网络攻防课程项目1 网络攻防扫描工具集成平台

扫描器选择

Nmap

Nmap（Network Mapper）是一个开源的网络扫描和安全审计工具。它被广泛用于网络发现、主机探测和安全评估。

Nmap的主要特点:

主机发现:

Nmap 可以快速检查网络上的活动主机，识别哪些设备在线。

端口扫描:

能够扫描目标主机的开放端口，以确定哪些服务在运行。支持多种扫描技术，如 TCP SYN 扫描、UDP 扫描等。

服务和版本检测:

Nmap 可以识别开放端口所对应的服务类型及其版本信息，帮助用户了解网络环境。

操作系统检测：

Nmap 能够通过分析响应包来推测目标主机的操作系统类型及其版本。

脚本引擎：

Nmap 包含一个强大的脚本引擎（NSE），用户可以编写或使用现成的脚本来执行更复杂的任务，例如漏洞检测和信息收集。

图形用户界面：

虽然 Nmap 主要通过命令行使用，它还提供了一个图形用户界面（Zenmap），便于新手用户使用。

Nikto

Nikto 是一个开源的网页服务器扫描器，用于发现潜在的安全漏洞和配置错误。它能够检测多种类型的漏洞，包括过时的软件版本、常见的安全问题以及服务器配置问题。

全面扫描：

- Nikto 能够扫描服务器的多个方面，包括文件和脚本的漏洞、HTTP 头信息、服务器版本等。

插件支持：

- Nikto 支持自定义插件，可以扩展其功能以适应特定的安全需求。

配置错误检测：

- 检测常见的配置错误，例如不安全的目录权限以及未加密的敏感信息。

扫描结果报告：

- Nikto 提供详细的扫描结果，并支持以多种格式导出报告，包括 HTML、CSV 和 TXT。

更新频率：

- Nikto 的漏洞数据库定期更新，以包含最新的漏洞信息。

其他扫描器

我也寻找了其他可用的扫描器，比如openvas(gvm),metasploit等等，但是由于想开发一个轻量的扫描器，并且缺少linux开发的经验，就选择了nmap和nikto两个比较简单的扫描器进行集成开发

项目开发框架选择

项目选择使用Python的Flask框架开发

源代码

app.py

主程序，定义了扫描器的函数以及处理扫描信息的函数

```
from flask import Flask, render_template, request
import subprocess

app = Flask(__name__)

@app.route('/')
def index():
      return render_template('index.html')

@app.route('/scan', methods=['POST'])
def scan():
      scanners = request.form.getlist('scanners')   # 获取多个扫描器
      target = request.form['target']
      reports = []

      for scanner in scanners:
            if scanner == 'nmap':
                  command = f'nmap {target}'
```

```
            elif scanner == 'nikto':
                command = f'nikto -h {target} 5'
            else:
                return "Invalid scanner", 400

            result = subprocess.run(command, shell=True, capture_output=True, text=True)
            report = format_report(scanner, result.stdout)
            reports.append(report)   # 直接使用 HTML 格式的报告

        return render_template('report.html', reports=reports)

def format_report(scanner, output):
        if scanner == 'nmap':
            return format_nmap_report(output)
        elif scanner == 'nikto':
            return format_nikto_report(output)
        else:
            return "未知扫描器的输出"

def format_nmap_report(output):
        lines = output.splitlines()
        report = "<h3>Nmap 扫描报告</h3>"

        # 提取关键信息
        host_info = ""
        open_ports = []

        for line in lines:
            if line.startswith("Nmap scan report for"):
                host_info = line
            elif "open" in line:
                open_ports.append(line.strip().split())  # 按空格分割，便于后续表格显示

        # 添加主机信息
        report += f"<p><strong>主机信息:</strong> {host_info}</p>"

        # 添加开放端口信息为表格
        if open_ports:
            report += "<h4>开放端口:</h4>"
```

```
58            report += "<table border='1' cellpadding='5' cellspacing='0' style='border-collapse: collapse;'>"
59            report += "<tr><th>端口</th><th>状态</th><th>服务</th></tr>"
60            for port_info in open_ports:
61                  port = port_info[0]   # 端口
62                  state = port_info[1]   # 状态
63                  service = port_info[2] if len(port_info) > 2 else "未知"   # 服务
64                  report += f"<tr><td>{port}</td><td>{state}</td><td>{service}</td></tr>"
65            report += "</table>"
66      else:
67            report += "<p>没有开放的端口。</p>"
68
69      return report
70
71 def format_nikto_report(output):
72      lines = output.splitlines()
73      report = "<h3>Nikto 扫描报告</h3>"
74
75      # 提取关键信息
76      vulnerabilities = []
77
78      for line in lines:
79            if line.strip():   # 只处理非空行
80                  # 检查特定关键字以提取漏洞信息
81                  if "created without the httponly flag" in line:
82                        vulnerabilities.append("未设置httponly的Cookie: " + line)
83                  elif "not present" in line:
84                        vulnerabilities.append("缺少安全头: " + line)
85                  elif "found" in line:
86                        vulnerabilities.append("发现不常见的HTTP头: " + line)
87                  elif "contains" in line:
88                        vulnerabilities.append("robots.txt条目: " + line)
89                  elif "returned a non-forbidden" in line:
90                        vulnerabilities.append("robots.txt中返回非禁止访问的路径: " + line)
91
```

```
        # 添加漏洞信息为表格
        if vulnerabilities:
            report += "<h4>发现的漏洞:</h4>"
            report += "<table border='1' cellpadding='5' cellspacing='0' style='border-collapse: collapse;'>"
            report += "<tr><th>漏洞描述</th></tr>"
            for vuln in vulnerabilities:
                report += f"<tr><td>{vuln}</td></tr>"
            report += "</table>"
        else:
            report += "<p>没有发现漏洞。</p>"

        return report


if __name__ == '__main__':
        app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
    <title>扫描器选择</title>
</head>
<body>
    <div class="container">
        <h1>选择扫描器</h1>
        <form action="{{ url_for('scan') }}" method="post">
            <label for="target">目标网址：</label>
            <input type="text" id="target" name="target" required placeholder="例如: example.com">

            <fieldset>
                <legend>选择扫描器：</legend>
                <div class="checkbox-group">
```

```
                                <label><input type="checkbox" name="scanners" value="nmap">Nmap</label>
                                <label><input type="checkbox" name="scanners" value="nikto">Nikto</label>
                            </div>
                    </fieldset>

                    <button type="submit">开始扫描</button>
                </form>
        </div>
</body>
</html>
```

