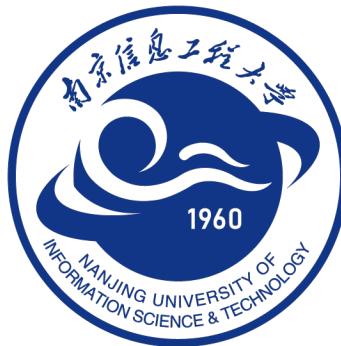


单位代码: 10300

南京信息工程大学

本科毕业设计



题 目 基于渗透测试的红队武器库系统

学生姓名: 于家瀛

学 号: 202083290297

专 业: 信息安全

学 院: 计算机学院、网络空间安全学院

指导教师: 陈先意

二〇二四年五月十日

目 录

中文摘要	III
英文摘要	IV
1 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 系统的主要工作和结构	2
2 相关技术简介	3
2.1 Flask 全栈开发技术简介	3
2.2 Scapy 的终端命令简介	3
2.3 XSS 攻击原理简介	4
2.4 基于 Tomcat 的网站搭建	5
3 概要设计	8
3.1 需求分析	8
3.2 总体架构	9
3.3 功能性	9
3.3.1 系统主页功能性	9
3.3.2 加密模块功能性	9
3.3.3 爆破模块功能性	10
3.3.4 扫描模块功能性	10
3.3.5 泛洪攻击模块功能性	11
3.3.6 DOS 攻击检测和防御	11
3.3.7 爬虫模块功能性	11
3.4 非功能性	12
3.5 数据库设计	12
4 详细设计	14
4.1 系统主页设计详述	14
4.2 加密模块设计详述	15

4.2.1 对称加密功能的实现	15
4.2.2 非对称加密功能的实现	16
4.2.3 勒索病毒模拟	17
4.3 爆破模块设计详述	18
4.3.1 爆破 WiFi 密码	18
4.3.2 爆破 RAR 压缩文件	19
4.3.3 爆破网站账号密码	19
4.3.4 爆破 SSH 密码	20
4.4 扫描模块设计详述	21
4.4.1 基于 Socket 的端口扫描	21
4.4.2 基于 Scapy 的端口扫描	21
4.4.3 基于 Ping 命令的 IP 扫描	22
4.4.4 基于 Scapy 的 IP 扫描	22
4.4.5 的子域名扫描	23
4.4.6 Web 站点信息搜集	24
4.5 泛洪模块设计详述	24
4.5.1 TCP 三次握手泛洪	24
4.5.2 Scapy 半连接泛洪	25
4.5.3 ICMP 泛洪	26
4.5.4 MAC 地址泛洪	26
4.5.5 ARP 泛洪攻击和欺骗	27
4.6 DOS 检测和防御模块设计详述	28
4.7 爬虫模块设计详述	29
4.7.1 图片爬虫	29
4.7.2 文字爬虫	30
4.7.3 超链接爬虫	32
5 总结	34
参考文献	35
致 谢	37

基于渗透测试的红队武器库系统

于家瀛

南京信息工程大学计算机学院、网络空间安全学院，江苏 南京 210044

摘要：探究国内外 Python 脚本开发设计多为独立的单功能系统，因此本系统旨在形成一个集成多种渗透测试工具的综合性应用系统。系统采用总体采用 B/S 架构，使用 Python 的 Flask 框架设计而成，主要涵盖了加密解密、密码爆破、端口扫描、IP 扫描、泛洪攻击、DOS 检测防御和爬虫等功能模块。各个模块内部之间功能相似但又有区别，基于相同目的而进行的不同操作。前端页面将各个模块放到菜单栏，便于用户选择适用工具。

关键词：Python 脚本开发；渗透测试工具；Flask 框架；网页爬虫

Red Team Weapon Library System Based on Penetration Testing

Yu Jiaying

School of Computer Science、School of Cyber Science and Engineering, NUIST, Nanjing 210044, China

Abstract: Exploring the development and design of Python scripts both domestically and internationally, which are mostly independent single function systems, this system aims to form a comprehensive application system that integrates multiple penetration testing tools. The system adopts an overall B/S architecture and is designed using the Flask framework of Python. It mainly includes functional modules such as encryption and decryption, password blasting, port scanning, IP scanning, flooding attacks, DOS detection and defense, and crawling. The functions within each module are similar but differ, and different operations are carried out for the same purpose. The front-end page places various modules in the menu bar, making it easier for users to choose suitable tools.

Key Words: Python script development; Penetration testing tools; Flask framework; Web crawler

1 绪论

1.1 研究背景及意义

渗透测试是评估计算机网络安全的重要方法，主要通过模拟黑客对服务器进行各种安全攻击。因此个人开发 Python 攻击脚本的意义对渗透测试工作中的理解和安全防护的思路有很大帮助意义。首先，目前行业内基于 Python 开发的渗透测试系统还尚未成熟，然而实际生产工作中需要一个系统全面的测试工具集。

其次，对于学习和研究网络攻防也很有意义。基于 Python 开发安全攻击脚本有助于增强自身 Python 编程水平，还可以安全攻击的底层原理细节有所了解。编写攻击脚本是一个很好的学习和研究过程。在这个过程中，个人可以深入了解网络协议、操作系统、应用程序等底层细节，从而加深对计算机科学和网络安全的理解。此外，个人还可以通过研究现有的攻击技术和防御策略，来开发更有效的防护手段。

然后，个人可以自主测试系统安全性。Python 攻击脚本可以用于测试系统的安全性，通过模拟真实的攻击场景，个人可以发现系统中存在的漏洞和弱点，并据此进行修复和改进。这种测试方法有助于确保系统的稳定性和安全性，减少潜在的安全风险。

再次，有利于研究和应对新型攻击。随着网络安全形势的不断变化，新型攻击手段层出不穷。个人开发 Python 攻击脚本可以有助于研究这些新型攻击的原理和方式，从而提出有效的防御策略^[1]。这对于保护个人和组织的网络安全具有重要意义。

最后，个人也需要懂得开发 Python 攻击脚本必须遵守法律法规和道德规范。任何未经授权的网络攻击都是违法的，并且会对他人造成严重的损失。因此，在开发和使用 Python 攻击脚本时，必须确保自己的行为符合法律和道德要求。同时，个人也应该积极参与网络安全社区的建设和发展，为提高整个社会的网络安全水平做出贡献。

1.2 国内外研究现状

首先，从国内来看，Python 凭借其优秀的易用性、可读性和具有强大丰富的第三方库的支持已成为大数据、人工智能、数据分析等领域的重要工具。随着云计算、大数据、人工智能等技术的快速发展，Python 的应用场景还也在不断扩大。但是目前 IT 领域对于基于 Python 的安全脚本开发还比较匮乏，大多停留在基础扫描器的复现和魔改。对于自动化更高的渗透测试脚本开发研究则更加难以满足实际工作中的自动化程度要要求^[2]。

从国外来看，Python 同样受到了广泛的关注和应用。Python 在国外的发展更加成熟和深入，许多知名的科技公司、开源社区和大学都在使用 Python 进行研究和开发。Python 在数据科学、机器学习、自然语言处理等领域的应用尤为突出，许多著名的开源项目和框架都是基于 Python 开发的。国外的安全脚本开发也相对比较成熟，像 Nmap 和 Burpsuite 的一些安全类软件的市场使用率非常高，但是对于基于 Python 的安全脚本开发也多为实验性的小功能，还没有形成功能集成化的系统^[3]。

总之，Python 安全脚本开发在国内外都呈现出浅尝辄止的研究现状，大多停留在实验教学阶段。对于各类安全工具的防火墙过滤、性能提升都较少能达到实际生产生活的需要。

1.3 系统的主要工作和结构

本系统的功能主要是协助渗透测试工程师对一些网站或者软件进行渗透测试，系统集成了诸多常用测试攻击功能。本系统采用 B/S 架构，实现了图形化的前端页面，可以非常便捷的为渗透测试者提供服务。基于 Flask 的设计模式可以使前后端交互更加立体，便于根据用户需求及时调整^[4]。

系统的主要工作内容是集成了多种渗透测试工具，以模块进行划分不同方面。大致可以划分为算法加密模块、爆破模块、扫描模块、泛洪攻击模块、DOS 检测防御模块和爬虫模块。每个模块下有提供了多种功能实现的方式，或者不同场景下的功能需求。比如扫描模块提供端口扫描和 IP 扫描，而其中 IP 扫描又分为基于 Ping 命令的 IP 扫描和基于 Scapy 的 IP 扫描。使得用户在渗透测试工作中可以灵活切换模式，适用不同测试环境。防御性的 Python 脚本开发依然是系统的主要功能。对于网络安全领域，攻防是相辅相成、相互依赖的，良好的防御系统会以强大的攻击系统为假想敌，进行不断优化^[5]。因此本系统对于 DOS 检测和防御的功能实现上也是考虑比较全面彻底，也是系统设计的主要工作。系统的泛洪攻击模块刚好可以对 DOS 防御检测模块进行测试，验证系统的可靠性。最后就是对系统进行拓展性的开发——加入了爬虫模块，该模块同样考虑了各种网页资源的爬虫。因此系统对图片爬虫、文字爬虫和超链接网页爬虫做了系统化集成，使得有爬虫需求的用户方便选取模式。

总之，系统是采用基于 Python 的 Flask 框架开发 Web 应用系统，集成了渗透测试所需要的多种安全工具。主要工作就是给用户提供恰当的测试工具对目标网站或是主机进行渗透测试。同时系统兼顾防御，该系统还可以对用户的主机进行 DOS 检测和防御。

2 相关技术简介

2.1 Flask 全栈开发技术简介

本系统是前后端不分离的基于 Python Flask 的全栈渗透测试系统，因此前端的可视化页面直接连接到后台 Python 逻辑代码。下面对于 Flask 全栈开发技术，进行基础性介绍，也更有助于理解系统的交互原理。

Flask 是一款简洁且轻量级的 Python Web 框架，常被称为微型框架。它提供了基础而稳固的核心功能，而额外的功能则完全依赖于各种扩展（extensions）来实现。这种设计使得 Flask 具有高度的灵活性和可扩展性，开发者可以根据具体项目的需求进行个性化定制。然而，这也意味着开发者需要熟悉并掌握这些扩展库的使用方法，以便能够充分利用 Flask 构建出功能强大的 Web 应用^[6]。

首先要导入 Flask 库，然后实例化一个 app 对象 `app = Flask(__name__)`，在每个函数前要使用路由装饰器 `route` 来装饰，这样在网站的 url 访问 route 装饰的地址时，便调用被该装饰装饰的函数。在主程序中执行 `app.run()` 方法时，如果使用浏览器访问 url 的根目录时，便调用 `index()` 方法，该方法则直接返回一个 HTML 页面，

在功能模块中，以 AES 加密模块为例，为了使用户体验更好，最好是将加密结果输出在当前页面上，而不是跳转到新的页面，这样不方便查看原明文加密为何密文，也不方便查看何密文解密为了输出的明文。因此，在 HTML 文件下方嵌入待穿参数的段落即可。代码如下：

```
<h3>Results</h3>
<p>Encryption Result: {{ encrypt_result }}</p>
<p>Decryption Result: {{ decrypt_result }}</p>
```

如此在进行 AES 加密解密操作过程中，便可以直观的感受到加密结果和解密结果与加密前或解密前的对照关系。

2.2 Scapy 的终端命令简介

在扫描模块里，本系统大量使用了 Scapy 的构造数据包方式来进行扫描。因此有必要在这里介绍一下 Scapy 的一些基础性的终端命令，便于理解后文提到的基于 Scapy 扫描类攻击方式。Scapy 是一个强大的 Python 库，用于网络包创建、发送、嗅探、解析和伪造。虽然 Scapy 通常通过 Python 脚本使用，但你也可以在 Python 交互式环境中

使用它，比如通过 Python 或 Jupyter Notebook。不过，直接通过终端命令来使用 Scapy 是不常见的，因为 Scapy 主要是一个 Python 库^[7]。

扫描模块的诸多代码都是在调用系统命令行去执行 Scapy 终端命令，首先是 show_interfaces()命令可以查看本机的网络接口，如若要进行嗅探操作，则需要使用命令 sniff(count=10)，coun 表示接收的嗅探数据包数量。因此，要对虚拟网卡进行数据嗅探，可以按照如下方法构造接收数据包：pkg=sniff(count=10, iface=VMware).使用命令 pkg[0]。show()便可以查看嗅探的结果。还可以对嗅探结果进行过滤，比如只查看 icmp 的数据包 pkg=sniff(count=8, filter=icmp, iface=VMware).

