首先需要一个十六进制阅读器。

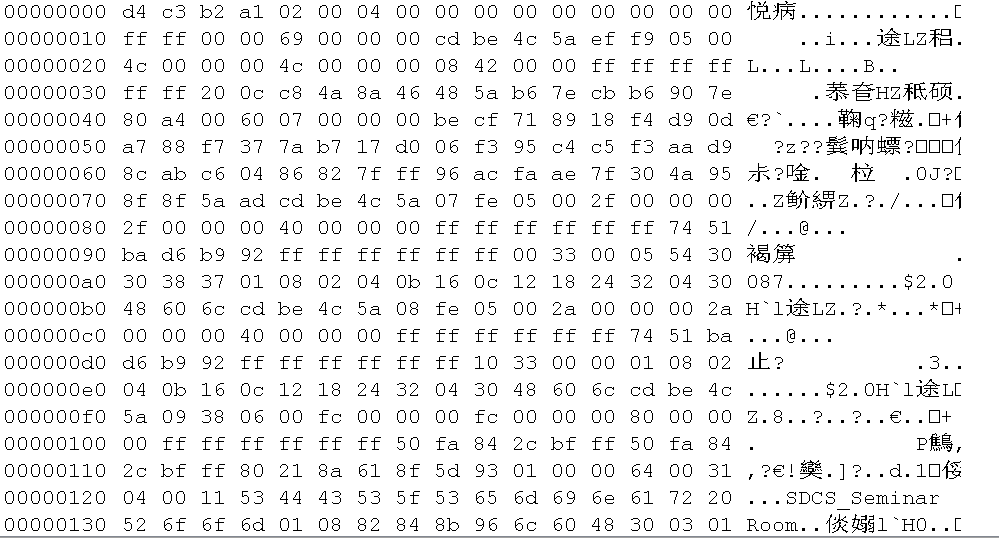
Linux下直接使用vim打开，然后esc调成管理台，输入”:%!xxd”回车即可

Windows下使用Notepad++，加载一个HEX viewer插件就可以。也可以使用收费软件Ultraedit。

第一节

（每8位算1字节，帧中前8位是第1字节，9~16位是第2字节，后同）

在十六进制下打开cap包，内容大概如下：



0000~0017

一共24字节。这24字节是头，代表这是个cap包。每次抓cap包开头都是这些值，固定的：

d4 c3 b2 a1 02 00 04 00 00 00 00 00 00 00 00 00 ff ff 00 00 69 00 00 00

之后是各Frame的内容。Frame分成3到4部分，下面以第一帧举例

第1部分：开头16字节（0018~0027）

前8字节记录了该Frame产生的Timestamp，前4字节是秒数（距1970.1.1 0:00 UTC）。后4字节是秒之后小数点那串数字直接拎出来化成十六进制写入。其实这8字节几乎没用。

后8字节是该Frame传输的字节数和cap捕获的字节数（这两个数字似乎永远一样，可以使用其中一个作后续操作）。但读法有点不一致，靠后面的字节是高位，越靠前越是低位。

如图中0020~0023位是4c 00 00 00。实际应该从后往前写00 00 00 4c，写成十进制是76。代表接下来的76字节就是这一帧的内容，76之后就是下一帧了。

第2部分：帧内容（0028~0073）

共76字节。根据第1字节可以判定这是什么类型的帧，从而进行相应处理。目前已知的类型有：

Data帧： 08

Probe Request： 40

Probe Response：50

Beacon帧： 80

QoS Data（含EAPOL Key）：88

Null Function：48

802.11 Block Ack Req： 84~87

802.11 Block Ack：94~97

Disassociate：a0~a3

CF End：e4~e7

CF Ack-poll：78~7b

Authentication：b0~b3

Association Response：10~13

Acknowledgement：d4~d7

Action：d0~d3

Clear-to-send：c4~c7

Deauthentication：c0~c3

Disassociate：a0~a3

红字要重点关注

那么，每帧排查，如果开头不是88就可以下一帧了，否则这可能是Key，继续分析。

第二节

知道这是QoS Data后，就可以进一步分析了。

QoS Data有3部分，第1部分是开头16字节，记录帧长度和发送时间；第2部分是第17~50字节，可以分析出这是QoS Data，以及部分参数。剩下的就是Data。