扫描器标签页，可以输入IP地址，以及选择扫描器

report.html

```
<!DOCTYPE html>
<html lang="zh">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
        <title>扫描报告</title>
</head>
<body>
        <div class="container">
                <h1>扫描报告</h1>
                {% for report in reports %}
                        <div class="report-content">{{ report|safe }}</div>
                {% endfor %}
                <a href="{{ url_for('index') }}">返回</a>
        </div>
</body>
</html>
```

报告扫描结果的网页

style.css

一些简单的样式，用于美化界面

```css
body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 20px;
}

.container {
      max-width: 800px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);
}

h1 {
      color: #333;
      text-align: center;
      margin-bottom: 20px;
}

h3 {
      color: #007BFF;
      margin-top: 30px;
      border-bottom: 2px solid #007BFF;
      padding-bottom: 5px;
}

h4 {
      color: #555;
      margin-top: 20px;
}

form {
      margin: 20px 0;
}

label {
      display: block;
```

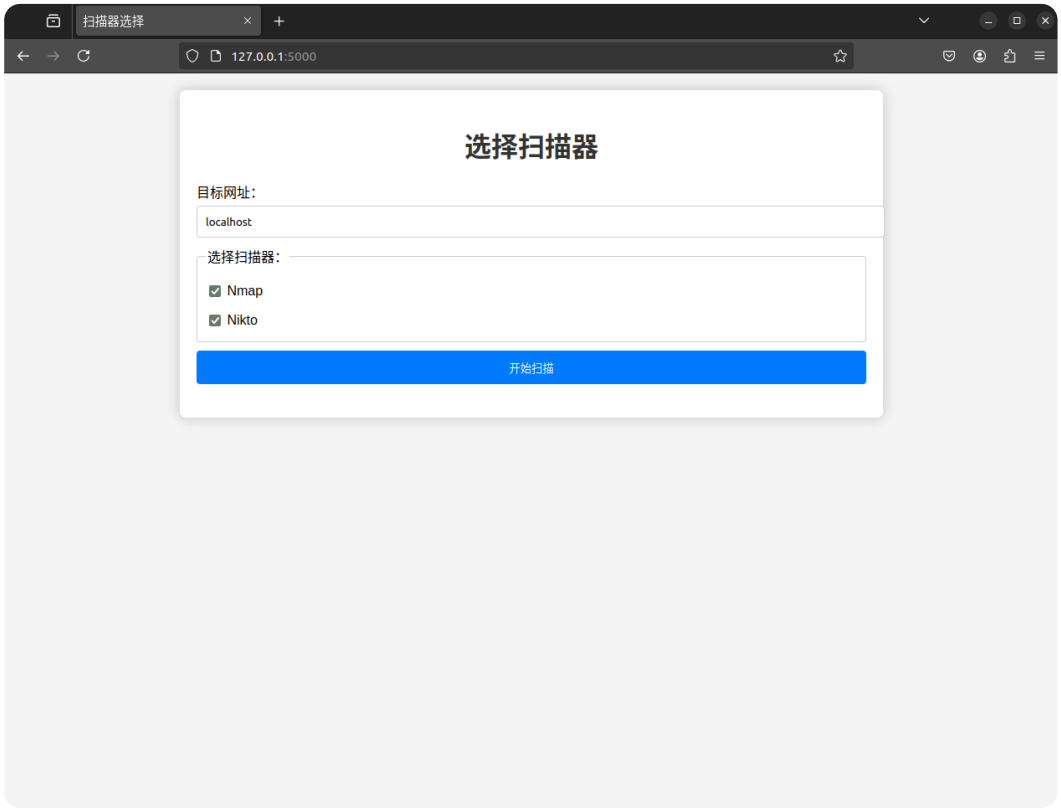
```
        margin: 10px 0 5px;
}

input[type="text"] {
        padding: 10px;
        width: 100%;
        border: 1px solid #ccc;
        border-radius: 4px;
        transition: border-color 0.3s;
}

input[type="text"]:focus {
        border-color: #007BFF;
        outline: none;
}

fieldset {
        border: 1px solid #ccc;
        border-radius: 4px;
        padding: 10px;
