



警示

1. 实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2. 当次小组成员成绩只计学号、姓名登录在下表中的。
3. 在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4. 实验报告文件以 PDF 格式提交。

院系	数据科学与计算机学院		班 级	1501	组长	曾瑜
学号	15352022	15352005	15352009			
学生	曾瑜	蔡景韬	蔡潇怡			
实验分工						
曾瑜	参与实验，共同完成实验报告		蔡景韬	参与实验，共同完成实验报告		
蔡潇怡	参与实验，共同完成实验报告					

Wifi 热点实验

【实验图标】



【实验内容】

下面实验时，请写明使用的手机品牌、型号。

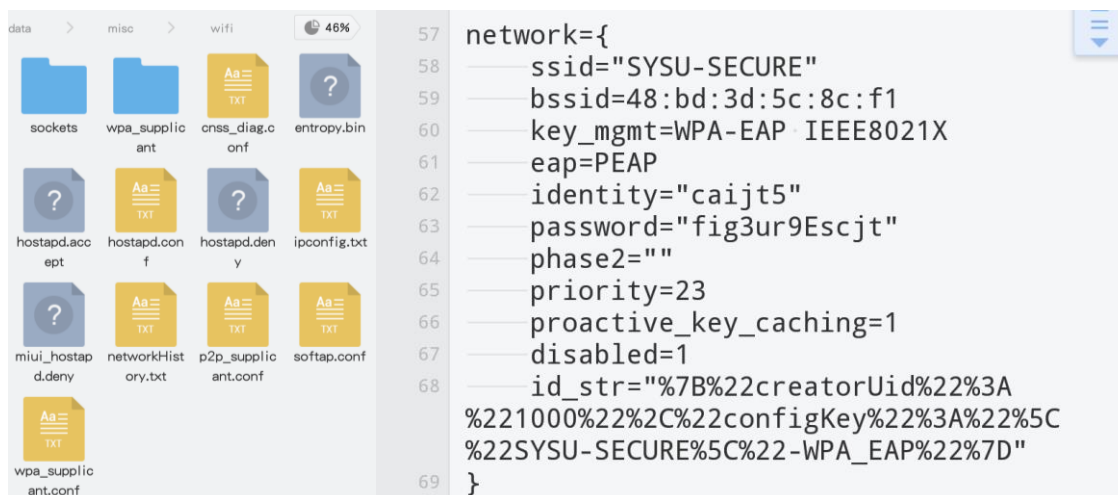
实验 1：简述什么是 WiFi 热点。



WiFi 热点是将手机接收的 GPRS、3G 或 4G 信号转化为 WiFi 信号再发出去，这样手机就成为了一个 WiFi 热点。只有具有无线 AP 功能的手机才能作为 WiFi 热点建立 WiFi 网络。具有 WiFi 功能的设备连接到手机建立的 WiFi 热点后，消耗的流量都是该手机卡的流量。

实验 2：在手机上，会保留搜索到的 wifi 信息，当处在相应 wifi 环境下时能免密自动连接。请给出所保留 wifi 的 SSID、保存密码的文件夹及文件具体路径、内容（如果使用工具软件，简述此工具的功能），请通过一连接实例找出并给出截图（请描述实验时的环境，例如在什么地点、场合搜索到什么 wifi，而该 wifi 需要密码才能连接上网）。

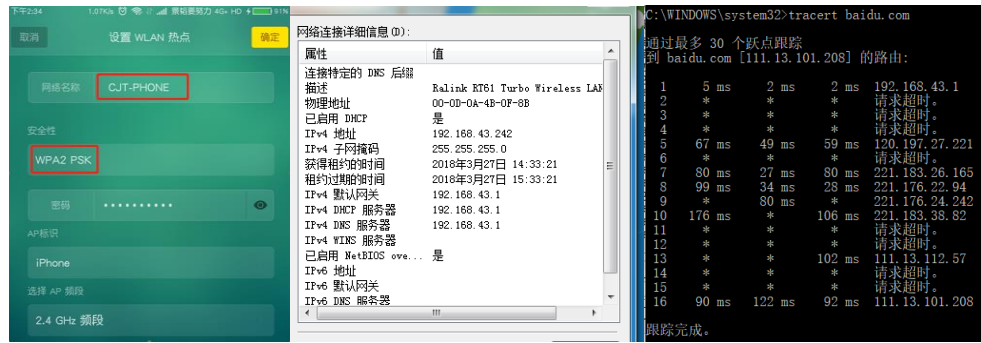
- 1、所保留 wifi 的 SSID、保存密码的文件夹及文件具体路径 data/misc/wifi/wpa_supplicant.conf
- 2、所保留文件内容如下：



