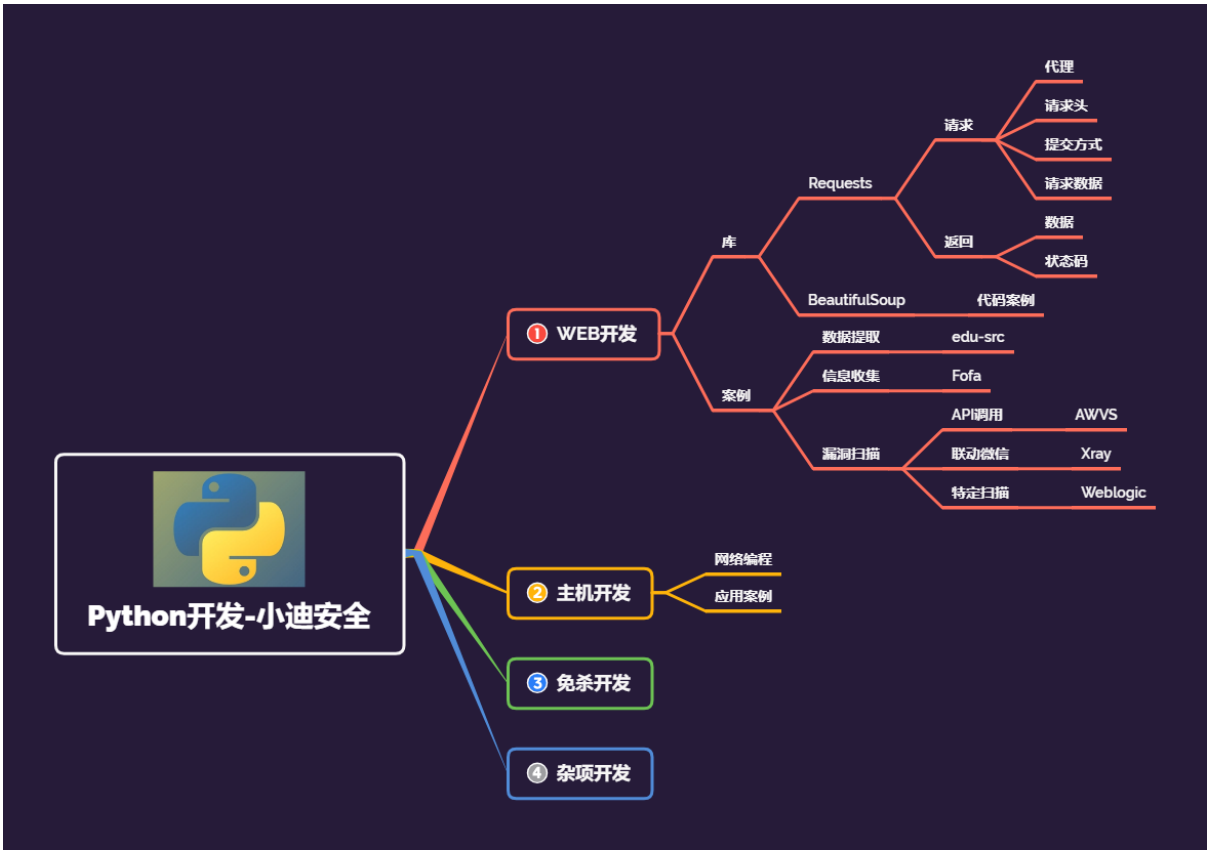
安全开发-Python-红队项目&漏扫调用&API 推送微信&任务自动添加并启动



#知识点：

1、Python-应用方向-红队项目

2、Python-Xray&Awvs&SQLMAP

3、Python-推送微信&自动添加&启动

#章节点：

Web 爬虫解析类

多线程异步处理类

Socket 网络通讯类

其他第三方库数据类

#应用点：

信息打点，POC&EXP 利用，工具 API 调用，文件流量监控，工具脚本开发,远控后门等

演示案例：

- Python-红队项目-Xray 调用推送微信
- Python-红队项目-Awvs 调用自动添加
- Python-红队项目-SQLMAP 调用自动添加

#Python-红队项目-Xray 调用推送微信

漏扫 API 调用-Xray

参考：https://docs.xray.cool/#/webhook/webhook

应用案例：可通过自动化扫描后将实时结果进行微信推送,也可以应用在其他安全工具上。

```
xray webscan --url http://x.x.x.x --webhook-output
http://127.0.0.1:5000/webhook
```

