CHAPTER 9

Customizing a Security Live Linux CD

Security and system-rescue activities are among the best uses of a live Linux CD. On one read-only medium, you can gather a range of tools to fix a broken operating system, scan file systems for intruders, watch traffic on the network, and do hundreds of other activities to check and protect your computing environment.

Putting together your own security live CD offers a means of having all the tools you need at your fingertips in a way that you can apply to any PC you can reboot. You have fewer worries when you go to fix a broken system because you can start with a clean, freshly booted operating system each time. In this way, less chance exists that the tools you would use to fix a system are themselves broken or infected.

Creating a custom security live CD also can mean adding tools that might not be on other security live CDs. For example, some software is licensed in a way that you can use it for personal use but might not be available for someone to redistribute. Examples of such things are proprietary drivers or utilities that come with a hardware peripheral. So these features that they can't legally put on a widely distributed Linux live CD might legally go on the one you carry around with you to use when it's needed.

This chapter covers some of the features that have been put into live Linux CDs that are used for security and rescue purposes. It also describes additional components you might want to add to your own custom live Linux security CD. For demonstration purposes, the chapter focuses on the BackTrack live CD.

Exploring Security Live CDs

Many live Linux CDs for security already are available today, so chances are good that you can start your own Live CD based on an existing security-oriented Live CD. Popular security live CDs include some that are based on Knoppix, Slackware, and Gentoo. You can choose where to start based on Linux distribution, as well as on the tools each CD provides.

To begin demonstrating the types of functions you might find on a security Live CD, this chapter starts with the BackTrack network security suite (www.remote-exploit.org). Chapter 2, "Playing with Live Linux CDs," includes an introduction to BackTrack, to help you get started and learn the basics. This chapter looks at BackTrack with an eye toward the following:

- **Understanding how it starts up**—You'll learn how BackTrack boots up and starts services, as well as the state in which users begin using the live CD (such as run levels, graphical and text-based interfaces, and user accounts).
- **Learning which tools it includes**—You'll become familiar with the types of features and tools included in BackTrack to help you put together your own set of custom computer security tools.

NOTE

BackTrack, like other Slax-based live CDs, includes many of the boot files provided from the Linux Live CD/USB Scripts site (www.linux-live.org). Refer to that site to learn more about how the scripts, kernels, and other components from linux-live.org can be used to produce live CDs from an installed Linux system.

Figure 9-1 illustrates the features that BackTrack pulls into its arsenal of security tools. Because BackTrack was created from a combination of tools from two other popular security-centric live Linux distributions (Auditor and Whax), you can get a good sense of the tools you would want to have on your own security live CD.

If you are using BackTrack, open the BackTrack menu (lower-left corner of the screen) to select security tools to try out. In Figure 9-1, the applications displayed include the Autopsy Forensic Browser, the Automated Image and Restore (AIR) tool for creating and restoring disk images, and the AutoScan utility for exploring your network.

FIGURE 9-1 Size up systems, hunt down exploits, and explore networks with BackTrack.

Because security live CDs are not generally meant to be used for gaming or other high-end graphics applications, a lot of disk space can be saved by including a simple window manager (or no window manager at all). Because it is lightweight but carries enough features to launch many useful X applications, the Fluxbox window manager is available on many of the smaller security live CDs (50MB to 180MB). BackTrack includes Fluxbox for users who want to use it; the KDE interface is used with BackTrack by default.

To understand the foundation on which this security tool chest is built, however, the next sections step you through the boot process and basic setup used to support the security tools.

Booting the BackTrack Security Live CD

To begin exploring how BackTrack is configured, look at the boot files it uses. As with most Linux live CDs, BackTrack uses the Isolinux boot loader. The boot loader is configured quite simply and is basically the same configuration you get with other live CDs based on Slax (www.slax.org).

