这是作者的系列网络安全自学教程，主要是关于网安工具和实践操作的在线笔记，特分享出来与博友共勉，希望您们喜欢，一起进步。前文分享了Python弱口令攻击、自定义字典生成，调用Python的exrex库实现，并结合Selenium和BurpSuite实现网站暴库案例；本文将分析Python攻防之构建Web目录扫描器，实现IP代理池。本文参考了爱春秋ADO老师的课程内容，这里也推荐大家观看他Bilibili和ichunqiu的课程，同时也结合了作者之前的编程经验进行讲解。

作者作为网络安全的小白，分享一些自学基础教程给大家，希望你们喜欢。同时，更希望你能与我一起操作深入进步，后续也将深入学习网络安全和系统安全知识并分享相关实验。总之，希望该系列文章对博友有所帮助，写文不容易，大神请飘过，不喜勿喷，谢谢！

下载地址：https://github.com/eastmountyxz/NetworkSecuritySelf-study
百度网盘：https://pan.baidu.com/s/1dsunH8EmOB_tlHYXXguOeA 提取码：izeb

文章目录

前文学习：

[网络安全自学篇] 一.入门笔记之看雪Web安全学习及异或解密示例
[网络安全自学篇] 二.Chrome浏览器保留密码功能渗透解析及登录加密入门笔记
[网络安全自学篇] 三.Burp Suite工具安装配置、Proxy基础用法及暴库示例
[网络安全自学篇] 四.实验吧CTF实战之WEB渗透和隐写术解密
[网络安全自学篇] 五.IDA Pro反汇编工具初识及逆向工程解密实战
[网络安全自学篇] 六.OllyDbg动态分析工具基础用法及Crakeme逆向破解
[网络安全自学篇] 七.快手视频下载之Chrome浏览器Network分析及Python爬虫探讨
[网络安全自学篇] 八.Web漏洞及端口扫描之Nmap、ThreatScan和DirBuster工具
[网络安全自学篇] 九.社会工程学之基础概念、IP获取、IP物理定位、文件属性
[网络安全自学篇] 十.论文之基于机器学习算法的主机恶意代码
[网络安全自学篇] 十一.虚拟机VMware+Kali安装入门及Sqlmap基本用法
[网络安全自学篇] 十二.Wireshark安装入门及抓取网站用户名密码（一）
[网络安全自学篇] 十三.Wireshark抓包原理（ARP劫持、MAC泛洪）及数据流追踪和图像抓取（二）
[网络安全自学篇] 十四.Python攻防之基础常识、正则表达式、Web编程和套接字通信（一）
[网络安全自学篇] 十五.Python攻防之多线程、C段扫描和数据库编程（二）
[网络安全自学篇] 十六.Python攻防之弱口令、自定义字典生成及网站暴库防护

前文欣赏：

[渗透&攻防] 一.从数据库原理学习网络攻防及防止SQL注入
[渗透&攻防] 二.SQL MAP工具从零解读数据库及基础用法
[渗透&攻防] 三.数据库之差异备份及Caidao利器
[渗透&攻防] 四.详解MySQL数据库攻防及Fiddler神器分析数据包

参考文献：

《安全之路Web渗透技术及实战案例解析》陈小兵老师
《Wireshark数据包分析实战》第二版 Chris Sanders
《TCP/IP协议栈详解卷一》 W.Richard Stevens

《Wireshark协议分析从入门到精通》-51cto老师
https://www.bilibili.com/video/av29479068
2019 Python黑客编程：安全工具开发 - bilibili 白帽黑客教程
Dirmap：一款高级Web目录文件扫描工具 - Freebuf 大神H4ckForJob
网站目录扫描工具 - CSDN谢公子大佬
Python黑客工具简述 - freebuf
Python打造一个目录扫描工具 - 博客园sch01ar大神