Scapy 的 send 命令是可以将构造的数据包发送出去的，注意数据包的构造顺序和规则，从左到右依次是 OSI 参考模型的底层到顶层，用圆括号包裹数据的详细内容，比如源端口：send(IP(dst=192.168.19.130)/ICMP()/HHHHHHHH, inter=1, count=5)。

显然只是发送数据是不够满足系统测试要求的，因为程序还需要对目标的响应结果进行判断，从而采取接下来的渗透操作。因此还需要使用 Scapy 的发送并接收流量包的函数命令 sr1()。该命令和 send()函数命令类似，只是需要在函数前放置一个变量接收。例如：pkg=sr1(IP(dst=192.168.19.130)/ICMP()/HHHHHHHH, inter=1, count=5)。最后使用 show()函数可以查看流量包的内容，比如查看[Raw]字段使用命令 pkg[Raw].load.

2.3 XSS 攻击原理简介

XSS 的英文全称是 Cross-Site Scripting，翻译为中文叫跨站脚本攻击，它的攻击原理主要是利用 Web 页面存在的安全漏洞，在网页中注入恶意脚本代码，构成新的闭合结构从而破坏原有的 DOM 结构。当网页用户浏览该网页时，攻击者构造的恶意脚本会被自动执行，从而达到攻击网站的目的。攻击者往往会采用有注入恶意脚本、使用户浏览受感染的网页、网页执行恶意操作的 XSS 攻击方式在目标网站上注入恶意的 JavaScript 脚本代码，抑或是其他类型的脚本代码，如 VBScript、ActiveX 等。

这些恶意脚本可以以各种形式存在，如嵌入在网页中的<script>标签内，或者作为 URL 参数的一部分^[8]。当其他网站用户访问包含恶意脚本的网页时，这些脚本会在用户的浏览器上执行，影响正常用户对于网站的使用。甚至 XSS 攻击还可以攻击管理员后台，窃取后台管理员的 cookie 信息。比如对如下靶场网页的 XSS 攻击，就是在输入框里输入闭合标签的内容，同时嵌入一个 script 脚本标签起到弹窗的效果。



图 2.6 XSS 闭合标签嵌入 script 脚本

由于这些脚本直接在用户的本地浏览器环境中执行，它们具有潜在的威胁，能够访问、篡改用户的页面内容，并执行多种恶意行为。这些恶意行为包括但不限于窃取用户的 Cookie 数据、操纵用户会话、重定向至恶意站点、插入恶意元素以及传播 XSS 蠕虫，这些都可能引发用户隐私泄露、账户安全威胁及财产损失等严重后果^[9]。

在 XSS 攻击的分类上，根据攻击手法的不同，可分为反射型（非持久型）、存储型（持久型）和 DOM 型。反射型 XSS 依赖于在 URL 中插入恶意脚本片段来执行攻击；存储型 XSS 则是将恶意脚本存储于服务器端，等待用户访问受污染的页面时触发；而 DOM 型 XSS 则是通过直接修改页面上的 DOM 结构来发起攻击。

为了有效防御 XSS 攻击，网站开发者需实施一系列安全措施，比如对用户输入进行详尽的验证和过滤，采用安全的编程实践与框架，并确保设置正确的 HTTP 响应头。同时，用户也应增强自身的安全意识，避免点击不明来源的链接或下载未知文件，从而为自己的网络安全筑起防线。

而本系统对可能存在 XSS 的漏洞的网页进行自动化扫描，也正是基于那些常见的恶意 payload 进行校验，对容易出现 XSS 漏洞的 DOM 结构进行渗透测试。当然这些可能的恶意 payload 被大量保存于 1 字典当中，Python 使用文件读写的方式对每条 payload 进行逐一测试。因为恶意 payload 的字典文件并不大，因此不必担心时间效率问题，扫描基本可以在数秒内完成，也就没必要增加多线程复杂代码也消耗资源。

2.4 基于 Tomcat 的网站搭建

在爆破模块的网站账号密码爆破功能里，靶场环境便是自主搭建的基于 Tomcat 和

MySQL 的网站。接下来将对网站搭建涉及到在 Linux 上对 Tomcat 安装、MySQL 安装、JDK1.8 安装的技术简介。

首先是 JDK1.8 的安装，安装步骤如下：

先将已经在 jdk 官方网站上下载好的 jdk-linux-x64.tar.gz 文件放到 /opt 目录下解压，使用 Linux 命令为 tar -zxvf jdk-8u11-linux-x64.tar.gz 进行解压操作。

然后是为 JDK 添加环境变量 JAVA_HOME，首先切换到 home 目录下，使用命令 ls -a 查看隐藏文件。接下里编辑 .bash_profile 文件。添加条目 export JAVA_HOME=/opt/jdk1.8.0_11 并且修改 PATH=\$PATH: \$HOME/bin: \$JAVA_HOME/bin. 第二行修改的内容实际上是以英文冒号为分隔符进行划分的环境变量参数。

```
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    . ~/.bashrc  
fi  
  
# User specific environment and startup programs  
  
export JAVA_HOME=/opt/jdk1.8.0_11  
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin  
  
export PATH
```

图 2.1 环境变量配置

接下里使用 source 命令启动一下 .bash_profile，Linux 命令为：source .bash_profile 继续使用 .env 命令查看当前配置的环境变量；

最后使用命令 java -version 查看 java 版本，如果有回显出 java 版本则配置成功。

```
[root@centos-7 ~]# java -version  
java version "1.8.0_11"  
Java(TM) SE Runtime Environment (build 1.8.0_11-b12)  
Java HotSpot(TM) 64-Bit Server VM (build 25.11-b03, mixed mode)  
[root@centos-7 ~]#
```

图 2.2 查看安装的 java 版本

其次是 MySQL5.6 版本的安装，之所以选择该版本是为了和上述安装的 Tomcat 版本兼容。在 CentOS 上安装完 MySQL 之后，便可以通过 Navicat 连接到 MySQL，然后创建数据库 woniusales，注意编码格式是 utf8mb64 和 utf8mb4_general_ci。最后给 woniusales 数据库导入 SQL 语句，选择文件 woniusales-20180508-V14.sql 导入即可。 MySQL 的安装步骤如下：

首先查看系统是否安装了 mysql，为空说明没有安装。命令为： rpm -qa | grep mysql
然后查看 mariadb 版本，并将其卸载。

查看命令： rpm -qa|grep -i mariadb

卸载命令： rpm -qa|grep mariadb|xargs rpm -e --nodeps

需要下载 MySQL 安装包文件。

wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm

继续安装 mysql-community-release-el7-5.noarch.rpm 包

rpm -ivh mysql-community-release-el7-5.noarch.rpm

等到安装完成之后，会在 /etc/yum.repos.d/ 目录下新增 mysql-community.repo 、 mysql-community-source.repo 两个 yum 源文件。

需要查看可用的 mysql 安装文件，使用命令为： yum repolist all | grep mysql

然后安装 mysql，使用命令为： yum install mysql-server

进一步检查 mysql 是否安装成功，使用命令为： rpm -qa | grep mysql

```
[root@centos-7 classes]# rpm -qa | grep mysql
mysql-community-release-el7-5.noarch
mysql-community-common-5.6.51-2.el7.x86_64
mysql-community-client-5.6.51-2.el7.x86_64
mysql-community-libs-5.6.51-2.el7.x86_64
mysql-community-server-5.6.51-2.el7.x86_64
[root@centos-7 classes]#
```

图 2.3 查看 mysql 是否安装成功

接下来启动 mysql 服务， systemctl start mysqld.service #启动 mysql

设置密码。mysql5.6 安装完成后，它的 root 用户的密码默认是空的，需要及时用 mysql 的 root 用户登录（第一次直接回车，不用输入密码），并修改密码。第一步登录 mysql 数据库，然后使用 mysql 数据库。在该数据库下执行以下 SQL 语句
update user set password=PASSWORD(这里输入 root 用户密码) where User=root;
flush privileges;

然后可以设置远程主机登录。 GRANT ALL PRIVILEGES ON *.* TO your username@% IDENTIFIED BY your password

最后可以执行以下命令，为 root 用户添加远程登录的能力。 GRANT ALL PRIVILEGES ON *.* TO root@% IDENTIFIED BY 123456;

接下来是 Tomcat 的安装，安装步骤如下：

首先讲官网下载好的 apache-tomcat-8.0.53.tar.gz 放到/opt 目录下解压，解压命令为：

tar -zxvf apache-tomcat-8.0.53.tar.gz

然后在 bin 目录下启动 Tomcat，启动命令为： ./startup.sh

接下来在 webapps 目录下放入蜗牛供销社网站 war 包： woniusales.war

下面进入 webapps/woniusales/WEB-INF/classes 里修改 db.properties 文件，使得 Tomcat 链接到数据库，修改内容如下图所示：

```
[root@centos-7 classes]# cat db.properties
db_url=jdbc:mysql://localhost:3306/woniusales?useUnicode=true&characterEncoding=utf8
db_username=root
db_password=123456
[root@centos-7 classes]#
```

图 2.4 修改的 db.properties 文件内容

最后在浏览器访问 192.168.19.130: 8080/woniusales 即可进入。展示爆破靶场 woniusales 登陆页面如下：



图 2.5 蜗牛供销存访问页面

3 概要设计

3.1 需求分析

当前 Python 的脚本开发领域分支庞大，各个领域的工具都革新好几个版本，比如扫描类工具 Nmap、爆破类工具 Hydra 等。这些成熟的脚本固然功能强大，可是操作起来非常麻烦，各类指令参数指定对新用户非常不友好。而且这些工具也不是基于优秀的脚本语言 Python 编写的，因此注定其无论是效率上还是用户适用面上限没有使用 Python 开发的脚本高。再加之安全工作者在实际渗透测试过程中，还需要来回切换好几个安全工具进行测试工作，极大的影响工作效率甚至扰乱测试思路。所以开发一个系统化集成各类安全工具的便捷系统对渗透测试工作者是一种很大的需求。

比如在进行渗透测试的工作中想要对一个网站进行漏洞验证，一般工作者会对 OWASP 逐一进行漏洞测试，时间和精力损失会很大。而本系统集成了多种漏洞测试，会为测试工作者排除一些漏洞可能。系统可以进行 XSS 扫描对目标网站进行 XSS 漏洞测试，还可以使用爆破模块对账号密码进行爆破，验证是否有弱口令漏洞，还可以使用泛洪攻击模块对服务器进行压力测试等等。

同时系统提供的可视化图形界面也是用户的一大需求，无需再查阅各种指令说明，

只需要在相应模块输入几条参数，便会对目标进行测试。比如用户使用系统的账号密码爆破功能模块的前端页面只需要输入目标 url 地址，系统便会使用多线程分任务的形式高并发的进行爆破攻击。

3.2 总体架构

系统的总体架构是基于 Python Flask 的 B/S 架构，集成了加密模块、爆破模块、扫描模块、泛洪攻击模块、DOS 检测和防御模块、爬虫模块这 6 大模块。前端交互页面由系统主页和各个模块的分页组成。对于加密模块下设了 Base64 编码、MD5 摘要算法、DES 加密、AES 加密和 RSA 加密。在浏览器的输入框内输入要加密的明文，并且输入符合规范的密钥进行加密处理，同样的也可以进行解密操作。

爆破模块则由一些常用的生活爆破工具和网站账号密码爆破组成。常用的生活爆破工具有 RAR 压缩文件爆破、WiFi 密码爆破、SSH 爆破，网站账号密码爆破的靶场环境是基于 Tomcat 的蜗牛供销存系统。扫描模块由端口扫描、IP 扫描、后台扫描、XSS 扫描和站点信息查询组成，其中端口扫描和 IP 扫描有基于 Scapy 的半连接扫描。泛洪攻击模块由 socket 三次握手泛洪、scapy 半连接泛洪、TCP Land 泛洪、ICMP 泛洪、MAC 地址泛洪、ARP 攻击和欺骗构成。最后，便是对 DOS 攻击的检测和防御。

本系统的泛洪攻击模块和 DOS 攻击检测防御模块可以搭配完成一组实验。那就是用系统集成的基于 socket 全连接泛洪和基于 Scapy 的半连接泛洪对靶场服务器进行 DOS 攻击，然后在服务端环境运行 DOS 检测和防御系统对发起的泛洪攻击进行检测和屏蔽。

最后作为系统拓展性模块的爬虫模块，主要集成比较热门的爬虫方式，即：图片爬虫、文字爬虫、超链接网页爬虫^[10]。互联网上存在大量的数据，如何甄别出爬虫系统的目标资源是主要思考方向，而将爬取的资源如何规整的保存下来显然需要用到 Python 的文件读写。