        margin: 10px 0;
}

.checkbox-group {
        display: flex;
        flex-direction: column;
}

.checkbox-group label {
        margin-bottom: 5px;
}

button {
        padding: 10px 15px;
        background-color: #007BFF;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        width: 100%;
        transition: background-color 0.3s;
}

button:hover {
        background-color: #0056b3;
}
```

```
87
88  .report-content {
89        background-color: #f9f9f9;
90        padding: 15px;
91        border-radius: 4px;
92        overflow-x: auto;
93        white-space: pre-wrap;
94        margin-top: 15px;
95  }
96
97  table {
98        width: 100%;
99        border-collapse: collapse;
100       margin-top: 15px;
101 }
102
103 table, th, td {
104       border: 1px solid #ccc;
105 }
106
107 th {
108       background-color: #007BFF;
109       color: white;
110       padding: 10px;
111       text-align: left;
112 }
113
114 td {
115       padding: 10px;
116       background-color: #f9f9f9;
117 }
118
119 td:hover {
120       background-color: #f1f1f1;
121 }
122
123 @media (max-width: 600px) {
124       .container {
125             padding: 15px;
126       }
127 }
```

项目测试

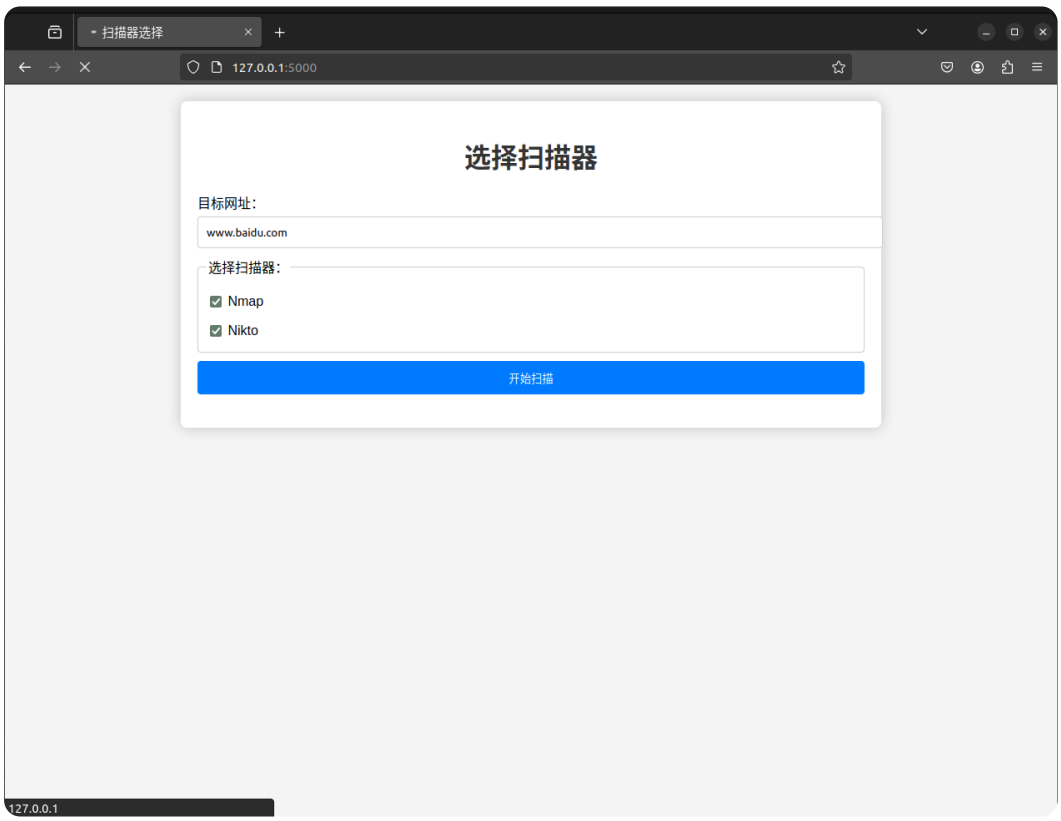
使用应用扫描

扫描本机



扫描结果

扫描百度


扫描器选择
127.0.0.1:5000
选择扫描器
目标网址：
www.baidu.com
选择扫描器：
Nmap
Nikto
开始扫描
127.0.0.1

扫描结果

扫描报告

Nmap 扫描报告

主机信息: Nmap scan report for www.baidu.com (182.61.200.7)