> 声明：本人坚决反对利用教学方法进行犯罪的行为，一切犯罪行为必将受到严惩，绿色网络需要我们共同维护，更推荐大家了解它们背后的原理，更好地进行防护。

一.Web目录扫描思路

1.网站目录和敏感文件扫描

网站目录和敏感文件扫描是网站测试中最基本的手段之一。如果通过该方法发现了网站后台，可以尝试暴库、SQL注入等方式进行安全测试；如果发现敏感目录或敏感文件，能帮我们获取如php环境变量、robots.txt、网站指纹等信息；如果扫描出了一些上传的文件，我们甚至可能通过上传功能（一句话恶意代码）获取网站的权限。

2.原理

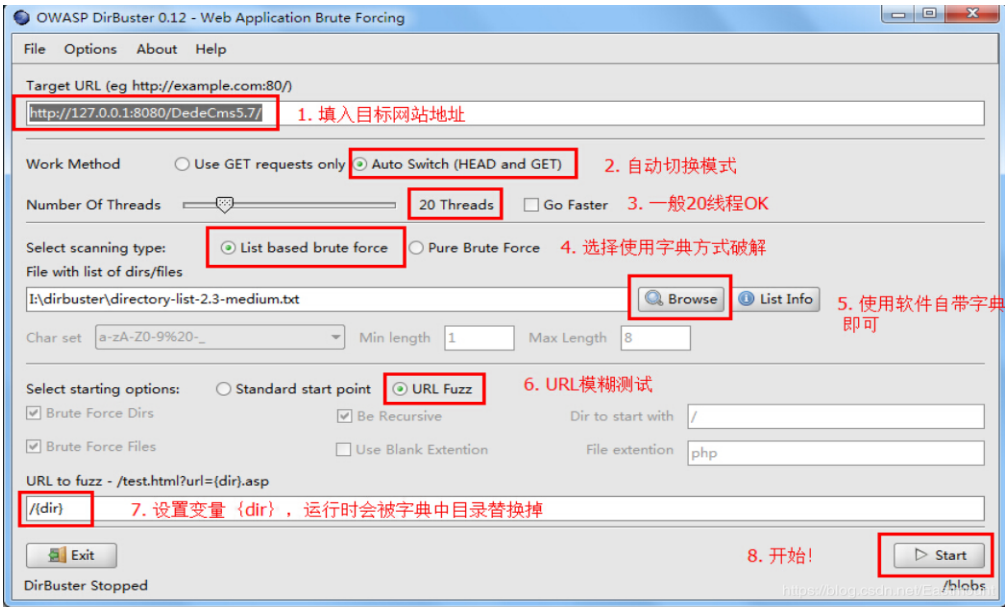
在Web目录扫描中，字典是非常重要的，一个好的字典能帮助我们的程序更好地发现漏洞和目标。那么，如何通过Python代码实现Web目录扫描呢？或者Web目录扫描器的原理是什么呢?

其原理是通过请求返回的信息来判断当前目录或文件是否真实存在。网站后台扫描工具都是利用目录字典进行爆破扫描，字典越多，扫描到的结果也越多。常见的Web目录扫描工具包括：御剑1.5、DirBuster、Dirsearch、Webdirscan、Cansina、Dirmap等。涉及的常用功能包括：能使用字典、支持纯爆破、并发引擎、能爬取页面动态生成字典、能fuzz扫描、能自定义请求（代理）、自定义响应结果及响应状态等。