The `isolinux.cfg` file is stored in the root (/) of the live CD, to direct the boot process. Three labels exist: `slax`, `linux`, and `memtest`. The `memtest` label simply runs the `memtest` command to check your RAM. The default `slax` label is the same as the `linux` label, except that the `slax` label sets the default `vga` mode to 1024×768, 16 colors (`vga=0x317`, the hexadecimal equivalent of `791`). The following options, that can be used from the boot prompt (using either `linux` or `slax` labels) are described on the message screen by pressing F1:

- `linux debug`—Interrupts the boot process on several occasions to open a shell to check progress.
- `linux copy2ram`—Copies a complete file system to RAM. This can be used only if enough RAM is available.
- `linux floppy`—Enables floppy automounting during startup.
- `linux load=modules`—Loads indicated modules from the `/optional` directory.

You can pass other boot options to BackTrack from the boot prompt, including `nocd`, `nohd`, and `nodma`.

You can set the `probeusb` parameter to find USB devices earlier in the process than it would normally do so. You can also set the specific amount of RAM you want to allow BackTrack to use when it mounts the root (/) directory by setting the `ramsize=` parameter (in bytes). By default, 60 percent of your available RAM is used for the `tmpfs` file system that will eventually hold the live CD's root file system.

Boot Components

These bootloader components are used during the default boot process for BackTrack:

- **Splash screen**—The first screen that appears when BackTrack boots is defined in the `boot/splash.cfg` file. That file reads in the BackTrack splash screen (`boot/splash.lss`) and line of text instructing the user to press Enter. Before booting, the user has the option of pressing F1 (to see several boot options contained in the boot/splash.txt file) and then pressing F2 to return to the original splash screen.
- **RAM disk**—From `append` options with the `slax` label, a 4MB RAM disk is created (`ramdisk_size=4444`).
- **Kernel**—The kernel is booted from the `boot/vmlinuz` file. The kernel is a 2.6 kernel that includes support for Unionfs and Squashfs file systems.
- **Initial RAM disk**—The initial RAM disk files and directories are provided from the `initrd.gz` file (`initrd=boot/initrd.gz`). This file is unzipped and mounted as the root file system on the RAM disk. After this file is unzipped,

the kernel can direct the `linuxrc` file (located at the root of this file system) to continue the boot process.

The linuxrc Script

The `linuxrc` script does a lot of work to set up the file systems for the live CD. Because the `linuxrc` script is derived from the same script that comes with Slax, the way BackTrack sets up its file systems is similar to that of Slax (and other distributions based on linux-live.org scripts).

Much of the processing of the `linuxrc` script is done using functions defined in the `liblinuxlive` script (also in the root of the `initrd` file). Here are some of the things that the `linuxrc` script does in BackTrack:

- Mounts the `/proc` and `/sys` file systems.
- Checks whether the `debug` option was given at the boot prompt. If it was, the rest of the script runs in debug mode.
- Runs `modprobe` to load essential modules. Many of the modules loaded are used to support needed file system types (`isofs`, `squashfs`, `unionfs`, and `vfat`). If an NTFS file system is present, `ntfs` support is also added. Support is also added for USB storage devices if the `probeusb` boot option was given.
- The root file system (/) is ultimately set up as a `tmpfs` file system. First, individual `squashfs` file systems, representing `/bin`, `/etc`, `/root`, `/usr`, `/var`, and others, are mounted read-only from the `/base` directory of the CD. Then they are merged with the writeable `/UNIONFS` directory. (See Chapter 5, "Looking Inside Live CD Components," for descriptions of UnionFS and Squashfs.)

  Later, the entire root file system is converted to a `tmpfs` file system. The advantage of a `tmpfs` file system is that it runs from system memory. This enables you to make the files and directories from the read-only medium (CD) writeable. An advantage of `tmpfs` over other file systems that are stored in memory, such as the `ramfs` file system, which can write only to RAM, is that `tmpfs` can store files in both RAM and swap space. A `tmpfs` file system automatically expands if more space is needed (as long as memory is available).
- An image appears behind text as it scrolls during the boot process. The location of the image is `/etc/bootsplash/themes/Linux/config` in the file named `bootsplash-1024x768.cfg`. The actual images used are stored in the `/etc/bootsplash/themes/Linux/images` directory.

- With the file system completely set up, `linuxrc` changes to the root of the newly formed directory structure (`pivot_root`). Then it runs `/sbin/init` to start the next phase of the boot process.