3.3 功能性

3.3.1 系统主页功能性

本系统提供了可视化的操作界面，系统主页标题栏部分包含了各个功能模块的导航，只需要点击相应模块便可以进入该功能模块。在主页的 body 部分提供了关于系统的一些说明和使用方法，方便用户快速上手。

3.3.2 加密模块功能性

加密模块主要由 Base64 编码、MD5 摘要算法、DES 加密、AES 加密、RSA 加密组

成。都是直接调用内置库，然后封装到系统的方法中。通过前端页面传入参数，调用后端逻辑代码处理，输出结果在页面上。因此用户只需要选择需要的加密或者编码类型，输入文本内容便可以获取加密结果。

勒索病毒原理上是对目标主机上的文件进行加密，因此在本模块也集成了模拟勒索病毒的功能。由于是本机测试化环境，因此在页面输入要加密文件的路径，随后开始模拟便可以看到路径文件夹内的文件被加密成功^[11]。现实生活中，勒索者获得酬金会将目标主机的加密文件解开，本系统也对已经加密的文件提供了逆向解密功能。

3.3.3 爆破模块功能性

爆破模块主要由一个字典生成器、生活上的常用爆破工具和网站账号密码爆破组成，生活爆破工具如： WiFi 爆破、 RAR 压缩文件爆破、 SSH 爆破；网站账号密码爆破是基于 Tomcat 的蜗牛供销存网站。

对于 WiFi 密码的爆破只需要传入 WiFi 名称，让然后便可以对目标 WiFi 进行暴力破解，破解成功会返回密码，并且自动连接到目标网络。 RAR 压缩文件爆破需要指定压缩文件的所在路径，然后系统会调用同目录下的 UnRAR.exe 进行解压测试，并夹杂着从字典文件里遍历的密码进行爆破。 SSH 爆破依然需要指定目标主机，输入 IP 便开始对目标的 22 号端口进行连接，一旦连接成功便返回成功的 root 密码。

而对靶场网站进行账号密码爆破时，只需要将目标网站的 url 输入到文本框，系统会多线程分任务的爆破目标网站。爆破成功后会返回成功的用户名和密码，用户使用该账户便可以登录蜗牛供销存这个靶场网站。

3.3.4 扫描模块功能性

扫描模块主要由基于 Socket 的端口扫描、基于 Scapy 的端口扫描、基于 Ping 命令的 IP 扫描、基于 Scapy 的 IP 扫描、基于 Ping 命令的子域名扫描、基于 Socket 的子域名扫描、 Web 站点信息搜集以及 XSS 漏洞扫描组成。

用户可以在端口扫描模块，选择是基于 Socket 的还是 Scapy 的扫描方式。同样地，在 IP 扫描也可以选择是基于 Ping 命令的扫描或是基于 Scapy 的扫描，子域名扫描同理。 Web 站点信息搜集则是需要输入站点的域名，随后会返回该域名的 whois 信息。 XSS 漏洞扫描是针对存在 XSS 漏洞的 XSS 测试平台进行扫描，会返回当前网页所有可能利用的恶意 payload.

3.3.5 泛洪攻击模块功能性

泛洪攻击模块集成了多种泛洪攻击模式，适用于不同场景下的泛洪攻击测试。本系统包含的泛洪攻击模式有：socket 三次握手泛洪、scapy 半连接泛洪、TCP Land 泛洪、ICMP 泛洪、ARP 泛洪攻击和欺骗。

对于基于 socket 三次握手、scapy 半连接、TCP Land 和 ICMP 的泛洪用户只需要输入目标主机的 IP 便可以对目标主机发起泛洪攻击。攻击效果表现为目標主机 CPU 上升，提供的 web 服务响应速度慢。如果可疑通过抓包软件分析，可以发现大量的泛洪数据流向目标主机。

ARP 欺骗比 ARP 攻击要更进一步，为了达到窃听的效果，需要让流量流经第三方主机，因此要对 ARP 解析的数据包进行编辑，而用户只需要输入目标主机、网关和第三方监听主机的 IP 地址即可。然后用户可在第三方主机进行抓包分析目标主机发往网关的流量信息。

3.3.6 DOS 攻击检测和防御

系统的 DOS 攻击检测主要是从三方面入手的。第一方面是采集 CPU 平均负载，用到的命令时 `uptime`；第二方面是端口的连接数量，使用命令 `netstat -ant`；第三方面是队列长度，使用 `ss -lnt` 命令。将这三方面数据采集工作分别封装到三个方法中，然后判断可疑的 DOS 攻击 IP。

系统的 DOS 攻击防御是最原始的也是最好用的——调用防火墙策略封禁 IP。这里需要注意靶场测试环境已经开启了防火墙，可以使用 `systemctl status firewalld` 命令测试。并且防火墙还要对测试的 3306 端口放行，可以使用命令 `firewall-cmd --add-port=3306/tcp --permanent` 进行放行操作，最后别忘记重载防火墙。

3.3.7 爬虫模块功能性

爬虫功能模块主要包含三大常见爬虫：图片爬虫、文字爬虫、超链接爬虫。本系统考虑了一定的实用性，图片爬虫选取的是《英雄联盟》官方网站的全皮肤爬虫；文字爬虫则是选取当当网 2020 到 2023 年每年的畅销书前 500，并将爬取结果保存到 Excel 文档中；超链接爬虫是爬取华云英格 IT 教育培训机构的官网所有超链接，并访问该链接，把页面保存到本地文件夹中。

3.4 非功能性

本系统的爆破模块非功能性有目标靶场环境的搭建、Python 的文件读写操作、Flask 的路由选择等组成。爆破模块的密码账号爆破，是基于 Tomcat 和 MySQL 的网站。多线程的爆破 WiFi 密码、多线程的爆破 RAR 压缩文件密码、多线程分任务的爆破蜗牛供销存的网站账号密码都是大量使用了 Python 循环读写文件的操作。

本系统的扫描模块非功能性有 Scapy 底层数据包的构建，Scapy 数据包的构建格式从左到右依次为数据链路层、网络层、传输层、应用层。为了使扫描速度更快，不能单纯的使用逐一线程的扫描，而是采用多线程分任务的形式进行。同样在用户观感上，也要对系统的可操作性进行改良，尽可能使用下列选项，方便用户选择适用的功能模块。

对于安全性需要求，系统目前尚未开放到公网，也许后续版本会使用安全壳协议对网站进行安全保证。该系统设计之初是打算实现一个用户登录的功能的，这样方便管理用户行为，也可以对用户的安全性需求进行满足。

本系统的突出点是性能需求，对无论是爆破还是扫描，都加入了多线程分任务的模式运行，极大的提高了运行效率，节省了爆破和扫描时间。

可拓展性依然被考虑在系统设计中，本系统的各个模块代码逻辑全是封装在不同的函数中，主程序只调用了一个 `app.run()` 方法，再加之路由装饰器的设计使系统也具有很好的可拓展性，可以通过新的 URL 地址拓展新的功能。

本系统也非常注重可用性需求，为用户提供了非常简洁友好的操作页面，使得用户在进行渗透测试时可以简单输入几个参数，便可以进行渗透测试工作，而无需打开多个软件，查阅资料输入相关操作命令。

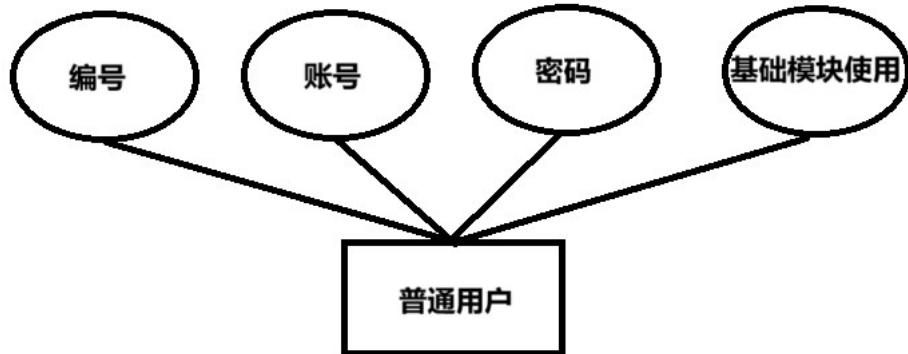
对于可靠性需求，该系统也是做了备份工作，一旦系统遭到入侵，可以立刻启动备份文件，快速恢复服务。

3.5 数据库设计

本系统的重点是安全脚本的开发，因此对系统的用户管理系统进行了简单的数据库设计。数据使用 MySQL 的 5.6 版本，库名为 `redteam`，使用了两张表，一张表保存普通用户的信息，另一张表保存会员用户的信息。会员用户相较于普通用户拥有一些其他模块的功能使用权，而管理员用户则对其他所有用户有管理权。普通用户仅仅对本系统的前四个模块具有使用权，因此没有太多的属性信息，普通用户使用账号密码登录使用系统，其中主键是编号。以下是普通用户的是数据项表格和 E-R 图：

表 3-1 普通用户数据项

属性	存储代码	类型	长度
编号	U_id	varchar	10
账号	U_username	varchar	20
密码	U_password	varchar	100



而会员用户相较于普通用户有了使用 DOS 检测和防御与爬虫模块的特权，其中主键是会员编号。因此在数据项中添加了会员到期时间，其格式为 datetime，采用的是倒计时规则，也就是表格中的日期是截止日期。并且会员用户的 E-R 图的分支相较于普通用户有所不同，首先是编号设置从编号变为会员编号，然后多出一个新的分支到期时间，对于功能模块也被替换为全部模块使用。全部模块相较于普通用户的加密模块、爆破模块、扫描模块和泛洪攻击模块，增加了 DOS 检测和防御模块和爬虫模块。并且扫描模块也开放了对网站是否存在 XSS 漏洞的扫描功能。以下是会员用户的数据项表格和 E-R 图：

表 3-2 会员用户数据项

属性	存储代码	类型	长度
会员编号	V_id	varchar	10
账号	V_username	varchar	20
密码	V_password	varchar	100
到期时间	V_deadline	datetime	20

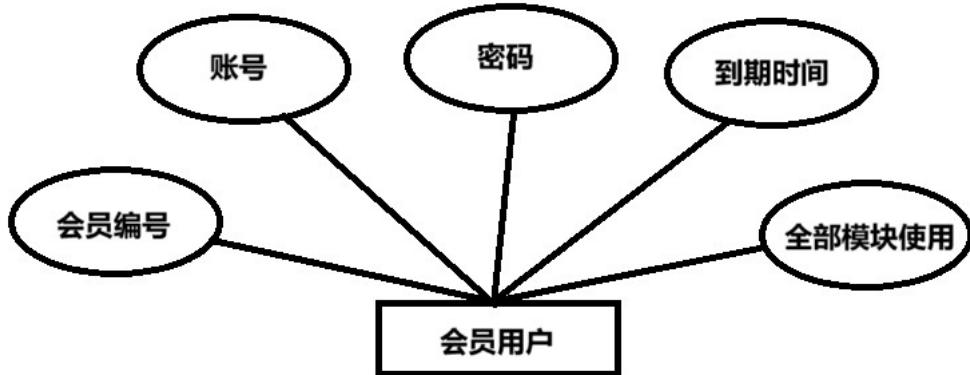


图 3.2 会员用户 E-R 图

在爆破模块的账号密码爆破功能实现上，已经在靶场里搭建了基于 Tomcat 和 MySQL 数据库的网上服装售卖系统，那里的数据库设计更加全面，但是作为靶场环境，不属于系统本身的数据库，也变不在此多做介绍。

4 详细设计

4.1 系统主页设计详述

系统的主页也是基于 Flask 框架全栈开发的开端和基础，系统的整个图形化界面展示便是由主页个模块的超链接连接起来的。系统主页将各模块的功能给统一罗列在头部标题栏中，点击各个模块便会跳转相应模块详情页，进而输入相关指令执行操作。

主页自然要美观一些，于是在主页文件 index.htm 头部链接了 css 样式。主页的 body 部分使用一个大标题加部分段落文字对系统进行使用说明，然后使用标签嵌入一张图片，使系统主页内容更加丰满。各部分超链接全部放置在 HTML 文件的标签和标签包裹的<a>标签内。在主页底部的<foot>标签嵌入系统的说明信息。如下展示系统页面：