3.工具介绍

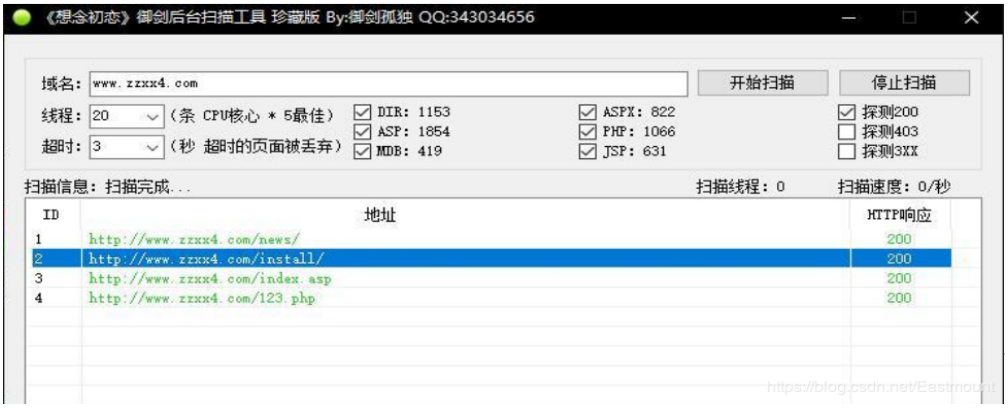
DirBuster

Kali Linux提供的目录扫描工具DirBuster支持全部的Web目录扫描方式。它既支持网页爬虫方式扫描，也支持基于字典暴力扫描，还支持纯暴力扫描。该工具使用Java语言编写，提供命令行（Headless）和图形界面（GUI）两种模式。其中，图形界面模式功能更为强大。用户不仅可以指定纯暴力扫描的字符规则，还可以设置以URL模糊方式构建网页路径。同时，用户还对网页解析方式进行各种定制，提高网址解析效率。



御剑

御剑系列的web工具一直是比较顺手的工具。这款御剑也是很好用的网站后台扫描工具，图形化页面，使用起来简单上手，因此也被大多数人所喜好。其作者可能是“御剑孤独”。

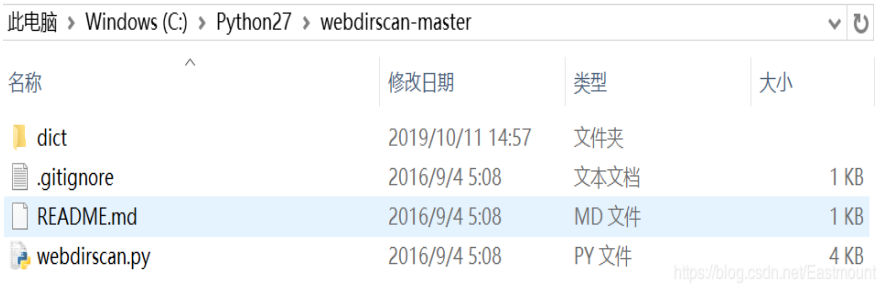


Webdirscan

webdirscan是一个很简单的多线程Web目录扫描工具，它是使用Python语言编写的，主要调用了requests第三方库实现。大家可以看看它Github上面的代码，和本篇博客原理较为相似。

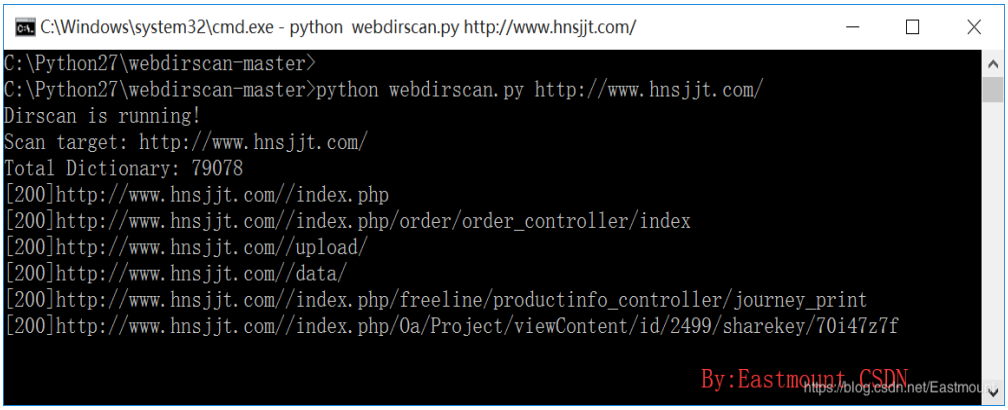
源代码：https://github.com/TuuuNya/webdirscan/