Customizing the Boot Process

When creating a live Linux security CD, you have many opportunities to customize a BackTrack live CD so it includes the features you want during the boot process:

- **Custom splash screens**—You can add your own images, both to the initial boot screen and as a background as messages scroll by. See Chapter 5 for information on creating images to use on your boot screens.
- **Multiple boot labels**—Just as `memtest` can be run from a boot label, you can add multiple floppy images to be run individually from Isolinux. You can copy any bootable floppy image to the `/boot` directory and add a label to `isolinux.cfg` so it can be launched from the boot prompt. This can be useful if you have utilities on floppy disk that came with hardware you want to configure. You can use the utilities without booting up the entire live CD.
- **Capability to add a file system**—Create a directory structure of files you want to have on the CD and convert that structure to a `squashfs` file system. Add the file system file to the `/base` or `/modules` directory, naming it with a `.mo` suffix.

For more ideas on configuring Isolinux to have your live CD boot as you would like, refer to Chapter 5.

The Initial System State

Security-oriented live CDs might be booting up on a computer that is broken in some way. Because problems might include a dysfunctional video card or a connection to a network from which the computer could be under attack, start-up often takes you to a minimal system state. This section describes the processing that occurs from when the `init` process takes and then continues to the point at which the user is presented with a login prompt.

Default Run Level

Even though BackTrack includes a graphical user interface, you will see a shell interface instead of an X desktop by default. BackTrack does this by starting the system at run level 3 (based on the `initdefault` value set in `/etc/inittab`).

Run level 3 is the most common run level for starting a system that is not using a GUI by default. Instead, it boots to a text-based login prompt and then, after the user logs in, presents that user with a shell interface. This level might or might not

start up network interfaces. In the case of BackTrack, the network interfaces are not started automatically. Here are some of the highlights of what occurs after the `init` process sets other processes in motion to start up system services:

- **`rc.S` script**—The `/etc/rc.d/rc.S` script is run every time the live CD is started (regardless of the `init` state). This script mounts and checks any file systems listed in `/etc/fstab`, turns on swap, starts `udev` (to manage removable devices), and configures plug-and-play devices.
- **`rc.M` script**—The `/etc/rc.d/rc.M` script runs when the live CD boots up to any multiuser run level (2, 3, 4, or 5). It does some logging (sending `dmesg` output to `/var/log/dmesg` and running the `syslogd` system log daemon). This script also runs the `rc.slax` script, which responds to several options that might have been added at the boot prompt. For example, the `noguest` option disables the guest user account and `passwd` causes the script to prompt the user for a new root password. For the `autoexec` option, the script runs the command given (for example, `autoexec="date"`) and then reboots.
- **System V `init` scripts**—One of the last things the `rc.S` script does is run the `/etc/rc.d/rc.sysvinit` script to start any System V `init` scripts that are set to run at the current run level. For example, in run level 3, the script starts any scripts beginning with the letter *S* in `/etc/rc.d/rc3.d` directory.
- **MySLAX scripts**—The only script currently in the `/etc/rc.d/rc3.d` directory is the `SInstall` script. If you added any MySLAX Modulator scripts to this directory (with names that begin with `Install_*`), they would be executed to install the software it contained. Information on adding MySLAX software is described later in this chapter.

If you had watched as the messages were displayed, the process just described took place between the message beginning with `"INIT:"` and the login prompt.

Customizing the init Process

Features you add to a security live CD during the `init` process should focus on the kinds of services you want to enable or disable during the live CD session. Here are some customizations you might want to consider:

- **Add boot options**—You can add support for different boot options that are dealt with during the init process. For example, the /etc/rc.d/rc.M script. That script reads the boot options (stored in /proc/cmdline) and turns various services on or off based on those settings.
- **Add system services**—If you want to have a particular service launched at boot time, you can add a script for starting that service to a run level directory. In this case, you could add start-up scripts (beginning with the letter *S*) to the `/etc/rc.d/rc3.d` directory.

After the boot process completes, the user sees the login prompt and some information on the screen. That information includes the root password and ways to start the GUI and the network.

