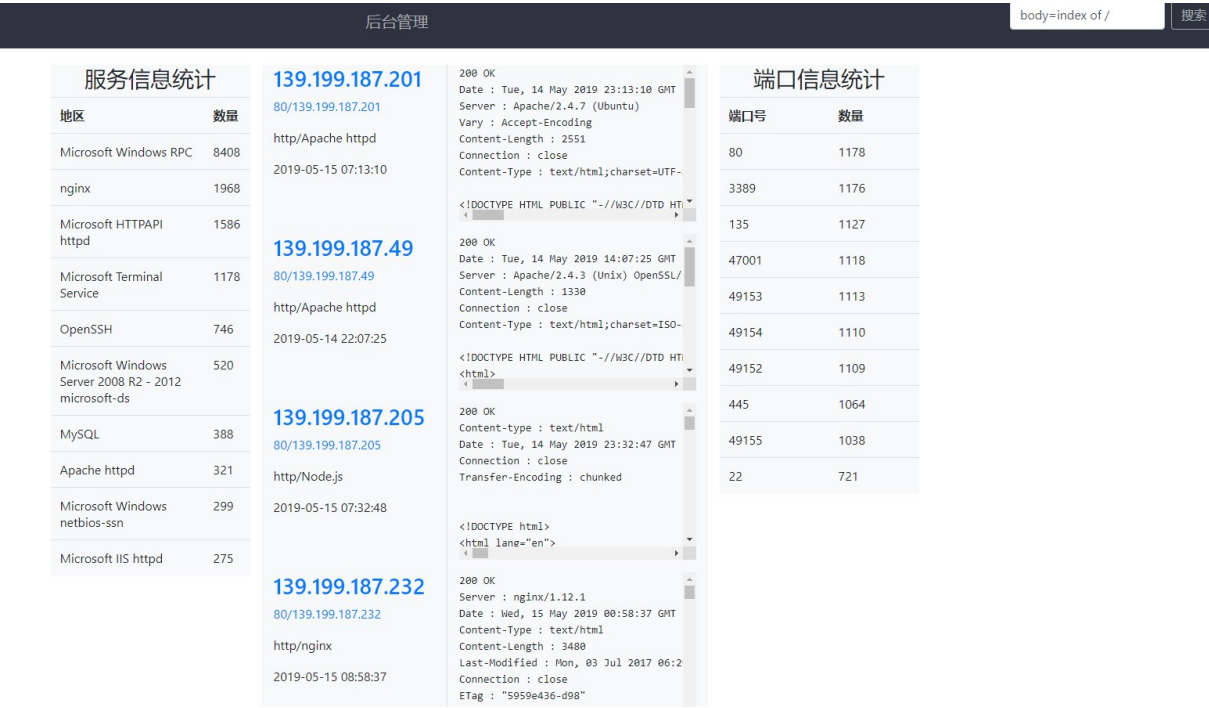


图 5.5 搜索结果界面

5.3.2 单 IP 结果显示

用户通过点击某个结果的 IP 地址可以查看该 IP 地址下面的所有信息，包括：端口、服务、数据包、操作系统等。

系统在单独的页面接收到 IP 地址后，从主数据库拉取该 IP 下面的所有信息。然后通过视图返回到前台进行展示。搜索结果显示如图 5.6 所示。

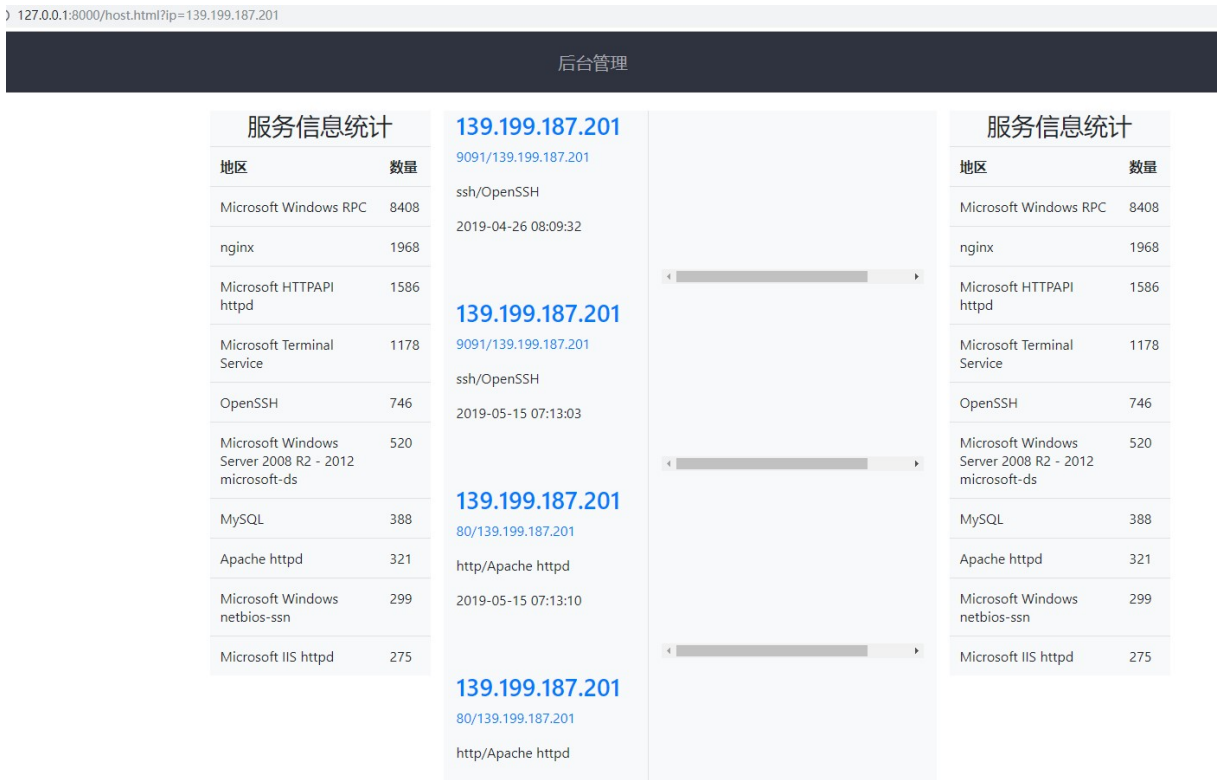


图 5.6 单 IP 搜索结果界面

5.4 扫描模块

扫描模块

5.4.1 端口扫描

通过命令行启动 `rabbitStart.py` 脚本，脚本启动后通过调用 `pika` 模块连接到 `rabbitmq` 中间件的 `work` 队列，并且设置 `basic_qos(prefetch_count=1)` 每次只推送一条数据到脚本，然后设置好回调函数，进入循环后系统如果推送数据则开始执行回调函数。

回调函数的执行是一个扫描周期的开始，回调函数接收到系统推送的 IP 地址后调用 `shell` 命令执行 `masscan` 扫描获取扫描结果后并使用正则表达式对返回的结果进行处理。处理后为：139.199.187.177:27017

139.199.187.177:22

139.199.187.177:8888

如图 5.7 是 `masscan` 扫描结果。

```
[root@VM_0_5_centos ~]# /var/opt/masscan/bin/masscan 139.199.187.177 -p1-65535 --rate=1000
Starting masscan 1.0.6 (http://bit.ly/14GzzcT) at 2019-05-19 03:44:57 GMT