我们将代码下载至本地，再进行扫描目标网站。



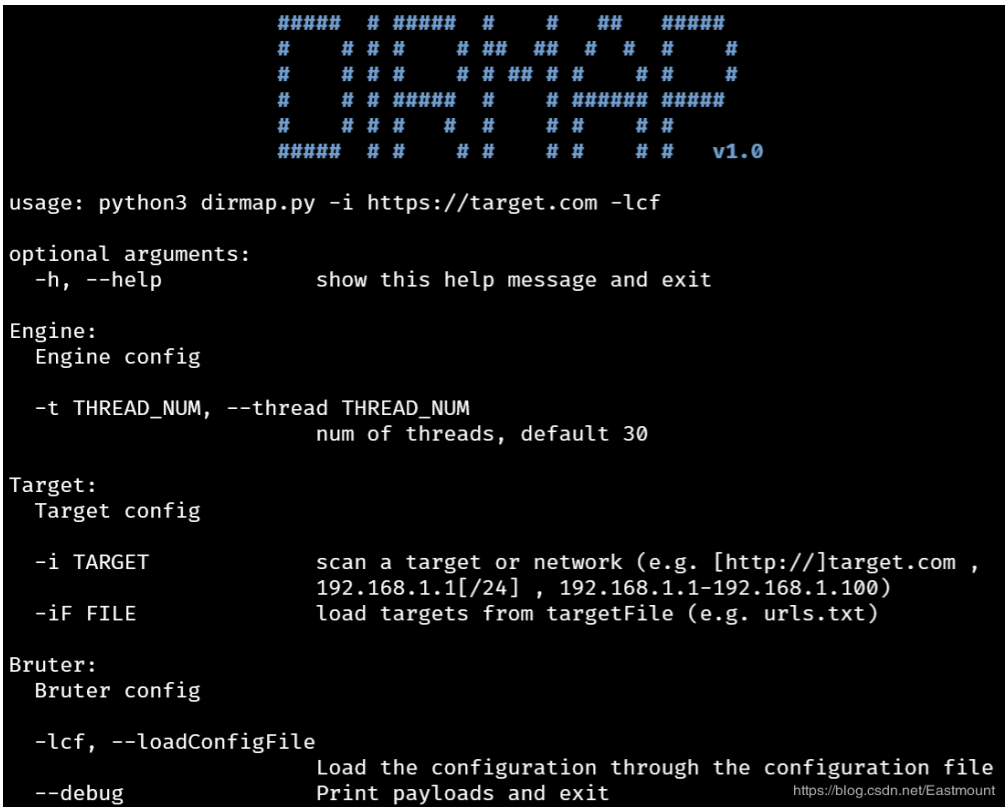
将CMD命令行打开，进入webdirscan路径下，指定扫描任务。

```
python webdirscan.py 目标网站
```



Dirmap

它是一个高级web目录扫描工具，功能将会强于DirBuster、Dirsearch、cansina、御剑。详见：https://github.com/H4ckForJob/dirmap