实验 3：分别在下列设备上建立 wifi 热点，指出采用什么身份认证，测试 PC 连接到手机热点、手机连接到 PC 热点的连通性。捕获热点的连接信息并加以分析。

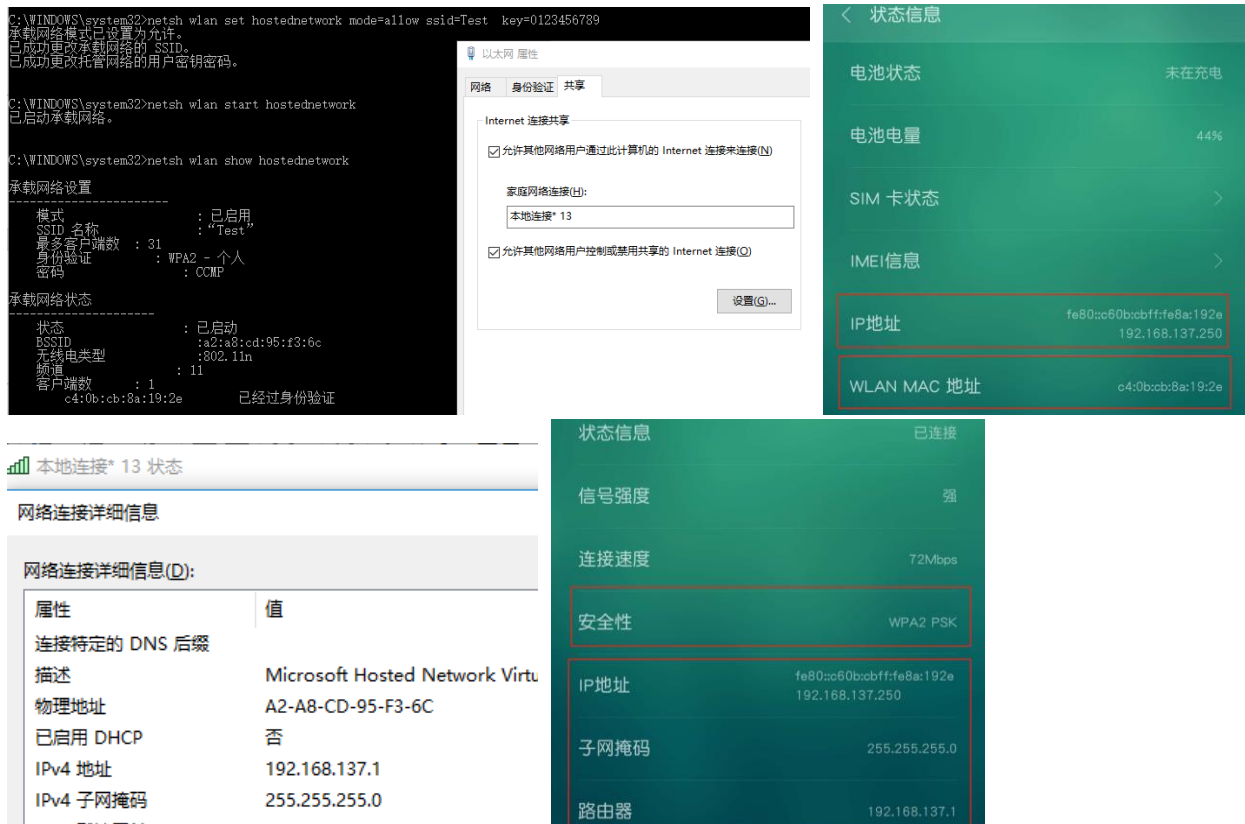
(1) 直接在手机建立。

- 1、在手机上建立热点，由图一可见该手机 wifi 热点使用 WPA2-PSK 进行身份认证
- 2、由图二可看出，PC 端连接上手机热点后，其通过 DHCP 获取的 ip 地址为 192.168.43.242
- 3、通过查阅手机的状态信息，可以看到其 ip 地址为 10.140.38.142，且每次重新断开并开启数据连接，其 ip 地址都会改变，该 ip 地址为中国移动网分配的公网 ip
- 4、使用 tracert 追踪 ping 百度网站的跃点情况如图三
- 5、综上可以推测，手机芯片中集成的网卡，可以发送 2.4G 频段的 AP 信号，并使用 DHCP 服务分配局域网 IP，最终客户端连接该热点，通过手机访问公网