Logging In

With BackTrack, you are expected to log in as the root user from a text-based login prompt. No other user account is available as a login account.

BackTrack comes with a root password already assigned. That password appears on the screen when you first boot up. Although this might seem like a security concern, remember that there is no network interface at this point, so you can run the `passwd` command to change the root password before starting up network interfaces and allowing any remote login services to the machine. You could also type `passwd` as a boot option to be prompted for a root password during the boot process.

Although other active user accounts exist, none is enabled for you to log into. Knoppix boots up to a desktop owned by the knoppix user (who can request root privilege without a password), but you need to add a new user to operate the live CD as a nonroot user (running `adduser username` steps you through the process).

As noted earlier, the login prompt is text-based and is preceded by instructions (the contents of `/etc/issue` are listed). Those instructions include the root password and commands for configuring the desktop (`xconf`), starting the desktop (`startx`), starting a simpler desktop (`gui`), and getting an IP address using DHCP (`dhcpcd`). Figure 9-2 shows the login screen just described.

Ways of configuring the login process include the following:

- Change the text shown before the login prompt by editing the `/etc/issue` file.
- Set up other user accounts so you don't have to log in as root.
- Configure the user environment. You can modify user environment variables by editing the `/etc/profile` file.
- Configure the user home directory. The `/root` directory is populated with configuration files that are set up in advance in the `/etc/skel` directory. You can add or change configuration files from there.

If the user can successfully log in as a root user, the next areas to look at are those that set up the desktop environment and features for configuring necessary peripherals (network interfaces, printers, sound cards, and so on).

FIGURE 9-2 The text-based login screen includes information from the `/etc/issue` file.

NOTE

One variable set in `/etc/profile` that you might look at is the `PATH` variable. BackTrack added a dot (`.`) to the `PATH`. This allows commands to be run from the current directory (an activity that typically needs to be asked for explicitly, by typing something such as `./command`). BackTrack needs this feature as part of the way it lets you run commands from the BackTrack menu. It basically opens a terminal window with the shell open to the directory containing the command. It expects you to be able to run the command from there.

Configuring the Desktop

Although many BackTrack features can be used from the shell, BackTrack helps users navigate through the hundreds of commands and utilities by creating a BackTrack menu for the desktop. Some of the design decisions that went into configuring the desktop for BackTrack are worth considering as you go to customize your own security CD.

Desktop Environment or Window Manager

KDE is used by default as the desktop environment for BackTrack. Because BackTrack doesn't have the size limitations that mini security distributions have of fitting on a 50MB bootable business card or 180MB mini disk, there is room for a larger desktop environment, such as KDE. Those mini distributions tend to go for lightweight window managers, such as Fluxbox, to provide a simple GUI.

Other window managers are available with BackTrack, including Fluxbox (`fluxbox`), Motif (`mwm`), and Twm (`twm`). So, with a slow CPU or low amounts of memory, users can switch to another window manager. Of course, you will lose some of the KDE features and tools if you switch to a simple window manager.

In particular, Fluxbox is used with a number of live Linux CDs. For example, Fluxbox is the primary window manager used on other security-oriented live CDs such as INSERT (www.inside-security.de/insert_en.html), PHLAK (www.phlak.org), and Knoppix-STD (www.knoppix-std.org). It is also offered as a Window manager on some mini live CDs, such as Damn Small Linux (www.damnsmalllinux.org).

To change the default desktop environment from KDE to a simpler Fluxbox window manager, edit the `/usr/X11R6/lib/X11/xinit/xinitrc` file. Then comment out the `startkde` line at the bottom and add `fluxbox` on a line by itself. Save the file. Then type `startx` to start the desktop, using the Fluxbox window manager. Fluxbox has been configured to incorporate the BackTrack menu into the basic Fluxbox menu (right-click the desktop to see it).

Backtrack Menu

The BackTrack submenu on the KDE menu is the most obvious enhancement BackTrack includes. This is where BackTrack gathers hundreds of security tools that are available from the live CD. The tools are divided into 15 categories. Figure 9-3 shows the BackTrack menu displayed from the KDE menu and from the Konqueror window.