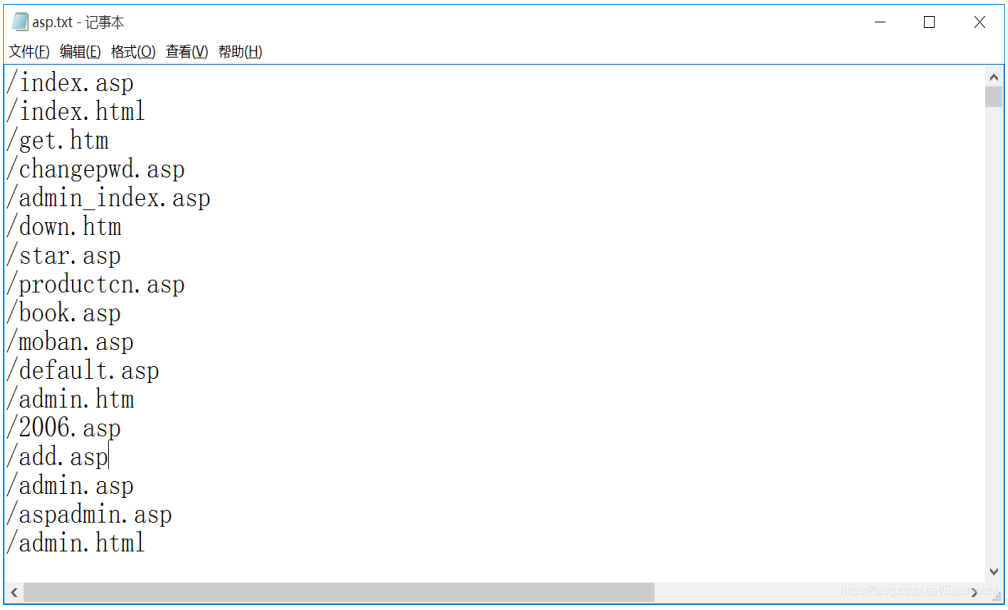
注意：工具的使用方法这里就不进行详细介绍了，希望读者下来自行学习，本文主要分享Python代码是如何实现Web目录扫描的。

二.Python构建Web目录扫描器

该程序主要实现以下3个功能：

- 判断Web目录或文件是否存在。通过requests发送请求实现，获取status_code状态码，状态码200表示成功。
- 通过读取文件后去 asp、aspx、jsp、php 常见目录，对其进行扫描。
- 由于很多安全产品能识别出你的恶意攻击请求，这里需要设置多线程调用，从而避免安全软件识别。

下面是Python实现Web目录扫描的代码，其中本地存在一个 asp.txt 文件（源自御剑），涉及了常见的网站目录。如下图所示：



完整代码：

```
# -*-  coding: utf-8 -*-
import requests
from Queue import Queue
import sys
import threading

#多线程实现Web目录扫描
class DirScan(threading.Thread):

    def __init__(self, queue):
        threading.Thread.__init__(self)
```

```
            self._queue = queue

        def run(self):
            #获取队列中的URL
            while not self._queue.empty():
                url = self._queue.get()
                #print url

                try:
                    headers = {
                        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) A
                    }
                    #发送请求
                    r = requests.get(url=url, headers=headers, timeout=8)

                    #Web目录存在
                    if r.status_code == 200:
                        #print '[*] ' + url
                        sys.stdout.write('[*] %s\n' % url)
                except Exception, e:
                    #print e
                    pass

    #定义队列及放入URL
    def start(url, ext, count):
        queue = Queue()

        f = open('%s.txt' % ext, 'r')
        for i in f:
            queue.put(url + i.rstrip('\n'))

        #多线程
        threads = []
        thread_count = int(count)

        for i in range(thread_count):
            threads.append(DirScan(queue))

        for t in threads:
            t.start()

        for t in threads:
            t.join()

    #主函数
    if __name__ == '__main__':
        url = 'http://www.xxxxx.com'
```

```
ext = 'asp'
count = 10
start(url, ext, count)
```

作者通过浏览器搜索 “inurl:asp”，寻找某网站为例，接着调用程序获取它的目录。

其扫描结果如下图所示，通过访问这些链接发现它们是真实存在的。

写到这里，一个简单的Web目录扫描器就实现了，希望对大家有所帮助 。后续如果将我们的程序扩展到BurpSuite工具，就能更好地进行抓包分析及安全测试，你可以去试试~