但Key有点特别：第43~第50字节被单独划分出来作为Logical-link Control字段，为固定值aa aa 03 00 00 00 88 8e。

所以有以上特征的可以被初步认为是Key。第52字节为03的话，可以直接认定这是EAPOL Key，即四次握手数据。

但我怎么知道这个EAPOL Key是第几次握手？

WPA资料显示，Key Information字段（56~57字节，共2字节，16位）的值就代表了这是第几次握手。

所有Key的Information中都有第3位（第n位即从低往高数倒数第n+1位，下同）为1，表示Key类型为Pairwise key（WPA加密方式恒为成对密钥）。

第0~2位，共3位，表示这次认证的MIC使用的加密类型，如果是1（001）则表明是WPA加密，使用HMAC-MD5计算MIC；如果是2（010）则表明是WPA2加密，使用HMAC-SHA1计算MIC。注意这只是区分计算MIC的算法，不影响其他。在一次认证中，四次握手这个值均为一致的。因此一次认证中，该字段低4位均为9（代表WPA）或a（代表WPA2）。

高12位中，不同次握手的值会不一样。

第1次为0x0089（WPA）或0x008a（WPA2）

第2次为0x0109（WPA）或0x010a（WPA2）

第3次为0x13c9（WPA）或0x13ca（WPA2）

第4次为0x0309（WPA）或0x030a（WPA2）

这16位的取值与该次握手的作用有关，介绍WPA加密的书籍里会有解释，此处略。

当有一组完整的这4种取值的Key帧时，表明这是一次完整的WPA握手，可以通过计算去破解密码。

此时需要从中提取一些信息，以便进行后续计算。这些信息分别为：

（注：以下的字节位置，均包含了帧开头16字节。如果用WireShark等观察，则需要减去16）

（1）AP Mac：第1/3次握手的第27~32字节，或第2/4次握手的第21~26字节。共6字节，其十六进制表示形式就是明文MAC地址，但后面MIC的计算时这些会保留6字节的形式写入字符串中。

（2）STA Mac：第1/3次握手的第21~26字节，或第2/4次握手的第27~32字节。共6字节。

以上两个是WiFi无线接入点和移动设备的物理地址，对任意设备来说都是固定不可改变的。（当然现在有伪造MAC地址的技术，但不在考虑范围内，如果不是握手过程中改变MAC，那么是否伪造MAC对破解结果不会有影响）

（3）AP Nonce：第2次握手的68~99字节，共32字节。

（4）STA Nonce：第1次握手的68~99字节，共32字节。

以上两个是两台设备在认证时各自随机产生的字符串，每次不同的认证的Nonce都会不一样。

（5）AP的SSID：Beacon帧的27~32字节，共6字节。

WiFi名字没有存放在握手包里，而是在AP以Beacon帧形式广播到空气中时公开，移动设备搜索WiFi时看到WiFi的名字就是接收Beacon帧后解析出来的结果。因此需要通过握手包中的AP Mac字段与各Beacon帧的Mac匹配，一致的就是握手过程涉及的AP。

（6）EAPOL MIC：第2次握手的132~147字节，共16字节。

这是STA在WPA加密算法计算后生成的MIC，发给AP校验的。

（7）STA的EAPOL报文：第2次握手的51~结束。字节数由帧中表示帧长度的值决定（一般是121）。

后来计算MIC时要用到这个。但是使用这个时MIC是还没计算出来的，因此MIC字段的16字节，要填成0，再保存。

得到这些信息后，就可以开始用字典破解了。

详细见文档《WPA(WPA2)加密流程》。