图 4.1 系统主页

4.2 加密模块设计详述

在任何加密前，需要对编码格式进行处理。对于 Base64 编码和 MD5 摘要算法直接导入 Python 内置的库即可，唯一需要注意的是需要将传输的文本内容进行和编译器统一格式的编码，比如统一使用 UTF-8 编码。因此需要在数据 data 后进行编码：data.encode(encoding='utf-8')。通过代码可见 md5() 函数先实例化一个对象，然后将传入的数据进行 UTF-8 编码处理，再调用 update() 函数即可完成该数据的 md5 摘要生成。

4.2.1 对称加密功能的实现

对称加密也被称为私钥加密或者单密钥加密。顾名思义，其加密和解密是使用相同的密钥来完成的。这种加密算法的特点不单单是密钥相同，其优势特点在于算法公开且计算量小，因此网络传输效率也高。

对称加密的应用场景十分广泛，不仅被广泛用于网络传输加密，还被用于文件加密、数据库加密、移动设备加密等。然而，对称加密也有一个致命的缺点，一旦这个单一密钥被泄露，那么第三方可以破解所有加密的数据。

常见的对称加密算法有 AES（高级加密标准）、DES（数据加密标准）和 3DES（三重数据加密算法）等。这些算法在各个领域得到了广泛的应用，包括网络通信、数据存储和文件传输等。

虽然对称加密在性能上通常优于非对称加密（如 RSA），但由于其密钥管理的复杂性（需要确保密钥的安全传输和存储），在实际应用中，通常会结合使用对称加密和非对称加密来确保数据的安全性和传输效率^[12]。例如，在 TLS/SSL 协议中，就采用了这种混合加密方式。

值得关注的是 DES 和 AES 这两个对称加密的密钥选取以及明文修改为 8 的倍数操作。DES 的密钥必须为 8 位，如本系统实验案例选取的密钥为：MyKey = bqwer1234。然后就是对于明文要进行 8 的倍数处理，用明文数据的长度模 8 可以得到余数，8 再减去该余数得到还差几个字符可以凑满 8 的倍数，因为用等号补齐，所以再乘上等号，最后加上原来的明文数据，刚好是 8 的倍数。核心代码为：data = data + (8 - len(data)%8)*=。AES 的明文数据同理，不过要注意因为 AES 密钥要为 8 的倍数，这样明文数据要凑成密钥的整数倍。核心代码为：data = data + (len(key) - len(data)) % len(key) * =。下面展示 AES 的加密解密函数功能前端页面：

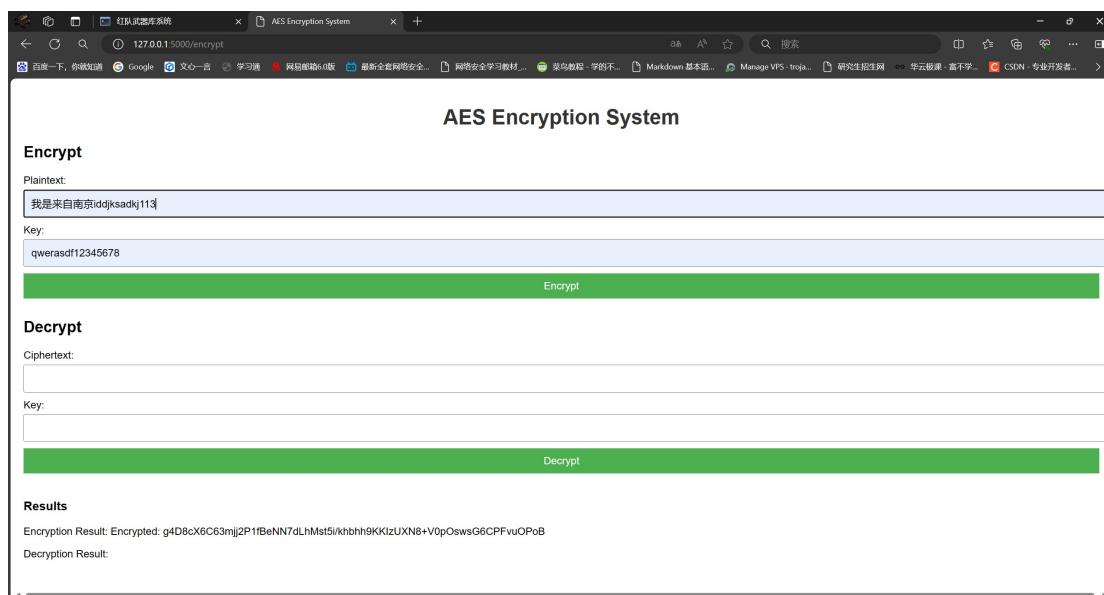


图 4.2 AES 加密模块

4.2.2 非对称加密功能的实现

首先需要 pycharm 导入 rsa 模块和 binascii 模块，然后生成 RSA 的公钥和私钥，其次进行公钥加密，最后还可以再进行私钥解密。由于 rsa 库已经封装好了函数，直接调用即可，下面展示系统调用策略代码：

```
# 第一步：生成公钥私钥
pub, priv = rsa.newkeys(2048)
```

```
# 第二步：公钥加密
encrypt = rsa.encrypt(GoodBye-NUIST.encode(), pub)

# 第三步：私钥解密
decrypt = rsa.decrypt(a2b_hex(encstr), priv)
```

需要注意，由于 RSA 算法的限制，它通常不直接用于加密大量数据^[13]。事实上，目前网络安全传输协议采用的是混合加密系统，往往利用 RSA 算法的严密性来加密一个对称加密算法的密钥，比如最常用的 AES 算法。然后用该对称密钥对实际传输的大量数据进行加密，这样兼顾了 RSA 算法的安全性和 AES 对称加密算法的效率。

4.2.3 勒索病毒模拟

由于勒索病毒的原理就是对文件进行加密处理，所以本系统模拟的也就是对文件的加密解密操作。核心技术是先将该文件进行 Base64 转码并加密保存，然后将字符右移 5 位，核心代码为：dest += chr(ord(c)+5)。将加密的字符串保存到文件中，为其追加.enc 后缀，最后删除原文件。



图 4.3 加密前的 jpg 文件

具体操作流程是先将文件以数据流的形式读到 Python 的句柄中，然后对数据流进行 Base64 编码，此时句柄中保存的文件是一长串字符串，然后设置一个空的字符串，用于追加加密后的数据流字符串。将 Base64 编码后的每个字符取其 ascii 码的序列号，将其右移五位再转换为字符，这样该文件打开后便是一堆乱码。最后将加密文件保存

到原文件所在处，添加加密后缀 `enc` 并将原文件删除。加密后便是 `enc` 结尾的加密文件（如下图所示），图片便无法查看。

名称	修改日期	类型	大小
<code>_init_.py</code>	2021/10/19 15:51	PY 文件	0 KB
<code>Crypto.py</code>	2024/5/9 22:42	PY 文件	3 KB
<code>extortion.py</code>	2024/5/10 14:51	PY 文件	2 KB
<code>test.jpg.enc</code>	2024/5/10 14:51	Wireshark capture f...	43 KB
<code>test.pdf</code>	2024/4/19 16:53	WPS PDF 文档	2,855 KB

图 4.4 加密后的 `enc` 文件

4.3 爆破模块设计详述

爆破的前提是拥有一个合适的密码字典，因此系统实现了一个简易的字典生成器。首先设置一个字符串，里面的字符便是接下来的密码组合。例如：`words = 1234567890`然后调用 `itertools.permutations(words, 8)` 其中的 8 代表复杂度是 8。也就是生成的密码都是 8 位的由 0-9 这个 10 个数字构成，当然也可以扩大 `words` 的字符类型，但是字典的生成时间就需要延长，这在实际生产中时间长一些不影响。本技术主要的还是文件写入操作循环的再文件里追加生成的密码，注意细节便是换行符要追加进去，这样使得密码本更具有可读性。

4.3.1 爆破 WiFi 密码

本技术首先需要导入 `pywifi` 包，系统操作分为两部分，一部分是连接函数，一部分是密码本文件读写。重点在于连接函数，第一步是抓取网卡接口 `wifi = pywifi.PyWiFi()`，然后是获取第一个无线网卡 `ifaces = wifi.interfaces()[0]`，然后需要断开所有连接 `ifaces.disconnect()`。这时候为了防止被检测出来进行密码爆破，让系统睡眠一秒 `time.sleep(1)`

第二步需要判断连接状态是否为未连接，否则直接返回系统已经连接成功^[14]。判断的代码为：`if wifistatus == const.IFACE_DISCONNECTED:` 在判断语句里，要进行 WiFi 连接文件的创建以及设定要连接的 WiFi 名称，最重要的是正确选择 WiFi 加密算法，WiFi 的加密算法是 `wps`，并且设定加密单元。最后就是调用传入的密码，记住要删除所

有连接过的 WiFi 文件，设定新的连接文件，设定一下连接时间，防止被检测出来爆破行为遭到封禁。对于读取密码本的操作是循环读取文件，并且传输密码给连接函数，一旦连接成功，系统终止，显示连接成功。

4.3.2 爆破 RAR 压缩文件

该项技术主要是使用 `rarfile` 库的 `extractall()` 函数进行解压测试，系统循环调用密码本的密码，进行解压测试，一旦解压成功，便会将事先设定好的 `flag` 置为 `False`，这样系统终止，否则一直遍历密码本的密码直至遍历结束。

在完成该功能的实验过程中，也是遇到了一些困难的，最主要的是该 `py` 文件目录下必须包含 `UnRAR.exe` 文件，否则无法进行解压。该文件可以在网上轻易找到下载链接，下载后复制到项目文件所在目录即可。同时该项功能的多线程技术也是很讲究的，不像后文要介绍的扫描类多线程那样先设定多少进程。

于是可以看到它是在逐行读取密码的循环中开启线程的，同时还注意到了使用 `join()` 方法，这样 Parent 父线程会等待 child 子线程运行完再继续运行，这样可以有条不紊的开展多线程工作，不至于漏读密码或者重复读取密码导致资源浪费。

4.3.3 爆破网站账号密码

测试网站是用 Tomcat 搭建的销售平台，因此在下图系统账号密码爆破模块输入该网站的 url 地址便开启对该网站的爆破。



图 4.5 输入目标网站 url 开始爆破

由于用户名和密码均未知，因此爆破采用多线程分任务的方式来提高效率，节省爆

破时间。首先使用 `file.readlines()` 函数将用户名字典 `username-top500.txt` 逐行存取为一个列表 `user_list`, 然后让每个线程负责 10 个用户的爆破。这里需要注意的是, `args=(sublist,)` 多写一个, 表示元组。然后是爆破函数的实现, 由于账号密码均未知, 因此采用双循环的方式进行爆破, 即外围循环遍历用户, 内层循环遍历密码。同时观察登录页面发现存在验证码, 并且通过测试发现验证码存在万能验证码漏洞, 因此验证的 `verifcod` 置为 0000 即可。同时根据登录成功的测试数据包发现, 服务器会返回一个 `login-success` 的值表示登录成功, 反之登陆失败则会返回 `login-fail` 值。因此, 可以通过判断 `resp.text` 里是否包含 `login-fail` 来判断是否登陆成功。由于账户爆破需要维持 `session` 状态, 因此使用 `session = requests.session()` 来发起登录请求。基于网站用户密码的爆破的会尽可能的找到所有用户的账号密码, 因此会花费大量时间进行字典扫描。

考虑可用性和测试可读性, 爆破出一组账号密码便输出一组信息, 如图下图所示。系统会继续寻找下一个用户, 直至密码本爆破完毕。



图 4.6 爆破成功一组账号密码

4.3.4 爆破 SSH 密码

尽管现如今为了主机安全, 许多服务器都不再使用密码的方式登录 ssh, 而是采用证书的方式进行登录, 但是 ssh 密码爆破也可以作为服务器的一种漏洞进行尝试。基于 Python 的 ssh 爆破需要导入 `paramiko` 库, 遍历字典文件, 循环调用 `paramiko.Transport()` 函数尝试连接目标服务器即可。连接成功后会输出正确的密码。



图 4.7 SSH 密码爆破成功

4.4 扫描模块设计详述

4.4.1 基于 Socket 的端口扫描

进行端口扫描可以发现网络设备上的开放端口，确定哪些端口是开放的。这些开放的端口可能暴露系统的服务和应用程序，成为潜在的攻击点。了解这些开放端口有助于渗透测试工作者更好地了解网络的安全状况。端口扫描不仅可以识别哪些端口是开放的，还可以识别在这些开放端口上运行的服务或应用程序。这有助于渗透测试工作者了解网络设备的功能和配置情况，从而更好地进行网络管理和安全维护。通过端口扫描，可以识别存在的漏洞和潜在的安全威胁。这些漏洞和威胁可能是由未打补丁的软件、配置错误或恶意软件引起的。通过及时发现这些漏洞和威胁，渗透测试工作者可以采取相应的措施来修补漏洞或防范威胁，提高网络的安全性^[15]。端口扫描是评估网络安全性的一种重要手段。通过扫描网络中的设备，可以了解网络的安全状况，发现潜在的安全风险，并为制定网络安全策略提供重要参考。通过端口扫描，管理员可以了解各种服务的运行状态，确保这些服务正常运行，并及时发现服务故障。