三.ip代理池

某些网站会对我们发送的请求进行有效拦截，这里可以尝试设置一个ip代理池，无论是网络爬虫还是请求发送，都能很好地解决这些问题。下面简单讲解一个获取IP代理的代

码，但遗憾的是，作者想把它移植到上面那段代码中，但验证的IP地址多数无法访问，导致失败。

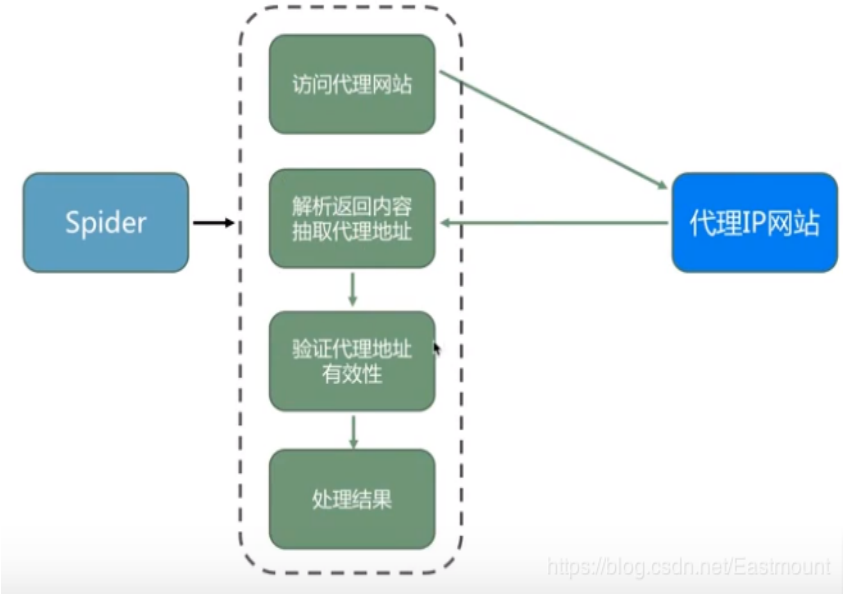
国内IP代理网站为：https://www.xicidaili.com/nn

https://www.xicidaili.com/nn

公告：本站所有代理IP地址均收集整理自国内公开互联网，本站不维护运营任何代理服务器，请自行筛选。

国家	IP地址	端口	服务器地址	是否匿名	类型	速度	连接时间	存活时间	验证时间
	114.239.144.71	808	江苏宿迁市泗阳县	高匿	HTTP			842天	19-10-11 17:21
	123.163.122.85	9999	河南洛阳	高匿	HTTP			1分钟	19-10-11 17:20
	175.43.58.29	9999	福建泉州	高匿	HTTP			1分钟	19-10-11 17:20
	106.110.200.238	9999	广东	高匿	HTTPS			1分钟	19-10-11 17:00
	222.189.144.246	9999	江苏扬州	高匿	HTTP			73天	19-10-11 17:00
	120.83.108.188	9999	广东揭阳市普宁	高匿	HTTPS			109天	19-10-11 17:00
	119.23.150.171	8118		高匿	HTTPS			1分钟	19-10-11 17:00
	123.163.122.203	9999	河南洛阳	高匿	HTTPS			21天	19-10-11 17:00
	27.152.91.83	9999	福建泉州	高匿	HTTP			1分钟	19-10-11 16:21
	218.75.69.50	39590	浙江杭州	高匿	HTTPS			374天	19-10-11 16:21
	120.83.111.218	9999	广东揭阳市普宁	高匿	HTTP			1分钟	19-10-11 16:20
	113.120.36.20	9999	山东济南	高匿	HTTP			1分钟	19-10-11 16:20
	163.204.245.254	9999	广东	高匿	HTTP			60天	19-10-11 16:20
	123.163.97.37	9999	河南洛阳	高匿	HTTP			152天	19-10-11 16:20
	113.194.30.119	9999	江西	高匿	HTTPS			1分钟	19-10-11 16:00