-- forced options: -ss -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 27017/tcp on 139.199.187.177
Discovered open port 22/tcp on 139.199.187.177
Discovered open port 8888/tcp on 139.199.187.177
```

图 5.7 masscan 扫描结果

脚本将结果处理完后连接到 `rabbitmq` 的 `ports` 的队列，以一个 `ip` 对应一个端口 (139.199.187.177:27017) 为最小单位写入 `ports` 队列。将所有扫描结果写入后，一个工作周期结束等待下一周任务。

5.4.2 服务扫描

通过命令行启动 `rabbitNmap.py` 脚本，脚本启动后通过调用 `pika` 模块连接到 `rabbitmq` 中间件的 `ports` 队列，并且设置 `basic_qos(prefetch_count=1)` 每次只推送一条数据到脚本，然后设置好回调函数，进入循环后系统如果推送数据则开始执行回调函数。

回调函数是一个扫描周期的开始，回调函数结果到 `ports` 队列推送过来的数据 (139.199.187.177:27017) 后，调用 `libnmap` 模块执行 `nmap` 命令并且自动对返回结果进行处理。将结果写入 `mongodb` 数据库。

5.4.3 数据同步

管理员不定期执行 `nmapToMysql.py` 脚本将数据从 `mongodb` 转存到 `mysql` 数据库, 同步成功一条数据后, 将该数据从 `Mongodb` 中删除。