1、命令漏扫触发本地 URL

2、Flask 启动进行监听处理

3、借助 Server 酱 API 推送微信

#Python-红队项目-Awvs 调用自动添加

漏扫 API 调用-AWVS

参考：https://blog.csdn.net/wy_97/article/details/106872773

应用案例：可通过脚本调用 AWVS 自动添加扫描,也可以应用在其他安全工具上。

1、启动工具&开启 API-KEY

2、创建新任务并记录任务 ID

3、启动新任务并记录返回 ID

#Python-红队项目-SQLMAP 调用自动添加

漏扫 API 调用-SQLMAP

参考：https://www.freebuf.com/articles/web/204875.html

应用案例：前期通过信息收集拿到大量的 URL 地址，这个时候可以配合 SqlmapAPI 接口进行批量的注入检测。

开发当前项目过程:（利用 sqlmapapi 接口实现批量 URL 注入安全检测）

0.启用 sqlmap-API 服务 python sqlmapapi.py -s

1.创建新任务记录任务 ID @get("/task/new")

2.设置任务 ID 扫描信息 @post("/option/<taskid>/set ")

3.开始扫描对应 ID 任务 @post("/scan/<taskid>/start")

4.读取扫描状态判断结果 @get("/scan/<taskid>/status")

5.如果结束删除 ID 并获取结果 @get("/task/<taskid>/delete")

6.扫描结果查看 @get("/scan/<taskid>/data")

```
from flask import Flask, request
import requests

app = Flask(__name__)

@app.route('/webhook', methods=['POST'])
def xray_webhook():
    url = 'https://sctapi.ftqq.com/你的KEY.send?title=Xray find
vuln!!!'
    try:
        #接受传递过来的数据转换json格式
        vuln=request.json
        content = """## xray 发现了新漏洞
        url: {url}
        插件: {plugin}
        漏洞类型: {vuln_class}
        请及时查看和处理
        """.format(url=vuln['data']['target']['url'],
plugin=vuln['data']['plugin'],vuln_class=vuln['type'])
        print(content)
        data={
            'desp':content
        }
        print(data)
        requests.post(url,data=data)
        return 'ok'
    except Exception as e:
        pass


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

```
import requests

def new_id(key,url):
    api_add_url = "https://localhost:3443/api/v1/targets"
    headers = {
        'X-Auth': key,
        'Content-type': 'application/json'
    }
    data =
'{"address":"%s","description":"create_by_reaper","criticality":"
10"}'%url
    r = requests.post(url=api_add_url, headers=headers,
data=data, verify=False).json()
    id=r['target_id']
    if id is not None:
        print(url+'->任务添加成功，等待启动...')
        print(url + '->任务 ID->'+id)
        return id

def start_id(key,id):
    # 核心代码
    url = 'https://localhost:3443/api/v1/scans'
    headers = {
        'X-Auth': key,
        'Content-type': 'application/json'
    }
    data = '{"profile_id":"11111111-1111-1111-1111-
111111111111","schedule":{"disable":false,"start_date":null,"time
_sensitive":false},"target_id":"%s"}' % id
    r = requests.post(url=url, headers=headers, data=data,
verify=False).json()
    if r['scan_id'] is not None:
        print('任务 ID->'+r['target_id']+"->任务启动成功，等待完
毕...")

if __name__ == '__main__':
    key='你的 key'
    for url in open('url.txt'):
        id=new_id(key,url.replace('\n',''))
        start_id(key,id)
```

```
import requests,time,json

#0.启用 sqlmap-API 服务  python sqlmapapi.py -s
#1.创建新任务记录任务 ID    @get("/task/new")
#2.设置任务 ID 扫描信息    @post("/option/<taskid>/set    ")
#3.开始扫描对应 ID 任务    @post("/scan/<taskid>/start")
#4.读取扫描状态判断结果    @get("/scan/<taskid>/status")
#5.如果结束删除 ID 并获取结果    @get("/task/<taskid>/delete")
#6.扫描结果查看 @get("/scan/<taskid>/data")

def new_id():
    headers = {
        'Content-Type': 'application/json'
    }
    url='http://127.0.0.1:8775'+'/task/new'
    resp=requests.get(url,headers=headers).json()
    taskid=resp['taskid']
    if resp['success'] is True:
        print('->1、创建任务 ID 成功，ID:' + taskid)
        return taskid

def set_id(id,scanurl):
    headers = {
        'Content-Type': 'application/json'
    }
    data={
        'url':scanurl
    }
    url = 'http://127.0.0.1:8775/option/%s/set' % id
    resp = requests.post(url,
data=json.dumps(data),headers=headers).json()
    if resp['success'] is True:
        print('->2、设置任务 ID 成功，ID:' + taskid)
        print('->2、设置任务 URL 成功，URL:' + scanurl)
        return taskid

def scan_id(id,scanurl):
    headers = {
        'Content-Type': 'application/json'
    }
    data = {
        'url': scanurl
```

涉及资源：

补充：涉及录像课件资源软件包资料等下载地址