(2) 在有无线网卡的 PC 上建立。

- 1、设置 PC wifi 热点: `netsh wlan set hostednetwork mode=allow ssid=Test key=0123456789`
- 2、开启 wifi 热点: `netsh wlan start hostednetwork`
- 3、显示 wifi 热点信息: `netsh wlan show hostednetwork`
- 4、关闭 wifi 热点: `netsh wlan stop hostednetwork`
- 5、如图一，配置 wifi 并开启，查看连接情况。可以看到，当手机连接上电脑 wifi 时，客户端数变为 1，且显示其 mac 地址。
- 6、查看手机的状态信息如图二，可以发现图一显示的 mac 地址确实为手机的 mac 地址，状态信息还显示所连 wifi 的 ip 地址。
- 7、查看电脑的无线连接属性如图三，可以看到，无线连接的 ip 地址为 192.168.137.1，即为手机所连 wifi 的网关
- 8、查看手机连接属性如图四，可以看到，PC 热点使用 WPA2 PSK 进行身份认证



- 9、在 PC 端使用 Wireshark 抓取数据包如下图，通过手机连接信息中的 ip 地址对数据包进行过滤（由



于中间重新关闭并开启 PC 热点，所以手机连接后所分配的 ip 地址有所不同)

ip addr == 192.168.137.207					
Time	Source	Destination	Protocol	Length	Info
117 12.025562	192.168.137.1	192.168.137.207	DHCP	344	DHCP Offer - Transaction ID 0xdb0ec1f0
119 12.038011	192.168.137.1	192.168.137.207	DHCP	344	DHCP ACK - Transaction ID 0xdb0ec1f0
122 12.195269	192.168.137.207	192.168.137.1	DNS	80	Standard query 0x68f2 A connect.rom.miui.com
123 12.200964	192.168.137.1	192.168.137.207	DNS	490	Standard query response 0x68f2 A connect.rom.miui.com A 111
124 12.206218	192.168.137.207	111.13.141.5	TCP	74	34499 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
125 12.244073	111.13.141.5	192.168.137.207	TCP	74	80 → 34499 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452
126 12.245961	192.168.137.207	111.13.141.5	TCP	54	34499 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
127 12.247801	192.168.137.207	111.13.141.5	HTTP	228	GET /generate_204 HTTP/1.1
128 12.277552	192.168.137.207	192.168.137.1	TCP	74	56430 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
129 12.285759	111.13.141.5	192.168.137.207	TCP	54	80 → 34499 [ACK] Seq=1 Ack=175 Win=30016 Len=0
130 12.286542	111.13.141.5	192.168.137.207	HTTP	164	HTTP/1.1 204 No Content
131 12.292466	192.168.137.207	111.13.141.5	TCP	54	34499 → 80 [ACK] Seq=175 Ack=111 Win=65535 Len=0

Frame 122: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
Ethernet II, Src: XiaomiCo_8a:19:2e (c4:0b:cb:8a:19:2e), Dst: a2:a8:cd:95:f3:6c (a2:a8:cd:95:f3:6c)
Destination: a2:a8:cd:95:f3:6c (a2:a8:cd:95:f3:6c)
Address: a2:a8:cd:95:f3:6c (a2:a8:cd:95:f3:6c)
...1.1... = LG bit: Locally administered address (this is NOT the factory default)
...0... = IG bit: Individual address (unicast)
Source: XiaomiCo_8a:19:2e (c4:0b:cb:8a:19:2e)
Address: XiaomiCo_8a:19:2e (c4:0b:cb:8a:19:2e)
...0... = LG bit: Globally unique address (factory default)
...0... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

- (1) 首先，PC 端（192.168.137.1）使用 DHCP 服务分配 ip 给连接客户端（手机）
- (2) 手机端所分配的 ip 为 192.168.137.207，通过比对 mac 地址 c4:0b:cb:8a:19:2e，可以知道这就是连接 wifi 的小米手机
- (3) 手机端每次访问网络，先向 PC 端（网关路由器）发送 DNS 请求，得到响应的目标 ip 地址后，再进行 TCP、HTTP 请求等的访问