5.4.3 索引建立

管理员不定期执行 `indexES.py` 脚本将数据从 `mysql` 读取出来写入 `ES` 数据库, 由 `ES` 数据库自动建立索引。

第 6 章 系统测试

为了确保系统在上线是的稳定性，除了开发过程中进行了单元测试，在编码工作结束后还对项目进行了系统测试。由于本论文篇幅原因，本章节对本系统的主要功能进行的功能测试进行表述。

6.1 测试环境

本系统测试过程中使用两台云服务器和一台物理机进行测试。

1) 云服务器 1:

配置信息：CPU 单核，内存 2G，操作系统 CentOS 7.3 64 位，带宽 1M。

软件：RabbitMQ 3.7.1，Python3.5.0

部署信息：端口扫描 x1，服务扫描 x2。

2) 云服务器 2:

配置信息：CPU 单核，内存 2G，操作系统 CentOS 7.5 64 位，带宽 1M。

软件：MongoDB 4.0.5，Python3.5.0

部署信息：端口扫描 x1，服务扫描 x2。

3) 物理机:

配置信息：CPU I7 八核，内存 8G，操作系统 Windows10 64 位，带宽 4M。

软件：MySQL 5.7.13，ElasticSearch 6.6.1，Python3.5.0

部署信息：网站，数据同步，索引建立。

6.2 搜索模块测试

搜索功能主要从搜索结果的准确性和稳定性进行测试，测试过程如表 6.1 所示。

表 6.1 搜索模块测试用例及结果

模块功能	搜索
测试前提	打开系统首页
测试步骤	输入搜索内容 点击搜索
测试数据	不输入任何数据 输入不存在搜索数据 Index of / 输入 port=8080 输入 port=wefwef
预期结果	作出反应 跳转至搜索结果页面 返回 body 包含 index of /的结果 返回 post 为 8080 的结果 返回首页
实际结果	测试结果与预期结果一致
测试状态	搜索模块正常

6.3 添加端口扫描任务模块测试

添加端口扫描任务主要查看任务是否添加成功，是否对用户做出正确提示，测试过程及结果如表 6.2 所示。

表 6.2 添加端口扫描任务测试用例及结果

模块功能	添加端口扫描任务
测试前提	成功登录后台管理
测试步骤	打开扫描任务管理页面 输入测试数据 提交
测试数据	错误数据 139.199.187.177 139.199.187.1/24
预期结果	提示失败 提示成功 提示成功
实际结果	测试结果与预期结果一致
测试状态	数据成功添加进 RabbitMQ 中间件

6.4 添加服务扫描任务模块测试

添加服务扫描任务主要查看任务是否添加成功，是否对用户做出正确提示，测试过

程及结果如表 6.3 所示。

表 6.3 添加服务扫描任务测试用例及结果

模块功能	添加服务扫描任务
测试前提	成功登录后台管理
测试步骤	打开扫描任务管理页面 输入测试数据 提交
测试数据	错误数据 139.199.187.177:8080 139.199.187.1:80
预期结果	提示失败 提示成功 提示成功
实际结果	测试结果与预期结果一致
测试状态	数据成功添加进 RabbitMQ 中间件

6.4 统计数据更新

表 6.4 统计信息测试用例及结果

模块功能	统计信息管理
测试前提	管理员登录打开统计信息管理页面
测试步骤	点击更新服务统计信息 翻页统计信息 点击端口信息统计
测试数据	刷新界面
预期结果	统计成功刷新页面 翻页成功 统计成功刷新页面
实际结果	测试结果与预期结果一致
测试状态	正常

第 7 章 结论

本论文设计并实现了一个完整的网络空间搜索引擎，并且对其原理进行了详细的说明。分析了市面上已有的类似的系统，并做了详细的对比，指出了他们的异同。在此基础上完善了自身的设计，并且实现了一个相对比较有好的可视化界面。整个系统采用分布式扫描以提升系统扫描效率。主要完成了以下任务：

- 1) 通过调用 `masscan` 提升了端口扫描的效率。
- 2) 通过调用 `nmap` 完善了服务识别。
- 3) 使用 `elasticsearch` 提升了系统全文检索能力。