为了提高扫描效率，节省时间采用多线程分任务端口扫描。可用端口为 65535 个，划分为 650 多个线程每个扫 100 个端口。首先要实例化一个 socket 对象，然后进行 connect 连接。如果没有抛出异常则该端口可用，否则 pass。

4.4.2 基于 Scapy 的端口扫描

半连接扫描（SYN 扫描）是指在源主机和目标主机的三次握手连接过程中，只完成前两次握手，不建立完整的连接。Scapy 半连接扫描是基于半连接 S / SA / RA 等标志

位来对端口进行判断。如果目标端口开放，则 S->SA，如果目标端口未开放，则 S->RA。因为 Scapy 可以构造底层数据包，因此通过修改指定源 IP 地址，可以实现 IP 欺骗，进而导致半连接，此类操作也可以用于 Flags 参数定义上。以下展示基于 Scapy 半连接的端口扫描的功能页面和扫描结果：

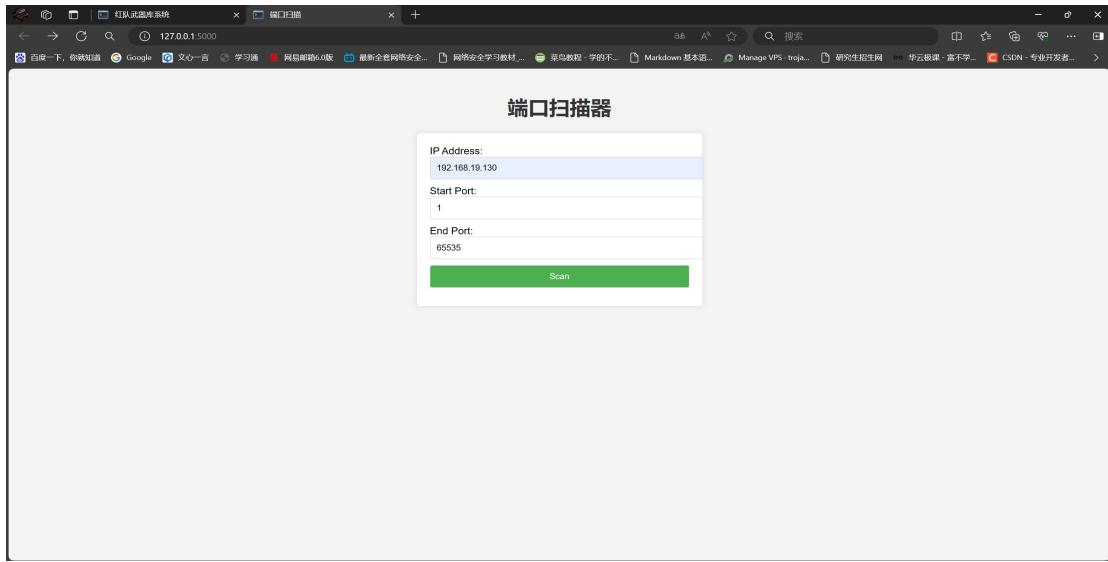


图 4.8 Scapy 半连接端口扫描

4.4.3 基于 Ping 命令的 IP 扫描

在公网上，不会进行 IP 扫描，通常是明确目标 IP 而进行端口扫描^[16]。如果要进行内网渗透，则必须要知道有哪些 IP 地址是存活的，可访问的，进而再进行端口扫描。IP 地址工作在 IP 层，ICMP，还有 ARP 协议也存在 IP 信息。先使用 ping 命令进行 IP 探测，但是此扫描方式存在 Bug，一旦防火墙禁止 ICMP，那么扫描结果失效。这里使用到了系统方法 `popen()` 这个方法和 `system()` 方法类似，唯一的不同是 `popen()` 不产生回显，因此需要一个变量来接收。也正因为如此，`popen()` 具有很好的隐蔽性。在进行 ping 命令的时候，如果目标主机可以 ping 通，那么会产生 TTL= 的特征，因此判断目标主机是否存活的策略便是使用一个 if 判断查看回显是否带有 TTL= 的特征。

为了提高 Ping 命令扫描 IP 的效率，需要指定相应参数，使只发送一个数据包，数据包的大小为 100 即可对连通性完成测试。因此系统使用的 ping 命令为：`ping -n 1 -w 100 {ip}`

4.4.4 基于 Scapy 的 IP 扫描

如何使用别的方式，让防火墙不存在封锁的行为呢？通过了解 ARP 协议，构造的

Scapy 底层数据包可以避免被防火墙屏蔽。基于 Scapy 的 IP 扫描主要涉及使用 Scapy 库发送和接收网络数据包，以探测和识别网络上的活动设备或主机的 IP 地址。

首先是构造 Scapy 数据包，`pkg = ARP(psrc=192.168.19.1, pdst=ip)`.然后使用 Scapy 终端命令的 `sr1()`函数发送该数据包，并将接收到的信息保存到变量 `reply` 中。最后尝试打印 `reply[ARP].hwsr`c 字段，打印成功则说明目标 IP 存活，否则说明目标 IP 不存在，继续遍历下一个 IP。

在系统页面输入目标网段，随后系统便将搜集到的存活 IP 返回，如下图所示。



图 4.9 基于 Scapy 的 IP 扫描

4.4.5 的子域名扫描

首先介绍一下系统基于 Ping 命令的子域名扫描。准备好一份常用子域名字典，然后将其读取到一个列表中。遍历该列表，并将其常用子域名拼接在二级域名前，使用 `popen()`方法 ping 该组合的域名，如果该域名存在，那么有两种情况，第一种是请求超时，第二种是返回 TTL，所以和基于 Ping 命令的 IP 扫描类似，通过查看回显文本里是否含有请求超时或者 TTL= 来判断是否存在该子域名即可。

然后是基于 Socket 的子域名扫描。基于 socket 库的 DNS 解析功能实现扫描，类似的，首先是准备好一份常用子域名字典，然后将其读取到一个列表中。遍历该列表，并将其常用子域名拼接在二级域名前，然后使用 socket 自带的 `gethostbyname()`方法尝试获取该域名的 ip 地址，如果获取成功，说明该域名存在，否则报错说明域名不存在。如下展示扫描出来的华云英格教育培训机构的子域名信息：

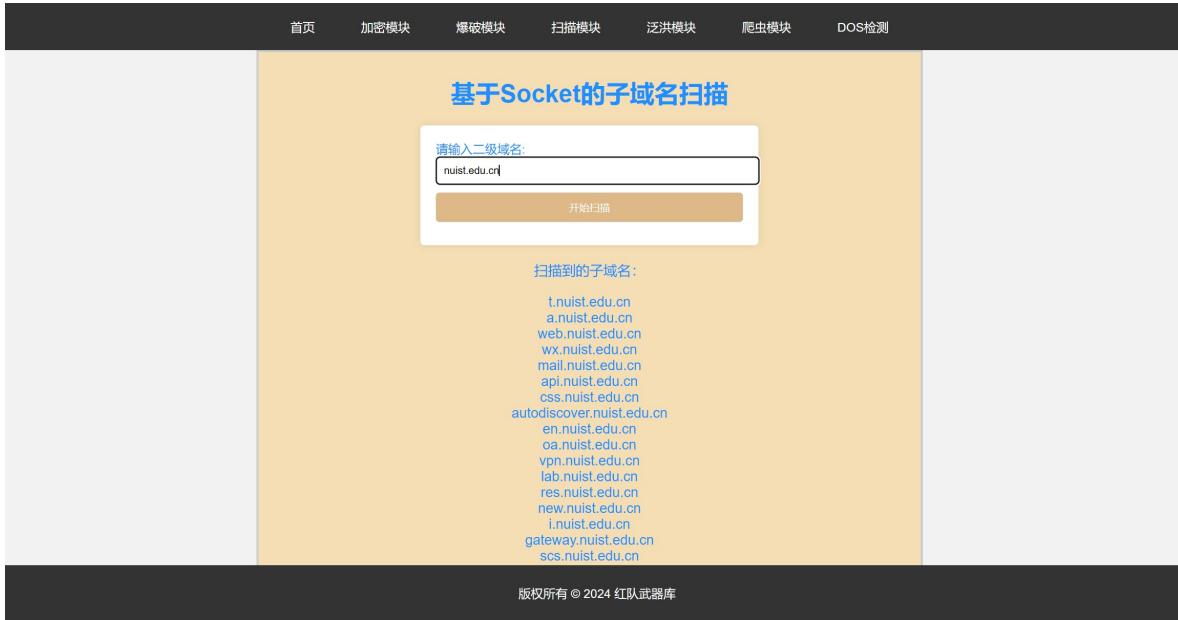


图 4.10 基于 Socket 的子域名扫描

4.4.6 Web 站点信息搜集

系统通过调用 whois 函数来实现对目标域名信息的搜集。虽然早期的 whois 查询多以命令列接口存在^[17]，不过现在可以轻松在网上找到简洁化的 web 端查询程序。事实上，这些 Web 查询应用的网页接口仍然是依赖 whois 协议向服务器发送查询请求。whois 不是一个扫描工具，而是一个查询工具。whois 通常使用 TCP 协议 43 端口，每个域名/IP 的 whois 信息由对应的管理机构保存，因此使用 whois 协议可以查到与之相关联的信息，比如注册人、注册商和注册日期等。本系统的站点信息搜集主要依赖 whois()函数实现，将其功能封装为本系统的一个函数便于调用。

4.5 泛洪模块设计详述

4.5.1 TCP 三次握手泛洪

本系统测试用 MySQL 服务器的 3306 端口连接，采用多线程进行泛洪攻击。SYN 泛洪攻击针对 TCP 三次握手的弱点，通过发送大量带有伪造源地址的 SYN 连接请求，耗尽目标主机的资源，使其无法为正常用户提供服务。

在 TCP 三次握手过程中，当服务器发出 SYN+ACK 应答但未收到客户端的 ACK 时，会等待一段时间（SYN Timeout，通常为几分钟）后放弃该未完成的连接^[18]。SYN 泛洪攻击正是利用这一点，结合 TCP/IP 协议对源 IP 地址的完全信任，通过创建大量伪造 TCP 连接请求，形成大量半连接状态。

服务器之所以无法连接正常的请求，是因为消耗了大量的系统资源和网络资源去维护一张半连接的列表，严重情况下甚至会导致服务器的崩溃。因为 TCP/IP 协议是一个开放性的协议平台，于是在互联网环境越来越复杂的现在，其安全机制不足以保障服务的可靠性，这也是 SYN 泛洪攻击能够成功的原因所在。

4.5.2 Scapy 半连接泛洪

Scapy 的半连接泛洪是构造 Scapy 数据包，先随机生成一个端口，作为数据包的源端口，并将其标志位 flags 设为 S，表示只进行三次握手的第一次握手 SYN，然后服务器返回一个 SYN ACK 请求，客户端只发送 SYN 完不成三次握手。通过多线程大量发送连接即可。

使用 Python 的 random() 函数生成一个随机端口，将其作为构造的 Scapy 数据包的源端口，指定目标端口为测试的 3306，设置 Flag 参数为 S，代表发送 SYN 请求。最后用一个死循环，不断听发送数据包给目标服务器。构造的数据包为： pkg = IP(dst=192.168.19.132)/TCP(sport=sport, dport=3306, flags=S) 然后打开 Wireshark 抓包工具，发现目标服务器收到了大量的 SYN 半连接请求，基于 Scapy 的半连接泛洪攻击得到验证。