Alongside the BackTrack menu in Figure 9-3 is the KDE Menu Editor. By opening the BackTrack menu in the KDE Menu Editor, you can see details about each menu item. For each application, you can see details about how it is run. You can open the KDE Menu Editor to work with the BackTrack menu by right-clicking the K Menu button and selecting Menu Editor.

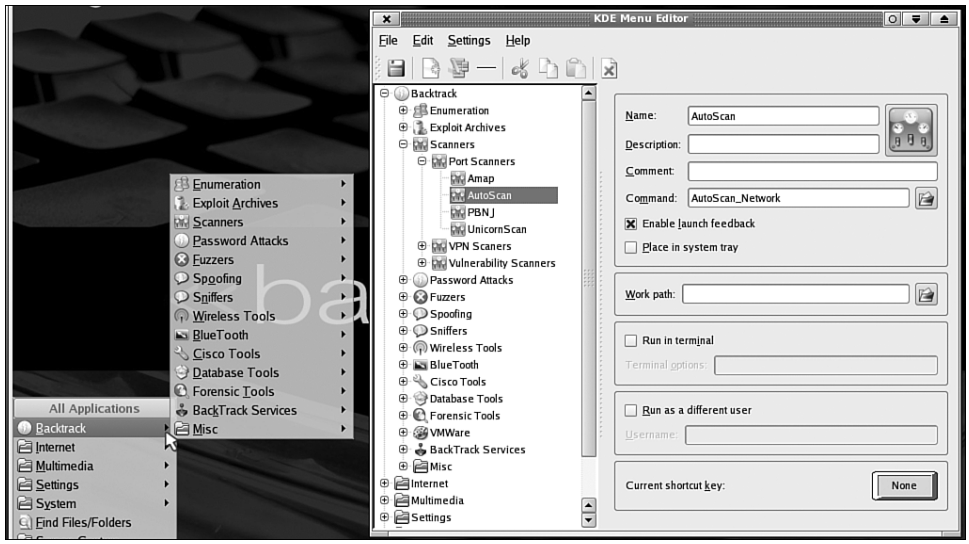

FIGURE 9-3 Security tools on the BackTrack menu are divided into 15 categories.

Because many of the security tools available are commands rather than graphical utilities, BackTrack takes an interesting approach to making those tools available from the desktop. Selecting a menu item that represents a command opens a shell that displays the help message associated with that command. In some cases, the current directory for that shell holds the command and any support files needed by that command. Because the default PATH enables you to run commands from the current directory, you can simply run the command (with the options you want) or refer to the files in that directory for further information.

KDE Settings

Because KDE is the default desktop for BackTrack, you can take advantage of the available tools for configuring the KDE desktop. Most the graphical administrative tools for KDE are available from the Settings and System menus on the KDE menu. To tailor the look and feel of the desktop, refer to Chapter 2. The same features described there for configuring KDE in Knoppix can be used to configure KDE here.

Configuring Network Interfaces

Because network interfaces are off by default with BackTrack, before you can use the tools for scanning network resources and accessing remote computers, you need

to turn the interface to your network. Here are two ways to get the interface to a standard wired Ethernet card configured:

- **From the shell**—You can grab an IP address from a DHCP server by typing the `dhcpd` command. You can have the interface brought up and down by typing **`ifconfig eth0 up`** (providing that `eth0` was assigned to your network interface). To bring the interface down, you could type **`ifconfig eth0 down`**.
- **From the KDE menu**—Select the KDE menu button → Internet → Set IP address. The network configurator window lets you type in a static IP address, subnet mask, and default gateway. You can also enter the IP addresses of the DNS servers.

For other types of networking interfaces, GUI tools also are available. To configure a dial-up connection, select the KDE menu button → Internet → Internet Dial-Up Tool. To configure wireless LAN connections, select the KDE menu button → Internet → Wireless Manager.

Again, because BackTrack uses KDE, graphical KDE tools also are available for configuring and managing network interfaces and features. Open the KDE Control Center by selecting the KDE menu button → Settings → Control Center. Then select Internet & Network to see Wireless Network (to autodetect or manually configure wireless interfaces) and Network Monitor.