虽然系统实现了最初设计的要求，但是在开发中还是发现了一些最初设计的不足。不支持目标软件或系统的漏洞扫描。

通过本次毕业设计锻炼了我的综合能力，让我对软件开发和网络安全行业有了更深刻的认识。提高了我的信息搜集能力，能根据自身的需要查找到相对应的资料。将搜集的信息用于项目的研究解决实际问题。提升了自身的论文撰写能力，从文章的结构分析和对大学的课程都有了更清晰的认识，将问题给不懂的人解释清楚。为自己即将踏入社会做好铺垫。

参考文献

- [1]习近平. 自主创新推进网络强国建设[EB/OL]. http://www.cac.gov.cn/2018-04/21/c_1122719824.htm.
- [2]水熊科技. 网络空间搜索引擎全方位评测[EB/OL].
<https://www.freebuf.com/sectool/129211.html>.
- [3]savon-noir. libnmap[EB/OL]. <https://github.com/savon-noir/python-libnmap>.
- [4]robertdavidgraham. masscan[EB/OL]. <https://github.com/robertdavidgraham/masscan>.
- [5]郝科委,余翔湛,赵洋.大规模网络高速扫描系统的设计与实现[J].智能计算机与应用,2018,卷缺失(5):112-117.
- [6]Gordon Lyon. nmap. <https://nmap.org/>.
- [7]elastic. Elastic. <https://www.elastic.co/>.
- [8]java_龙. elasticsearch 教程--中文分词器作用和使用.
<https://blog.csdn.net/an88411980/article/details/83747230>.
- [9]徐玉玉事件. <https://baike.baidu.com/item/徐玉玉/19919942?fr=aladdin>.
- [10]WannaCry 病毒. <https://baike.baidu.com/item/WannaCry/20797421?fr=aladdin>.
- [11]周玮. 十三五规划纲要: 第六篇拓展网络经济空间.
<http://news.eastday.com/c/lh2016/u1ai9260896.html>.
- [12]breezey. Django 中@login_required 用法简介.
<https://www.cnblogs.com/breezey/p/6715641.html>.
- [13]马程.网络空间搜索引擎的原理研究及安全应用[J].网络空间安全,2016,卷缺失(5):6-10.
- [14]于博菲.基于物联网技术的搜索引擎与设备安全[J].金属世界,2015,卷缺失(1):47-50.
- [15]李强,贾煜璇,宋金珂,等.网络空间物联网信息搜索[J].信息安全学报,2018,卷缺失(5):38-53.
- [16]齐权,贺劼,鲁悦.网络空间资产普查与风险感知系统[J].信息技术与标准化,2018,卷缺失(9):53-56, 69.
- [17]袁新昌.网络空间资源探测系统的设计与实现[D].[出版地不详]:北京邮电大学,2017.
- [18]王智民.网络资源探测及可视化呈现系统的设计与实现[D].[出版地不详]:电子科技大学,2018.
- [19]王宸东,郭渊博,黄伟. 非入侵式网络安全扫描技术研究[J].信息安全与通信保密,2016,卷缺失(9): 67-72,76.
- [20]陈卓.基于无状态连接的工控系统扫描平台的设计与实现[D].[出版地不详]:北京邮电大学,2018.
- [21]Shodan[EB/OL]. <https://www.shodan.io>.

致谢

大学时光匆匆而过，想当初我们还是刚步入校园求学的学子，如今，已成为即将步入工作岗位的社会人士，要去承担起属于自己的一份社会责任。在这几年过程中，我不仅学到了丰富的专业知识，更学会了如何去承担责任。在此，我要由衷地感谢培养我的母校，给予我指导的老师，陪伴我学习与生活的同学。

我要感谢我的母校——湖南理工学院，是她给予了我汲取知识的殿堂，给了我锻炼自己能力的舞台。我要感谢教过我，在我人生路上遇到困惑时为我指点迷津的老师，在此，我要特别地感谢我的指导老师——甘靖老师，从毕设论文的选题，到课题研究，再到分析与详细设计，一直是甘靖老师在指导我，为我答疑解惑。我要感谢陪伴我学习与生活的同学，是他们让我体验到了多姿多彩的大学生活，让我收获了同学们之间最真挚的友谊。

大学生活即将结束，标志着工作生涯即将开启，我依然要抱着积极向上的生活态度和刻苦钻研的进取精神去面对生活，面对工作。

最后，祝愿母校顺利实现“创大申博”，实现更好的发展，祝愿老师们身体健康、生活幸福，祝愿同学们工作顺利、前程似锦。