(3) 使用树莓派创建 wifi 热点，PC 端连接热点并抓取数据包。

- 1、先配置树莓派的相关设置，使用 wlan0 开启热点如图一
- 2、查看 wlan0 的 ip 地址，为 192.168.12.1，如图二

```
File Edit Tabs Help
pi@raspberrypi: ~/create_ap
Failed to create interface mon.wlan0: -95 (Operation not supported)
wlan0: could not connect to kernel driver
Using interface wlan0 with hwaddr b8:27:eb:c6:50:bf and ssid "cjt2"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA c4:0b:cb:8a:19:2e IEEE 802.11: associated
wlan0: AP-STA-CONNECTED c4:0b:cb:8a:19:2e
wlan0: STA c4:0b:cb:8a:19:2e RADIUS: starting accounting session 5ACC7703-00000000
wlan0: STA c4:0b:cb:8a:19:2e WPA: pairwise key handshake completed (WPA)
wlan0: STA c4:0b:cb:8a:19:2e WPA: group key handshake completed (WPA)
wlan0: STA 14:5f:94:6d:87:03 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED 14:5f:94:6d:87:03
wlan0: STA 14:5f:94:6d:87:03 RADIUS: starting accounting session 5ACC7703-00000000
wlan0: STA 14:5f:94:6d:87:03 WPA: pairwise key handshake completed (WPA)
wlan0: STA 14:5f:94:6d:87:03 WPA: group key handshake completed (WPA)
wlan0: STA 00:0d:0a:4b:17:a3 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED 00:0d:0a:4b:17:a3
wlan0: STA 00:0d:0a:4b:17:a3 RADIUS: starting accounting session 5ACC7703-00000000
wlan0: STA 00:0d:0a:4b:17:a3 WPA: pairwise key handshake completed (WPA)
wlan0: STA 00:0d:0a:4b:17:a3 WPA: group key handshake completed (WPA)

pi@raspberrypi: ~
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.80.110 netmask 255.255.0.0 broadcast 172.16.255.255
    inet6 fe80::2843:488a:1d5e:2779 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:93:08:ea txqueuelen 1000 (Ethernet)
    RX packets 55578 bytes 49419820 (47.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11911 bytes 1227039 (1.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 583 bytes 48375 (47.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 583 bytes 48375 (47.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=103<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.12.1 netmask 255.255.255.0 broadcast 192.168.12.255
    ether b8:27:eb:c6:50:bf txqueuelen 1000 (Ethernet)
    RX packets 2182 bytes 277247 (270.7 KiB)
```

- 3、在 PC 端抓取数据包

ip addr == 192.168.12.1					
Time	Source	Destination	Protocol	Length	Info
9 0.187026	192.168.12.45	192.168.12.1	DNS	74	Standard query 0x3617 A www.google.com
10 0.219140	192.168.12.1	192.168.12.45	DNS	74	Standard query response 0x3617 A www.google.com
20 4.582581	192.168.12.45	192.168.12.1	DNS	69	Standard query 0xca6e A baidu.com
21 4.588945	192.168.12.1	192.168.12.45	DNS	101	Standard query response 0xca6e A baidu.com A 111.
33 4.673086	192.168.12.45	192.168.12.1	DNS	73	Standard query 0x3f6d A www.baidu.com
34 4.675918	192.168.12.1	192.168.12.45	DNS	135	Standard query response 0x3f6d A www.baidu.com CN

Frame 10: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Raspberr_c6:50:bf (b8:27:eb:c6:50:bf), Dst: Projecti_4b:17:a3 (00:0d:0a:4b:17:a3)
Destination: Projecti_4b:17:a3 (00:0d:0a:4b:17:a3)
Source: Raspberr_c6:50:bf (b8:27:eb:c6:50:bf)
Address: Raspberr_c6:50:bf (b8:27:eb:c6:50:bf)
...0... = LG bit: Globally unique address (factory default)
...0... = IG bit: Individual address (unicast)
Type: IPv4 (0x0800)

- (1) 可以看到，192.168.12.1 对应的设备确实为树莓派（b8:27:eb:c6:50:bf）
- (2) PC 端所分配到的 ip 地址为 192.168.12.45
- (3) PC 端访问网站时，会先发送 DNS 请求给树莓派（网关路由器），得到响应的目标 ip 地址后，再



进行 TCP、HTTP 请求等的访问

实验 4：在实验 3 基础上，查看建立的热点信息（例如，BSSID、无线电类型、频道、已连接用户的客户端数目和 mac 地址等）。如何判断 wifi 热点有没有被蹭网？请实验验证。

- 1、使用 netsh wlan show hostednetwork 显示热点信息
- 2、由图一可以看到，热点 BSSID、无线电类型、频道、已连接用户的客户端数目、mac 地址等信息
- 3、查看已连接用户 mac 地址，与已知的 mac 地址进行比对即可判断 wifi 热点有没有被蹭网