The bottom line is, if you can accept the overhead that comes with using KDE (disk space and memory required), using KDE as your desktop environment on a security live CD offers a lot of the tools you need for basically configuring your system. By offering the Fluxbox, XFCE, blackbox, or other low-end window manager, someone using your security live CD can still switch to those window managers. Although not all KDE tools will be available, they will be capable of running most of the security utilities that come with BackTrack.

NOTE

In BackTrack, the BackTrack menu has been integrated into the Fluxbox window manager. Although a Fluxbox user will be missing some KDE configuration tools, most of the security tools should be available to run directly from the BackTrack menu that appears when you right-click the Fluxbox desktop.

Running Security Tools

With a solid live CD in place, including an appropriate boot loader, user environment, and basic configuration tools, it's time to move on to choosing the software that makes the live CD special. In the case of a security-oriented live Linux CD,

that means finding the tools to check systems and networks, evaluate and plug security holes, and deal with problems, such as broken or exploited systems.

Again, BackTrack is used to illustrate the kind of tools you might include on your own security-oriented live Linux CD. Literally hundreds of utilities, resources (such as exploit databases), and services are built into BackTrack that you might find useful for your own Live CD.

Figure 9-4 illustrates the types of features that BackTrack includes.

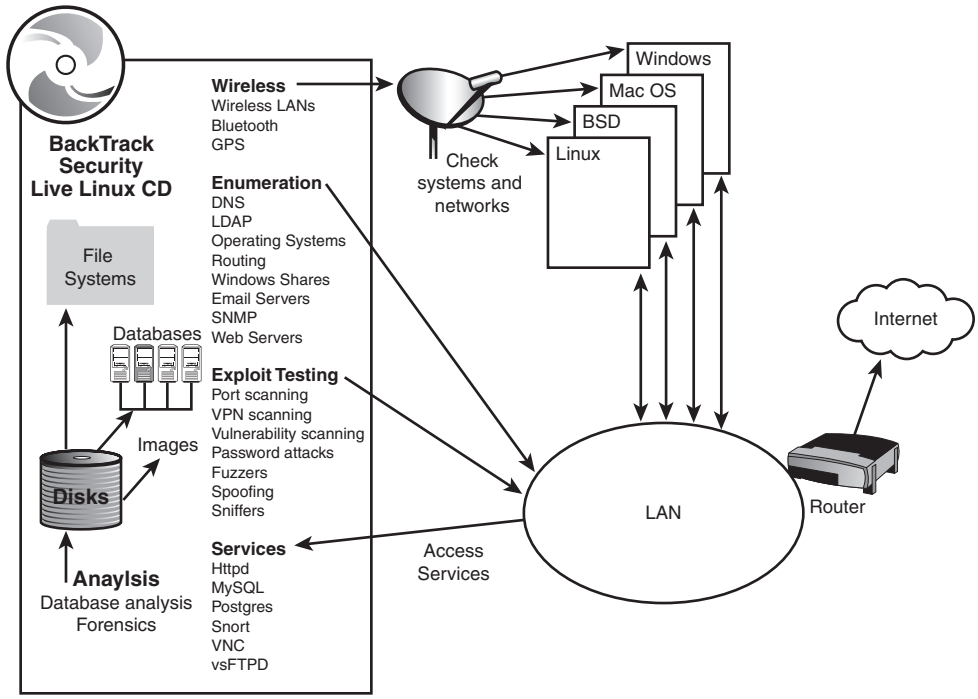

FIGURE 9-4 Features in BackTrack range from remote-exploit to password-cracking tools.

As you can see from Figure 9-4, features added to a security-oriented Linux live CD can go well beyond a password checker and a port scanner. To support the security tools in BackTrack are features such as archives of security information, to check for vulnerabilities to known exploits. Tools also are available for tracking down problems with particular types of devices, such as BlueTooth or wireless LAN cards.

The coming sections take separate looks at the security topics illustrated in Figure 9-4. You can use those sections to learn a bit about how those tools work and to help evaluate the kinds of features you want to have on your own security live CD.