开放端口:

端口	状态	服务
25/tcp	open	smtp
80/tcp	open	http
110/tcp	open	pop3
443/tcp	open	https

Nikto 扫描报告

发现的漏洞:

漏洞描述
未设置httponly的Cookie: + Cookie BAIDUID created without the httponly flag
未设置httponly的Cookie: + Cookie BIDUPSID created without the httponly flag
未设置httponly的Cookie: + Cookie PSTM created without the httponly flag
缺少安全头: + The anti-clickjacking X-Frame-Options header is not present.
发现不常见的HTTP头: + Uncommon header 'x-ua-compatible' found, with contents: IE=Edge,chrome=1
发现不常见的HTTP头: + Uncommon header 'x-xss-protection' found, with contents: 1;mode=block
发现不常见的HTTP头: + Uncommon header 'traceid' found, with contents: 173035717526991582827957302681872485730
发现不常见的HTTP头: + No CGI Directories found (use '-C all' to force check all possible dirs)

缺少安全头: + The anti-clickjacking X-Frame-Options header is not present.
发现不常见的HTTP头: + Uncommon header 'x-ua-compatible' found, with contents: IE=Edge,chrome=1
发现不常见的HTTP头: + Uncommon header 'x-xss-protection' found, with contents: 1;mode=block
发现不常见的HTTP头: + Uncommon header 'traceid' found, with contents: 173035717526991582827957302681872485730
发现不常见的HTTP头: + No CGI Directories found (use '-C all' to force check all possible dirs)
robots.txt条目: + /crossdomain.xml contains 2 lines which should be manually viewed for improper domains or wildcards.
发现不常见的HTTP头: + Server leaks inodes via ETags, header found with file /robots.txt, fields: 0xafe 0x59b382b2ce270
robots.txt中返回非禁止访问的路径: + File/dir '/s?/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
robots.txt中返回非禁止访问的路径: + File/dir '/home/news/data/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
发现不常见的HTTP头: + Uncommon header 'tracecode' found, with contents: 27788183802392204042103114
robots.txt中返回非禁止访问的路径: + File/dir '/bh/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
robots.txt中返回非禁止访问的路径: + File/dir '/shifen/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
robots.txt中返回非禁止访问的路径: + File/dir '/homepage/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
未设置httponly的Cookie: + Cookie H_PS_PSSID created without the httponly flag
未设置httponly的Cookie: + Cookie BDSVRTM created without the httponly flag
发现不常见的HTTP头: + Uncommon header 'bdpagetype' found, with contents: 3
发现不常见的HTTP头: + Uncommon header 'bdqid' found, with contents: 0xcbdd6afa00010d39
robots.txt中返回非禁止访问的路径: + File/dir '/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
robots.txt条目: + "robots.txt" contains 133 entries which should be manually viewed.

项目打包

使用docker进行项目打包

Dockerfile

```
FROM ubuntu:24.04

ENV DEBIAN_FRONTEND=noninteractive

# 更新包列表并安装必要的工具
RUN apt-get update && \
```

```
7       apt-get install -y python3 python3-pip python3-venv
  nmap nikto && \
8       apt-get clean && \
9       rm -rf /var/lib/apt/lists/*
10
11  # 设置工作目录
12  WORKDIR /app
13
14  # 复制依赖文件
15  COPY requirements.txt .
16
17  # 创建虚拟环境并安装依赖
18  RUN python3 -m venv venv && \
19      . venv/bin/activate && \
20      pip install --no-cache-dir -r requirements.txt
21
22  # 复制应用代码
23  COPY . .
24
25  # 暴露应用的端口
26  EXPOSE 5000
27
28  # 设置环境变量以确保Flask使用正确的主机和端口
29  ENV FLASK_APP=app.py
30  ENV FLASK_RUN_HOST=0.0.0.0
31
32  # 运行Flask应用
33  CMD ["venv/bin/python", "-m", "flask", "run"]
34
```

使用Ubuntu作为基础镜像，安装扫描工具和创建虚拟环境，安装python依赖，最后执行应用

项目上传

本项目已经上传到dockerhub

欢迎下载试用

网址