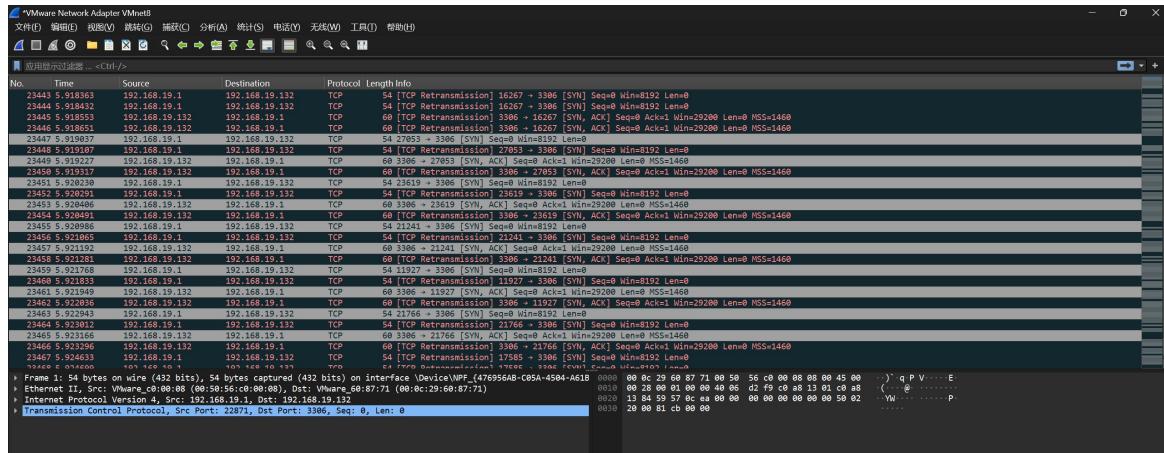


图 4.11 Wireshark 抓包发现大量 SYN 半连接

TCP Land 泛洪本质上是源地址和目的地址都是相同 IP，这样可以起到迷惑服务器的效果，算法性能不好的服务器可能因此而陷入崩溃^[19]。同样是先生成一个随机端口作为源端口，然后将数据包的源地址和目的地址均设为服务器的 IP 地址，标志位 flags 设为 S，代表发送 SYN 请求。因此系统后台构造的 Scapy 数据包为： pkg = IP(src=192.168.19.132, dst=192.168.19.132)/TCP(sport=sport, dport=3306, flags=S)

4.5.3 ICMP 泛洪

互联网控制消息协议（ICMP）泛洪攻击，作为一种传统的拒绝服务攻击（DoS 攻击）形式，虽然已不再是单一的主流攻击手段，但攻击者仍倾向于将其与其他策略结合，形成复杂难防的多媒介攻击。ICMP 泛洪攻击基于互联网控制消息协议，通过发送大量的 echo-requests（或 ping 请求）和 echo-replies（或 ping 响应）来测试网络设备的可达性和健康状况。

在 ICMP 泛洪攻击（亦称 ping 泛洪攻击）中，攻击者旨在通过发送海量的 ICMP 数据包，使目标网络路由器或特定 IP 地址的带宽达到饱和状态，或使设备因处理过多无效数据包而耗尽资源^[20]。当设备忙于处理这些无效的 ICMP 请求时，其用于响应合法请求或服务的资源（如内存、计算能力和网络带宽）将变得极为有限，从而导致服务不可用。

本系统的 ICMP 泛洪攻击同样是基于 Scapy 进行构造 ICMP 数据包来实现的。如果发送 ICMP 给广播地址，这样该网段内所有用户都会受到该流量包，这就是 ICMP 风暴形成的原理。在下图系统页面输入目标主机 IP，然后设置 payload 内容和数量进行 ICMP 泛洪攻击。



图 4.12 ICMP 泛洪模块参数指定

4.5.4 MAC 地址泛洪

本系统的 MAC 地址泛洪原理为本系统不停发送随机生成的 MAC 地址数据包，这

些数据包都会流向交换机，交换机看到这些陌生的 MAC 地址会将其记录在路由表中，最后填满路由表。随后进行广播，这样同网段便都可以收到数据包，方便以入侵到该网段的攻击者进行数据监听。首先是生成随机的 MAC 地址和该网段的 IP 地址：randmac=RandMAC(*: *: *: *: *: *)。然后使用 Python 的 random() 函数生成随机的源 IP 和目的 IP。最后构造 Scapy 数据包，设置其源 IP、目的 IP 和源 MAC 地址和目的 MAC 地址均为上述代码生成的随机 IP 和 MAC 地址，发送给该网段。代码如下：

```
packet=Ether(src=randmac, dst=randmac)/IP(src=srandip, dst=drandip)
sendp(packet, iface='VMware Virtual Ethernet Adapter for VMnet8', loop=0)
```

4.5.5 ARP 泛洪攻击和欺骗

ARP 攻击是利用地址解析协议进行的网络攻击行为，系统的 ARP 攻击会短时间在内网环境中发送大量的 ARP 流量包，严重情况下将会导致局域网环境断网。而 ARP 欺骗在一定程度上非常依赖伪造的 IP 地址和 MAC 地址，攻击者不断的发送错误的伪造 ARP 数据包进行响应，这样可以导致本地局域网环境紊乱，进行错误的地址解析，进而导致中间人攻击。

攻击者通过发送伪造的 ARP 响应流量包，不断的来篡改目标主机对于 ARP 的缓存条目，改变 IP 和 MAC 的对应关系。在局域网环境中，如果有一台机器不断向其他机器，特别是网关，发送无效的、伪造的 ARP 应答信息包，就可能造成严重的网络堵塞^[21]。严重情况下甚至会导致局域网交换机的崩溃，彻底摧毁目标网络的网络环境，使得该网络下的用户全部断网。

简而言之，ARP 攻击就是将 IP 对应的 MAC 地址解析为不存在的 MAC 地址，导致目标断网。ARP 欺骗，就是攻击机不停的发送 op=2 的响应数据包（op=1 代表请求），这样目标主机在请求网关时候，得到错误的 ARP 信息将数据包发送给了攻击机，而攻击机开启了 IP 转发，这样又会将数据包发送给网关，而网关在返回数据包时，也得到了错误的 ARP 响应信息，将数据包发送给了攻击机，最后攻击机转发给目标主机，这样目标主机无感的完成一次请求，但是流量流过了攻击机，完成 ARP 欺骗。

本系统的 ARP 是将一台 Windows10 作为被欺骗主机，而将流量经过第三方主机 Kali，因此在 Kali 的 wireshark 上可以检测到 Windows10 请求网关和访问网页的流量。在系统的 ARP 欺骗模块输入被攻击主机的 IP 和 MAC 地址，还有网关的 IP 和 MAC 地址，以及第三方窃听主机的 IP 和 MAC 地址即可完成 ARP 欺骗操作。

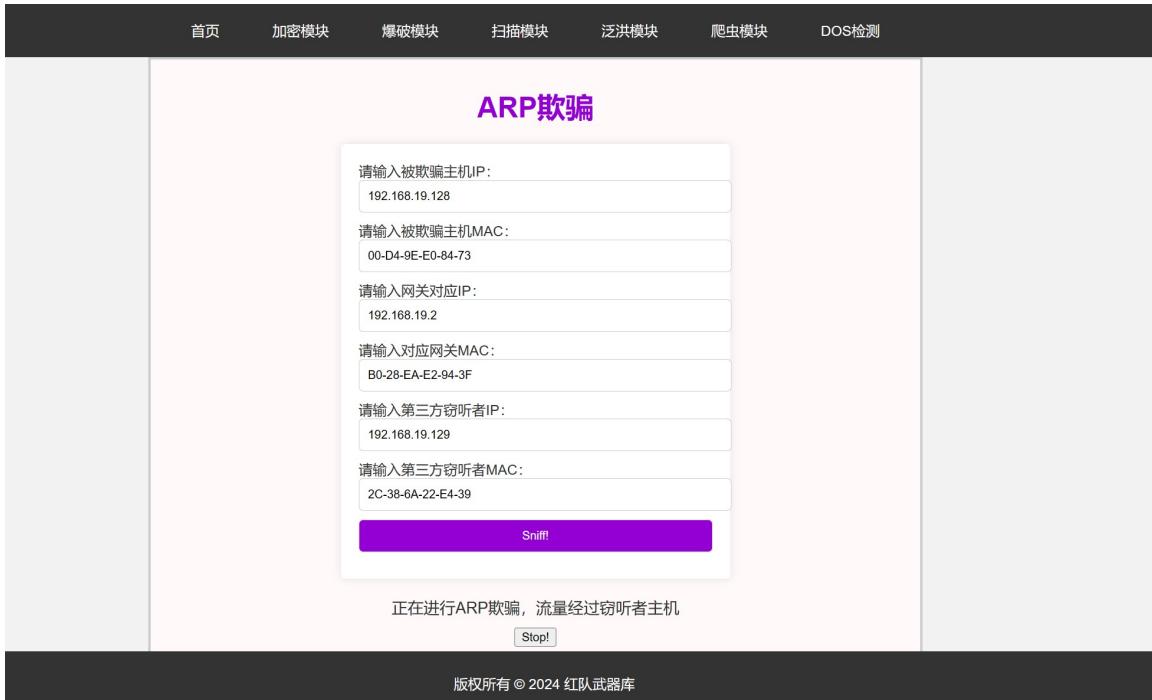


图 4.13 ARP 欺骗模块参数指定

4.6 DOS 检测和防御模块设计详述

在上一章详述了泛洪攻击的攻击过程，可以通过查看系统的 CPU 占用、网络连接数和队列等候长度等信息得以体现攻击成功。这也为 DOS 攻击的检测和防御提供了设计思路。总体思路便是对系统资源数据进行监控，然后获取可疑 IP 使用 Linux 系统自带的 firewall 添加策略进行 IP 封禁。

首先等装了三个函数来分别采集 CPU 的平均负载、端口连接数量和队列长度。对于系统的平均 CPU 负载是调用系统命令 uptime 实现的，然后使用 Python 的 split() 方法对回显结果进行分割提取 CPU 负载值。类似的，也是使用系统命令 netstat -ant 和 ss -lnt 来采集端口连接数量和队列长度。随后便对这些实时检测的数据进行判断，数据值过大便导出可疑 IP。

重点是对连接数量最多的 IP 地址的采集，首先使用命令 netstat -ant | grep : 3306 获取所有的连接数据，然后使用 split() 方法以换行符为界划分到列表中^[22]，然后切片获获取连接的 IP。这些 IP 全部保存在一个列表中，接下里就是要对列表进行排序。好在 Python 内置了 Counter() 方法，可以进行列表计数排序，这样便可以获取最多的连接 IP 也就是可疑的 DOS 攻击 IP^[23]。

最后便是对可疑 IP 进行防火墙封禁操作，直接使用防火墙命令：firewall-cmd --add-rich-rule=rule family=ipv4 source address={ip} port port=3306 protocol=tcp reject 进

行封禁。当在 Linux 服务器上运行该程序时，会显示可疑的 DOS 攻击 IP 已经被封禁。

在 Linux 服务器上启动该程序，便会实时检测采集的三组数据，一旦发现可以 IP，便为防火墙添加策略封禁可疑 IP。

```
CPU-Load: 22.79, TCP Conn: 203, TCP Queue: (0, 80)
CPU-Load: 22.79, TCP Conn: 342, TCP Queue: (81, 80)
当前系统TCP连接负载和CPU使用率过高，存在DOS攻击的可能性，可疑IP地址为: 192.168.19.1.
已经成功将可疑攻击源 192.168.19.1 进行封锁，流量将不再进入.
CPU-Load: 21.53, TCP Conn: 154, TCP Queue: (0, 80)
CPU-Load: 21.53, TCP Conn: 154, TCP Queue: (0, 80)
```

图 4.14 可疑的 DOS 攻击 IP 被封禁

为了今已确认可疑 IP 已经被封禁，可疑使用防火墙命令 `firewall-cmd --list-all` 查看如下图所示的防火墙策略表，确实添加一条策略对可疑 IP 的 `reject` 操作。

```
[root@centos-7 PythonProjects]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
    services: dhcpcv6-client ssh
    ports: 3306/tcp
    protocols:
    masquerade: no
    forward-ports:
    source-ports:
    icmp-blocks:
    rich rules:
      rule family="ipv4" source address="192.168.19.1" port port="3306" protocol="tcp" reject
[root@centos-7 PythonProjects]#
```

图 4.15 查看防火墙策略表

4.7 爬虫模块设计详述

4.7.1 图片爬虫

该系统的图片爬虫目标是网络游戏《英雄联盟》的全皮肤爬虫。首先需要进入英雄联盟官网主页，进入“资料库”栏目，使用浏览器的检查功能进行页面分析。在对应皮肤页面下右键查看页面源代码以及检查，进行页面分析，使用爬虫脚本对目标图片的 url 地址进行正则匹配。然后，在几个重要的 js 文件里利用正则表达式进行匹配，拼接得到正确的图片 url，通过 `requests` 的 `get` 方法访问下载。最后，Python 程序在本地创建多级文件夹，规整保存个英雄皮肤图片。值得注意的是，为了防止被网页检测到是爬虫，需要设置休眠时间^[24]，利用 `sleep()` 函数进行休眠控制。