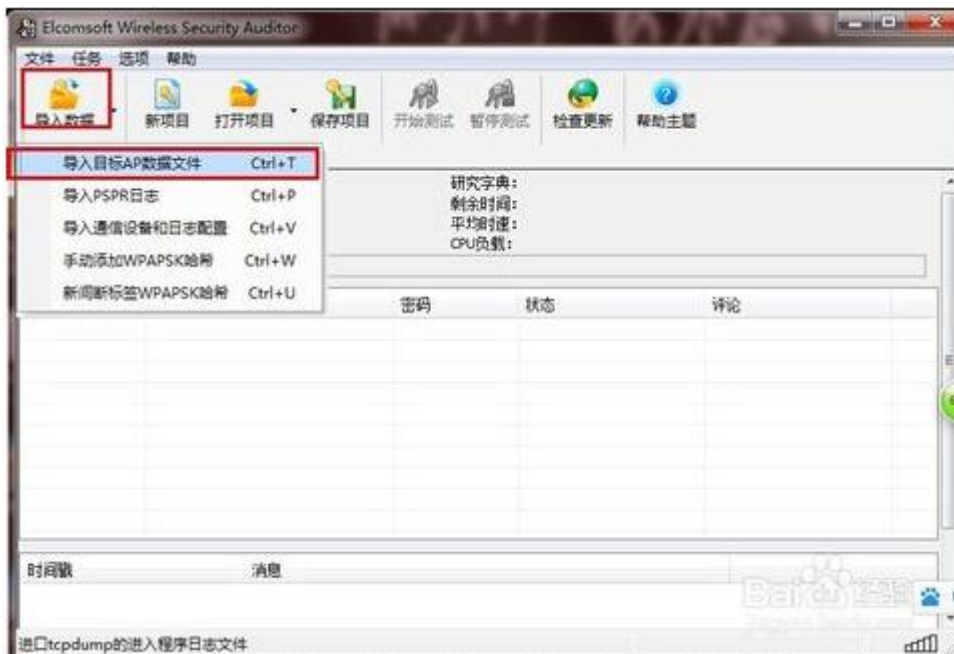
实验 5：如何对热点密码进行暴力破解攻击？写出思路、使用工具，并实例验证。（使用了什么破解工具、简述此工具的功能。蹭网是否成功？）

思路：

- 1、抓握手包：用设备抓握手包，即 WiFi 验证时客户端发送的数据包，该数据包内有已被加密的密码哈希
- 2、选择字典：抓握手包后用软件配合字典进行破解。

由于 Windows 下无法抓握手包，课上也没来得及用树莓派抓包，这里就简述一下步骤：

- 1、安装 EWSA
- 2、抓握手包
- 3、将握手包导入 EWSA



- 4、添加字典：



5、点击“开始测试”即可开始跑包。

实验 6：如何发现和判断流氓热点？写出思路、使用工具，并实例验证。

观察 MAC 时间戳。生成相同网络的接入点会拥有高度同步的内部时钟，因此，接入点不断地交换时间戳以实现同步，这个时间是毫秒级的，同步增量为 25 微秒。但大多数流氓热点在进行时间戳同步时往往会出现各种各样的错误，因此可以通过检测这种错误来发现流氓热点。

实验总结：根据以上实验，请对 wifi 热点的安全性做一个综述（如有引用文献资料，请标出）。

我们可以通过电脑和手机中创建 wifi 热点来给设备提供无线网络的连接，在设备中可以查看到身份认证、IP 地址和 MAC 地址等网络连接信息。

在无线网络存在巨大的安全隐患，公共场所的免费 WiFi 热点有可能就是钓鱼陷阱，而家里的路由器也可能被恶意攻击者轻松攻破。网民在毫不知情的情况下，就可能面临个人敏感信息遭盗取，访问钓鱼网站，甚至造成直接的经济损失。常见公共 WiFi 热点安全隐患有两类。架设 WiFi 热点基站程序简单、成本低，个人使用自带的手机或路由器即可操作，导致虚假公共 WiFi 热点普遍存在。常见隐患有两类，一、非法个人或组织建立 WiFi 热点伪装成公共 WiFi 热点，并植入恶意内容，监听用户通信链路，收集用户隐私信息，这就是常见的“寄生虫 WiFi 热点”；二、非法个人或组织攻击无辜 WiFi 设备，破解其漏洞后植入恶意程序，进行非法操作，这类被称作钓鱼 WiFi 热点。

在移动设备连接到 WiFi 热点时，应确认 WiFi 热点的安全性来防范可避免的安全隐患。

【交实验报告】

上传实验报告：<ftp://222.200.180.109/>

截止日期（不迟于）：两周之内完成

上传小组实验报告。上传文件名格式：小组号_实验名.pdf（由组长负责上传）

例如：文件名“6_网络攻击分析实验.pdf”表示第 6 组的网络攻击分析实验报告

注意：不要打包上传！