扫描器之敏感目录爆破

一、实验介绍

1.1 实验内容

通过调用字典访问url通过网页返回的状态来判断是否存在此目录。

1.2 实验知识点

- requests
- threading

1.3 实验环境

- python2.7
- Xfce终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有Python基础的用户，熟悉python基础知识加深巩固。

1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
$ wget http://labfile.oss.aliyuncs.com/courses/761/shiyanlouscan7.zip
```

二、开发准备

在项目目录 data 新建一个 dir.txt ，里面的内容为url目录字典。可以自己创建也可以从 shiyanlouscan7.zip 中获取。

三、实验步骤

3.1 简述

敏感目录爆破，通过字典爆破网站目录结构，可能会得到敏感的目录结构

如果学习过之前的几章，这节课是非常的轻车熟路了。主要就是两个python库 threading requests 的使用。

3.2 装载字典文件

在 lib/core 中创建 webdir.py 文件。

首先将字典文件加入到队列中，设置一些需要初始化的值。

```
def __init__(self,root,threadNum):
    self.root = root
    self.threadNum = threadNum
    self.headers = {
         'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.20 (KHTML, like Gecko) C
hrome/19.0.1036.7 Safari/535.20',
         'Referer': 'http://www.shiyanlou.com',
         'Cookie': 'whoami=w8ay',
         }
    self.task = Queue.Queue()
    self.s_list = []
    filename = os.path.join(sys.path[0], "data", "dir.txt")
    for line in open(filename):
        self.task.put(root + line.strip())
```

3.3 检测网页状态

为了提升检测网站的速度，我们只需要用 head 访问网页头来判断返回的状态码即可：

```
def checkdir(self,url):
    status_code = 0
    try:
        r = requests.head(url,headers=self.headers)
        status_code = r.status_code
    except:
        status_code = 0
    return status_code
```

3.4 线程函数

线程函数主要是从队列中取出数据，然后循环访问。

```
def test_url(self):
    while not self.task.empty():
        url = self.task.get()
        s_code = self.checkdir(url)
        if s_code==200:
            self.s_list.append(url)
        print "Testing: %s status:%s"%(url,s_code)
```

3.5 工作线程

work 函数是调用的主函数，通过 work 函数来启动线程，开始任务。

```
def work(self):
    threads = []
    for i in range(self.threadNum):
        t = threading.Thread(target=self.test_url())
        threads.append(t)
        t.start()
    for t in threads:
        t.join()
    print('[*] The DirScan is complete!')
```

3.6 输出函数

在工作线程 test_url 中我们有

```
 if s_code==200:
    self.s_list.append(url)
```

这样一段代码， s_list 是访问成功得到的列表，输出函数我们输出列表 s_list 即可：

```
def output():
    if len(self.s_list):
        print "[*] status = 200 dir:"
        for url in s_list:
            print url
```

我设定的是状态码为200的时候才会加入，当然大家也可以在工作线程 test_url 设置状态码不等于404的时候加入。

3.7 代码整理

总代码如下：

```
#!/usr/bin/env python
# __author__= 'w8ay'
import os
import sys
import Queue
import requests
import threading

class webdir:
    def __init__(self,root,threadNum):
        self.root = root
        self.threadNum = threadNum
        self.headers = {
             'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/535.20 (KHTML, like Geck
o) Chrome/19.0.1036.7 Safari/535.20',
             'Referer': 'http://www.shiyanlou.com',
             'Cookie': 'whoami=w8ay',
             }
        self.task = Queue.Queue()
        self.s_list = []
        filename = os.path.join(sys.path[0], "data", "dir.txt")
        for line in open(filename):
            self.task.put(root + line.strip())

    def checkdir(self,url):
        status_code = 0
        try:
            r = requests.head(url,headers=self.headers)
            status_code = r.status_code
        except:
            status_code = 0
        return status_code

    def test_url(self):
        while not self.task.empty():
            url = self.task.get()
            s_code = self.checkdir(url)
            if s_code==200:
                self.s_list.append(url)
            print "Testing: %s status:%s"%(url,s_code)

    def work(self):
        threads = []
        for i in range(self.threadNum):
            t = threading.Thread(target=self.test_url())
            threads.append(t)
            t.start()
        for t in threads:
            t.join()
        print('[*] The DirScan is complete!')

    def output():
        if len(self.s_list):
            print "[*] status = 200 dir:"
            for url in s_list:
                print url
```

3.8 集成到扫描器

动手实践是学习 IT 技术最有效的方式！ 开始实验

单个模块我们当然可以单独使用了，不过我们可以通过调用扫描器来使用。
将模块加入扫描程序主文件 `w8ay.py` 中。

```
from lib.core import webcms,PortScan,common,webdir

reload(sys)
sys.setdefaultencoding('utf-8')
def main():
    root = "https://www.shiyanlou.com/"
    threadNum = 10
    #IP Ports Scan
    ip = common.gethostbyname(root)
    print "IP:",ip
    print "START Port Scan:"
    pp = PortScan.PortScan(ip)
    pp.work()

    # DIR Fuzz
    dd = webdir.webdir(root,threadNum)
    dd.work()
    dd.output()

    #webcms
    ww = webcms.webcms(root,threadNum)
    ww.run()

    #spider
    w8 = SpiderMain(root,threadNum)
    w8.craw()

if __name__ == '__main__':
    main()
```


不要忘了 import 我们的 `webdir` 模块

现在。我们的扫描器运行流程是：

1. 域名->转换ip->端口扫描
2. 敏感目录扫描
3. CMS识别
4. 爬虫信息收集 ->基于爬虫的各类模块

四、实验总结

这个版本的敏感目录扫描时最为初级的扫描工具，在实际当中，如果网站有防火墙，waf之类的东西都可以轻易的防住这些扫描，该如何突破呢？方法很多，有兴趣可以自行查询资料。

上一节 (/courses/761/labs/2670/document)
下一节 (/courses/761/labs/2672/document)

课程教师

new4
共发布过1门课程

查看老师的所有课程 > (/teacher/102428)