第一步，进入英雄联盟官方站的资料库，然后会显示诸多游戏人物，先随便点击一个英雄。然后右键检查。在检查模块的 Network 栏目内，可以找到全部英雄的索引文件 `champion.js` 这个 json 文件。如图 4.16 所示该文件保存了所有英雄人物的 json 信息。

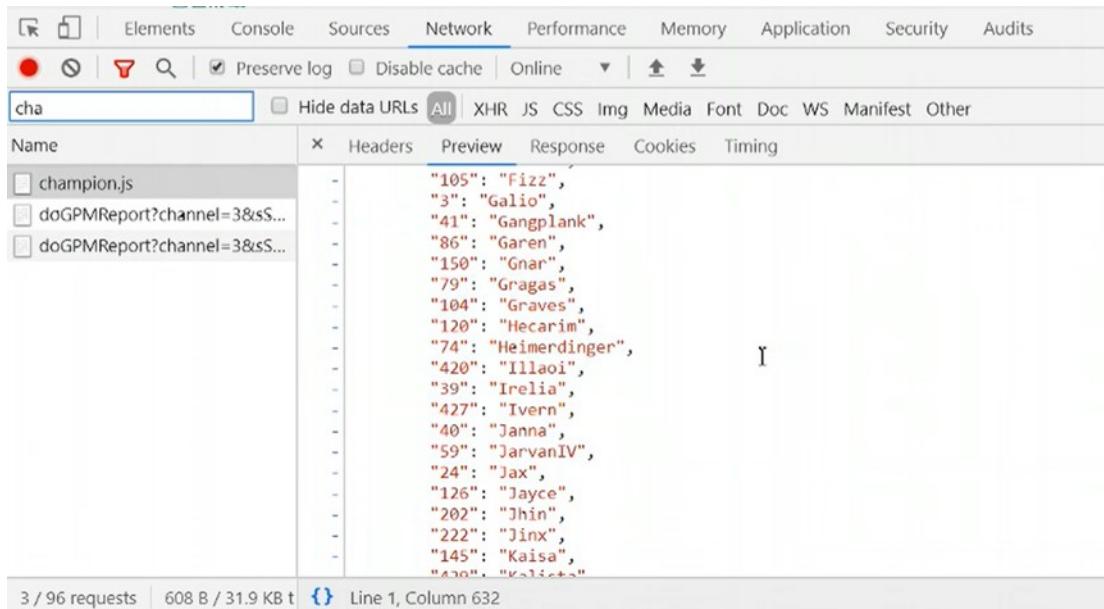


图 4.16 所有英雄的 json 信息

第三步，根据游戏人物的 json 信息使用正则表达式进行 url 匹配，再和 Headers 的 url 拼接起来，就是一个图片的完整 url 地址。拼接的过程中需要注意对特殊字符进行转义处理。

第四步，对于特殊的游戏人物的名称处理。因为系统是以游戏人物的英文名保存的，但是 Windows 系统对文件名有要求，不得出现斜杠等特殊字符，于是还要解决一些皮肤名称出现特殊符号而引起的错误，例如游戏里的 K/DA 系列皮肤和 M&B 系列皮肤。

#解决皮肤名称里的\引起的错误

```

name = name.replace(/\/, )
name = name.replace(\\, )

```

第五步，系统会调用函数自动创建文件夹分类保存每个英雄的多个皮肤。

```
if not os.path.exists(f./img/{n}):
```

4.7.2 文字爬虫

本系统的文字爬虫是爬取当当网的畅销书 top500，并将其保存到 Excel 表格中方便查看。首先是对当当网的页面进行源码分析，检查图书的页面信息。



图 4.18 当当网图书源码分析

对页面前端代码的分析，得知图书的基本信息包含在类名为 `bang_list_box` 的标签里。主要有书名、评论数、作者名称、价格、折扣、出版社、出版时间等使用 `xpath` 的格式将其信息记录下来，用于接下来追加到 Excel 表格中。

然后就是设计 Excel 的文件存取，首先是要定义表头，然后使用 `xlwt.Workbook()` 方法创建工作表。其次，利用一个 `for` 循环写入表头数据，一行行追加到当当网畅销榜 TOP500 书籍信息.xls 文件中。以下展示爬取当当网的畅销书前 500 的 Excel 表格：

year	Rank	name	pinglun	author	chuban	jiage	yuanjia	discount	
2018.1.	1.	活着	2980683	余华	2012-08-0	22.10	¥28.00	7.9折	
2018.2.	2.	我喜欢生	708550	周国平	2017-02-0	¥33.70	¥45.00	7.5折	
2018.3.	3.	神奇校车·第一季	1553087	乔安娜·柯尔	2014-04-0	¥150.00	¥150.00	10.0折	
2018.4.	4.	我的第一本哲学书	9882398	布鲁斯迪	2016-06-0	¥47.30	¥49.80	9.5折	
2018.5.	5.	人间失格	2226398	弗劳	2015-08-0	¥10.00	¥25.00	4.0折	
2018.6.	6.	小熊和最好朋友	1580077	郑利强	2007-11-0	¥17.50	¥35.00	5.0折	
2018.7.	7.	雪落香杉林	306409	余秋雨	2017-06-1	¥39.00	¥52.00	7.5折	
2018.8.	8.	少年读史记	1380845	步少印童书	2015-09-0	¥95.00	¥100.00	9.5折	
2018.9.	9.	神奇校车·第二季	1780921	李革治	2018-05-1	¥198.00	¥198.00	10.0折	
2018.10.	10.	你坏	1550213	杨伟	2018-06-1	¥39.20	¥39.60	9.9折	
2018.11.	11.	东野圭吾	2306629	阿兰·德·	2014-05-0	¥29.60	¥39.50	7.5折	
2018.12.	12.	浮生六记	1319088	亚历克斯·	2015-08-0	¥32.00	¥32.00	10.0折	
2018.13.	13.	正面管教	2149119	苏豫平	2016-07-0	¥38.00	¥38.00	10.0折	
2018.14.	14.	写给儿童的1132389条爱桐	2014-04-0	¥337.30	¥355.00	9.5折			
2018.15.	15.	作家榜名著	1578836	戴雅、仲伟	2017-01-1	¥39.80	¥39.80	10.0折	
2018.16.	16.	摆渡人	1354777	张嘉佳	2015-06-0	¥25.90	¥36.00	7.2折	
2018.17.	17.	马尔克斯	2386943	乔安娜·柯尔	2017-08-0	¥39.60	¥55.00	7.2折	
2018.18.	18.	追风筝的人	2864192	布鲁斯·	2008-05-0	¥35.60	¥36.00	9.9折	
2018.19.	19.	所谓情商	812583	大兵	2016-09-0	¥30.40	¥32.00	9.5折	
2018.20.	20.	生活需要点	588712	博集天卷	2017-12-2	¥25.20	¥36.00	7.0折	
2018.21.	21.	东野圭吾	2158856	东野圭吾	2017-08-0	¥35.20	¥59.60	5.9折	
2018.22.	22.	小王子	1012012	新经典	2013-01-0	¥23.00	¥32.00	7.2折	
2018.23.	23.	三体	全	2331222	安托万·德·	2010-11-0	¥92.10	¥93.00	9.9折
2018.24.	24.	围城	1218134	李维宏	1991-02-0	¥28.80	¥39.00	7.4折	
2018.25.	25.	万历十五年	914258	宋东文化	2006-06-0	¥26.00	¥26.00	10.0折	
2018.26.	26.	岛上书香	1320055	刘慈欣	2015-05-0	¥17.50	¥35.00	5.0折	
2018.27.	27.	自在独行	916398	钱钟书	2016-05-0	¥37.10	¥39.00	9.5折	
2018.28.	28.	魔法拼音	305808	黄仁宇	2017-05-0	¥98.00	¥98.00	10.0折	
2018.29.	29.	云边有个小呀	1636349	加·泽文	2018-07-0	¥23.10	¥42.00	5.5折	
2018.30.	30.	平凡的世界	1449684	读客文化	2017-06-0	¥81.00	¥108.00	7.5折	
2018.31.	31.	果麦经典	742125	贾平凹	2014-09-0	¥32.00	¥32.00	10.0折	
2018.32.	32.	我不	1760857	王开岭	2017-09-0	¥38.60	¥39.00	9.9折	
2018.33.	33.	月亮和六	621510	姜白露	2017-09-0	¥11.34	¥27.00	4.2折	
2018.34.	34.	半小时漫画	649959	张嘉佳	2017-04-0	¥39.90	¥39.90	10.0折	
2018.35.	35.	夏洛的网	985408	朱国华·路遥	2014-08-0	¥19.30	¥26.00	7.4折	
2018.36.	36.	摆渡人2	430009	余华	2017-09-0	¥30.80	¥42.80	7.2折	
2018.37.	37.	东野圭吾	823187	新经典	2016-11-0	¥26.30	¥39.50	6.7折	
2018.38.	38.	霍乱时期的	594686	戴尔·卡耐基	2015-06-0	¥37.10	¥49.50	7.5折	

图 4.19 当当网畅销榜 TOP500 书籍信息

值得注意的是，文件存取是发生在最后的，本爬虫程序是先将书籍信息存放在 `dflist` 列表中，等指定书籍信息全部爬取结束，再开始存取的 Excel 文件中，所以最后不再打印

正在爬取的书籍时会卡顿一会儿显示程序运行结束。

最后程序运行结束，便可以查看到生成的 Excel 文件里的内容为下图所示。之所以有 2000 条数据是因为是 4 年的 TOP500，正好是 2000 条数据。

4.7.3 超链接爬虫

本系统爬取的华云英格培训机构网站主页的所有超链接。首先爬虫用到的库有 BeautifulSoup4、requests、re 模块。然后使用 request 进行页面请求，获取页面的 html，值得注意的是 request 的 get() 函数返回的类型是 Response，事实上这个类是包含了 HTML 的，只不过还需要调用该类的 text() 方法才可以获取网页的源代码。而本项爬虫要找的超链接是在 DOM 树的 a 结点下的。以下是树结构的逻辑示意图：

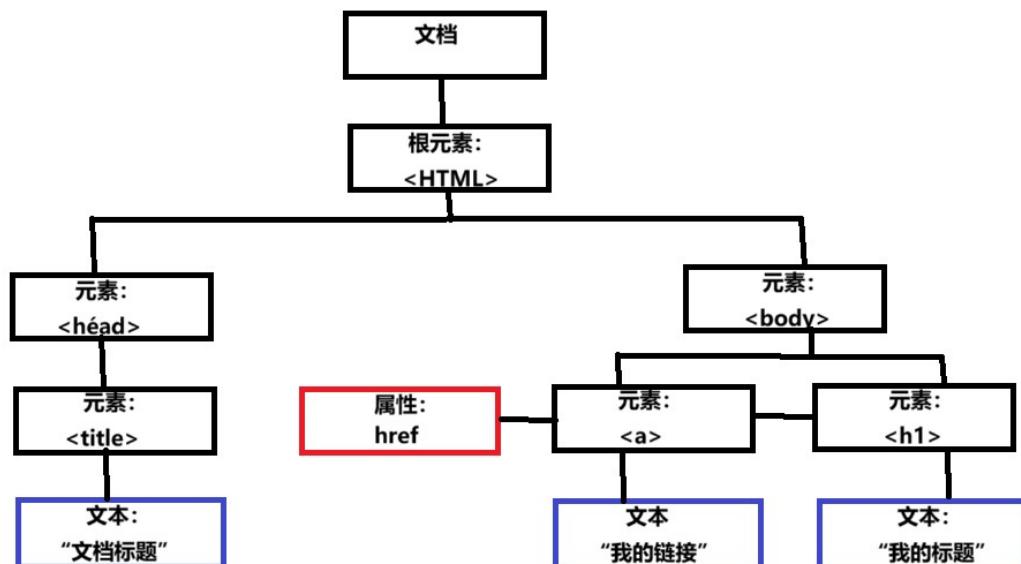


图 4.20 结构的逻辑示意图

通过上图知道了超链接在 html 中所处的位置，获取这些超链接主要有两种方法。一种方法是使用正则表达式把 html 当成字符串暴力处理，思路上就是类似 if <a + 其他字符> in html 的模式^[25]。以下展示正则表达式匹配规则代码：

```

resp = requests.get('https://ke.huayuns.com/')
links = re.findall(<a href=(.+?), resp.text)
  
```

另外一种是调用 BeautifulSoup 库强大的解析功能，在 html 树结构中进行定位。在程序代码编写过程中还需注意如下一些语法细节：

- BeautifulSoup()函数返回的是一个解析过的对象，相当于把访问的 html 文本做了处理；
- 调用这个解析过的对象的 find_all()方法，即返回一个该 html 中所有的 a 节点组成的列