其基本思路如下，通过Python爬虫获取IP地址、端口和协议类型，其代码的基本思路如下：



下面是对应的HTML源代码，需要抓取的是tr值，每行代表一个IP地址。

完整代码：

```
# -*- coding:utf-8 -*-
import requests
import re
from bs4 import BeautifulSoup as bs
import telnetlib

#爬取数据
def proxy_spider():
    #设置请求
    headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
    url = 'https://www.xicidaili.com/nn'
    r = requests.get(url=url, headers=headers)
    print r

    #解析 通过re.compile('|[^odd]')解析奇数和偶数行
    soup = bs(r.content, 'lxml')
    datas = soup.find_all(name='tr', attrs={'class': re.compile('|[^odd]

    for data in datas:
        soup_proxy_content = bs(str(data), 'lxml')
        soup_proxys = soup_proxy_content.find_all(name='td')
        ip = str(soup_proxys[1].string)
        port = str(soup_proxys[2].string)
        types = str(soup_proxys[5].string)
        #print ip, port, types

        #判断IP地址是否存活
        proxy = {}
        proxy[types.lower()] = '%s:%s' % (ip, port)
        #proxy_check(ip, port, types)
```

```
            proxy_telnet(ip, port, types)

    #获取能成功使用的代理ip内容 调用requests代理访问方法
    def proxy_check(ip, port, types):
        proxy = {}
        proxy[types.lower()] = '%s:%s' % (ip, port)
        #proxy = {'http':'119.254.84.90:80'}
        try:
            r = requests.get('http://1212.ip138.com/ic.asp', proxies=proxy, t
            #print r.text
            ip_content = re.findall(r"\[(.*?)\]", r.text)[0]
            if ip == ip_content:
                print proxy
        except Exception, e:
            print e
            pass

    #检测IP地址是否存活
    def proxy_telnet(ip, port, types):
        proxy = {}
        proxy[types.lower()] = '%s:%s' % (ip, port)

        try:
            telnetlib.Telnet(ip, port, timeout=2)
            print 'True:', proxy
        except:
            print 'False:', proxy

    proxy_spider()
```

输出结果如下图所示，IP地址和端口成功抓取，但是很多无法使用，读者可以自行试试。

```
>>>
<Response [200]>
False: {'http': '114.239.144.71:808'}
False: {'http': '123.163.122.85:9999'}
True: {'http': '175.43.58.29:9999'}
False: {'https': '106.110.200.238:9999'}
False: {'http': '222.189.144.246:9999'}
True: {'https': '120.83.108.188:9999'}
True: {'https': '119.23.150.171:8118'}
False: {'https': '123.163.122.203:9999'}
False: {'http': '27.152.91.83:9999'}
False: {'https': '218.75.69.50:39590'}
False: {'http': '120.83.111.218:9999'}
False: {'http': '113.120.36.20:9999'}
False: {'http': '163.204.245.254:9999'}
False: {'http': '123.163.97.37:9999'}
```

获取IP地址之后，通过如下设置可以使用代理IP地址进行访问。

proxy = {‘http’:‘119.254.84.90:80’}

r = requests.get(‘http://www.xxxx.com’, proxies=proxy, timeout=6)

四.总结

希望这篇文章对你有所帮助，这是Python网络攻防非常基础的一篇博客，后续作者也将继续深入学习，制作一些常用的小工具供大家交流。作者B站的视频推荐几乎都是网络安全和Python编程，这个算法写得不错，最近挤空闲的时间看了100多部视频。Python攻防之弱口令、字典暴库还在撰写中，论文汇报的PPT也快100页了，接下来需要学会精简和总结。种一棵树最好的时间是十年前，其次是现在，忙点好，一起加油。

(By:Eastmount 2019-10-10 晚上11点 http://blog.csdn.net/eastmount/)