表；再对表中每一个节点调用 get()方法，获得该节点下 href 标签（键）的所有值，也就是要找 http:// 打头的超链接了。

在实际实验过程中，系统爬取的 href 值不全是超链接，还会是一些文本，要将其过滤掉，比如观察到有些 href 值会是含参的：? p=这样的值在进行文件读取时会出错，因此要将? 替换为空。再者就是有的 href 值是以/开头的，要将主页网址和该地址拼接起来。拼接代码如下：

```
if link.startswith('/):
```

```
link = https://ke.huayunsys.com + link
```



图 4.21 特殊的 href 值

该爬虫程序的文件保存的文件名也是精心设计的，考虑到超链接可能有多个斜杠分割，于是用 split()方法分割为列表，取最后一个斜杠后的字符串作为文件名的一部分，然后让系统随机生成年月日加在后面，最后添加.html 后缀，形成完整的文件名。程序运行结束后，在当前目录的 page 文件夹内便可以看到如下图所示的保存文件。

名称	修改日期	类型	大小
20240508_120322.html	2024/5/8 12:03	Microsoft Edge HT...	45 KB
20240508_120323.html	2024/5/8 12:03	Microsoft Edge HT...	50 KB
20240508_120324.html	2024/5/8 12:03	Microsoft Edge HT...	42 KB
20240508_120325.html	2024/5/8 12:03	Microsoft Edge HT...	40 KB
20240508_120326.html	2024/5/8 12:03	Microsoft Edge HT...	217 KB
20240508_120327.html	2024/5/8 12:03	Microsoft Edge HT...	217 KB
20240508_120328.html	2024/5/8 12:03	Microsoft Edge HT...	65 KB
20240508_120329.html	2024/5/8 12:03	Microsoft Edge HT...	40 KB
20240508_120330.html	2024/5/8 12:03	Microsoft Edge HT...	45 KB
20240508_120331.html	2024/5/8 12:03	Microsoft Edge HT...	51 KB
20240508_120332.html	2024/5/8 12:03	Microsoft Edge HT...	51 KB
20240508_120333.html	2024/5/8 12:03	Microsoft Edge HT...	50 KB
20240508_120334.html	2024/5/8 12:03	Microsoft Edge HT...	50 KB
20240508_120335.html	2024/5/8 12:03	Microsoft Edge HT...	48 KB
20240508_120336.html	2024/5/8 12:03	Microsoft Edge HT...	48 KB
20240508_120337.html	2024/5/8 12:03	Microsoft Edge HT...	57 KB
20240508_120338.html	2024/5/8 12:03	Microsoft Edge HT...	57 KB
20240508_120339.html	2024/5/8 12:03	Microsoft Edge HT...	58 KB
20240508_120340.html	2024/5/8 12:03	Microsoft Edge HT...	58 KB
20240508_120341.html	2024/5/8 12:03	Microsoft Edge HT...	75 KB
20240508_120342.html	2024/5/8 12:03	Microsoft Edge HT...	49 KB
20240508_120343.html	2024/5/8 12:03	Microsoft Edge HT...	86 KB
20240508_120344.html	2024/5/8 12:03	Microsoft Edge HT...	51 KB

图 4.22 保存的超链接文件

5 总结

系统是基于 Python Flask 开发的 web 系统，集成了诸多渗透测试模块。主要包括加密模块、爆破模块、扫描模块、泛洪攻击模块、DOS 检测和防御模块以及爬虫模块。为用户提供友好可视化界面的同时兼顾系统性能，采用多线程分任务的形式进行爆破扫描类工作。本系统也不单单只是一个攻击系统，还可以针对 DOS 攻击做出检测和防御。系统设计过程中遇到的一些问题，也根据报错信息和代码调试得到了解决。

在加密模块的设计中，主要是调用了许多内置的库，因此技术难度并不高，关键在于整合好各类加密的封装方法，使系统可以有条不紊的完成各种加密解密操作。在了解了加解密基本原理之后，学习了勒索病毒的攻击原理。勒索病毒就是将目标主机的文件先做 Base64 编码，然后使用一定的加密算法进行加密。因此本系统也集成了该项功能，也是本系统对于加解密模块的一种拓展性应用。

在爆破模块的设计中，字典生成器写的有一些简单了，笔者会在后续版本中加入一些特征值，是生成的字典针对于某一目标具有高准确性。比如根据目标的身份证信息、家庭信息、学历信息、兴趣爱好、社交帐号、密码习惯等方面生成一套专门针对目标的密码本。该模块的账号密码爆破是在有验证码的情况下进行爆破的，好在目标网站存在万能验证码漏洞，否则对于账号密码的爆破会变得更为复杂，也许需要 AI 图片识别或者打码平台进行验证码识别来辅助爆破。

扫描模块的设计是考虑到了防火墙对于 ping 命令的屏蔽，进而采取构造 Scapy 数据包的方式绕过一些防火墙。Scapy 的半连接扫描是没有完成三次握手的，系统不断发送 SYN 请求但是对服务器的 SYN ACK 没有做出回应，因而完不成三次握手。对于子域名扫描是导入了一份常用子域名字典进行穷举扫描的。

泛洪攻击模块最值得注意的是 ARP 欺骗，本质上是系统为攻击不断发送 ARP 响应包，这使得无论是目标主机还是网关在进行数据应答时都将数据包通过错误的地址解析发送给了攻击机，而攻击机开启了 IP 转发，因此目标主机和网关都没有感知到数据包流过了第三方主机。

DOS 攻击检测和防御模块是系统兼顾防御性的体现。本模块脚本要在 Linux 系统上运行，实时检测 CPU 负载、连接数量和队列长度，一旦数据异常便进行防火墙封禁 IP 的操作。系统很简单，没有雷达等可视化界面图，但是也把 DOS 检测的底层原理搞清楚了，对于学习网络安全的防护也是很有帮助的。同样也为后续系统发展提供了方向，

继续实现实时的统计图等图形界面。

爬虫模块是系统功能的一种拓展，集成了图片爬虫、文字爬虫、超链接爬虫。无论哪种爬虫都需要对目标网页的源代码进行分析，然后定位爬取资源的位置信息。

言而总之，本系统的各个模块并非完全割裂，再一次渗透测试过程中，很可能用到多个模块的功能。该系统将各个阶段可能用到的工具集成在一个统一系统里，方便用户使用，极大的提高了渗透测试效率。

参考文献

- [1] 柴志刚.Python 软件供应链攻击自动检测系统的设计与实现[D].北京邮电大学,2023.DOI:10.26969/d.cnki.gbydu.2023.002099.
- [2] 丁洁.Python 脚本语言的 Web 开发应用探究[J].数字通信世界,2021,(10):163-164.
- [3] 李宗杰.Python 脚本语言在 Web 开发中的应用探究[J].电子元器件与信息技术,2020,4(12):136-137.DOI:10.19772/j.cnki.2096-4455.2020.12.067.
- [4] 邱洋 . 基于符号执行的 Python 攻击脚本分析 [D]. 上海交通大学,2016.DOI:10.27307/d.cnki.gsjtu.2016.000533.
- [5] 张露雨.基于 Web 渗透测试的 XSS 漏洞探索与研究[J].网络安全技术与应用,2024,(04):5-8.
- [6] 常会鑫 .Web 安全漏洞检测系统设计及优化 [D]. 华北理工大学,2023.DOI:10.27108/d.cnki.ghelu.2023.001284.
- [7] 张汝娴.基于强化学习的 XSS 漏洞模糊测试工具的设计与实现[D].北京邮电大学,2023.DOI:10.26969/d.cnki.gbydu.2023.001009.
- [8] 乔明秋,高松 . 基于 Kali Linux 的渗透测试及防范研究 [J]. 信息与电脑(理论版),2023,35(03):227-230.
- [9] 邱金水.基于安全技术的软件开发与运维研究[J].软件,2022,43(05):13-15.
- [10]陈志伟.基于 SSM 技术的网络安全渗透测试系统的开发 [J].鞍山师范学院学报,2021,23(02):57-60.
- [11]裴兰珍,罗贊骞,景勐,等.网络安全漏洞渗透测试框架综述 [J].电子信息对抗技术,2016,31(02):10-13+22.
- [12]吴羽翔.基于模块化设计的 Web 应用程序漏洞利用框架研究与开发[J].计算机光盘软件与应用,2014,17(04):161-162.

- [13]吉强 .Payton 在网络安全加密解密领域中应用研究 [J]. 网络安全技术与应用,2023,(12):27-29.
- [14]尚博.Python 在网络空间安全中的应用[J].电子技术,2022,51(07):40-41.
- [15]陈泽生 , 孙涛 , 周敏 , 等 .Python 脚本应用于网络安全防护 [J]. 中国教育网络,2022,(06):58-60.
- [16]郭瀚亭 . 基于 Django 框架的文件分享平台的设计与开发 [J]. 信息记录材料,2022,23(03):139-141.DOI:10.16009/j.cnki.cn13-1295/tq.2022.03.075.
- [17]李同金 . 基于 Python 的端口扫描技术研究 [J]. 电子世界,2022,(02):38-39+42.DOI:10.19353/j.cnki.dzsj.2022.02.015.
- [18]王昊.基于 python 对漏洞扫描任务自动化部署的研究[C]//中国通信学会无线移动通信委员会,中移铁通有限公司信息和产品开发中心.中国移动“5G+AICDE”技术研讨会论文集 . 中移铁通有限公司信息和产品开发中心;,2021:5.DOI:10.26914/c.cnkihy.2021.070267.
- [19]庄海燕.面向网络安全与执法专业的 Python 程序设计语言课程内容改革研究[J].电脑与电信,2021,(05):39-42.DOI:10.15966/j.cnki.dnydx.2021.05.011.
- [20]王晓东 .Python 安全应用程序开发方法研究 [J]. 福建电脑,2020,36(07):70-72.DOI:10.16707/j.cnki.fjpc.2020.07.017.
- [21]Abdullayeva F ,Suleymanzade S .Cyber security attack recognition on cloud computing networks based on graph convolutional neural network and graphsage models[J].Results in Control and Optimization,2024,15100423-.
- [22]Bazzoli E ,Criscione C ,Maggi F , et al.XSS Peeker: A Systematic Analysis of Cross-site Scripting Vulnerability Scanners.[J].CoRR,2014,abs/1410.4207
- [23]Omran A ,Dietrich S ,Abouelmagd A , et al.New ArcGIS tools developed for stream network extraction and basin delineations using Python and java script[J].Computers and Geosciences,2016,94140-149.
- [24]M B ,V P ,D C L , et al.Scripting MODFLOW Model Development Using Python and FloPy.[J].Ground water,2016,54(5):733-739.
- [25]Chang Z ,Liang H ,Cao L .Security bipartite synchronization of MASs resilient to DoS attacks[J].Transactions of the Institute of Measurement and Control,2024,46(8):1489-1499.

致 谢

基于 Python 的 Flask 框架开发的一套攻击脚本集合系统对我来说是一个挑战，也是一个契机。说它是挑战是因为我在大学四年的生活中，从未完成过工作量和难度这么大的系统设计。而且本科阶段对 Python 的编程也不是很熟悉，更不熟悉像 vue、BootStripe 这类前端框架技术。在此首先要感谢的是指导老师的技术指导和思路拓宽，能够在我焦头烂额调试代码的时候给我指点迷津，我觉得这是让我很感激的。更要感谢指导老师对我的信任和理解，使我能够自由发挥，以自己兴趣为基础去完成网络安全领域的安全脚本攻击系统的开发。再次对毕业导师献上诚挚的感谢！

其实在互联网高速发展的今天，网上有很多资源可供学习，虽然我花了大量的功夫去搜集高质量的教学资料，但是也是收获颇丰，不至于闭门造车毫无头绪。对 Python 安全脚本的开发也是受 B 站的一位 up 主启发，也是他提供了优质的课程为我打好脚本开发的基础。于是在此对这些互联网上分享知识的技术人献上诚挚的感谢！

我在系统设计之初，是对这个系统信心满满的，因为我和 Python 也不是第一天打交道了，之前也有用 Python 做过爬虫。但是在开发过程中，系统的复杂程度远比我想像要高得多，因为是网络安全的攻击系统，所以靶场环境也需要自己搭建，这无疑给自己增添很多工作量。依然记得为了搭建基于 Tomcat 和 MySQL 的网站时，查遍所有资料，无数次恢复虚拟机快照，仍不气馁，矢志不渝的钻研下去。我为自己有这份钻研精神感到自豪，也非常感激永不言弃的自己！