Running Enumeration Tools

Enumeration tools are used to find out what information about a system, network, or service is available to potential intruders. Including enumeration tools on a live CD lets you test the computers on your network so you can shut off access to information that an intruder could use to do you harm.

BackTrack includes dozens of enumeration tools that can be used to find information about your Internet domain, LDAP servers, operating system, routing, and other services. The following enumeration tools come with BackTrack.

Checking Domain Information

To run commands for checking information about an Internet domain from a Domain Name System (DNS) server, select K menu → Enumeration → DNS. About a half-dozen menu items are available to check out DNS information.

Selecting DNS → Walk lets you run the `dnswalk` command (www.visi.com/~barr/dsnwalk) to debug the zone database for your DNS servers. The `dig` command (K menu → Enumeration → DNS → Dig) is a popular utility for looking up a range of DNS information. `dnsenum.pl` (Perl utility) is another useful tool for checking out your domain. With it, you can find DNS servers, try zone transfers, and check for host names from a list you provide.

Checking LDAP Information

Using the `ldapenum` Perl script (http://sourceforge.net/projects/ldapenum), you can get password and system information from LDAP services running on Windows 2000 and 2003 systems. You can get to the script by selecting K menu → Enumeration → LDAP → Ldap Enum. This same tool can be used to try to log into the accounts it finds. The `ldapenum` command can also be used to check for groups and members.

The Luma (http://luma.sourceforge.net) graphical LDAP management tool (K menu → Enumeration → LDAP → Luma) includes eight plug-ins for finding and working with data on LDAP servers. Plug-ins let you search LDAP address books, search selected LDAP servers, manage users, and access other kinds of information from an LDAP server.

Determining Operating Systems

Tools from the Operating System enumeration menu (K menu → BackTrack → Enumeration → Operating System) can help determine the type of operating system being run. Select P0f from that menu to use a passive fingerprinting technique to determine the type of operating system a computer is running, based on TCP/IP

packets sent from that computer. The `p0f` command (http://lcamtuf.coredump.cx/p0f.shtml) is considered to be passive because it finds information about a system by reading TCP/IP packets on the network to determine the type of operating system and other properties of the host.

Tools such as `p0f` can get information about a remote system without actually contacting it. The `p0f` tool can study TCP/IP packets from any system contacting your network. It can also get information from a system based on the remote system accepting or rejecting a connection from the system running `p0f`.

The `nmap` command (www.insecure.org/nmap) is another tool that is commonly used to explore a remote system. With `nmap`, you can scan whole networks or single hosts. By scanning packets, `nmap` can determine what operating systems hosts are running on a network and what services those hosts are offering. It can also learn what type of firewall each host is running.

The Nmap Front End (`nmapfe` command) provides a graphical interface to `nmap`. Figure 9-5 shows the results from scanning a Linux machine on the local network. The system's name is `einstein`, and TCP ports 21 (ftp), 22 (ssh), rpcbind (111), and X11 (6000) are open. The `ftp` and `ssh` services are shown in red, indicating that services are active on those ports.

The `sinfp.pl` Perl script (www.gomor.org/sinfp) can determine the type of operating system running on a computer, based on the IP address you give the command. For Linux systems, you can also determine information about the version of the current kernel.

The `xprobe2` (http://xprobe.sourceforge.net) command can glean a lot of information about a computer's operating system by simply giving the command the system's IP address. It uses several different information-gathering techniques to find out about the selected system. It tries to determine the distance from the remote system, checks open ports on the system, and checks to determine the type of operating system it is running.

Determining Network Routing

Tools for uncovering information about network routing (from the Routing/Network menu) include the Autonomous System Scanner, or `ass` command (www.phenoelit.de/irpas/docu.html), and the `firewalk` command (www.packetfactory.net/projects/firewalk). To trace packets to a destination, you can use the `itrace` command (http://perfinsp.sourceforge.net/itrace_ppc.html), `traceroute` command (ftp://ftp.ee.lbl.gov), or tctrace command (www.phenoelit.de/irpas/docu.html). `itrace` does a `traceroute` using ICMP echo requests; `tctrace` uses TCP SYN packets to trace packets to their destination.

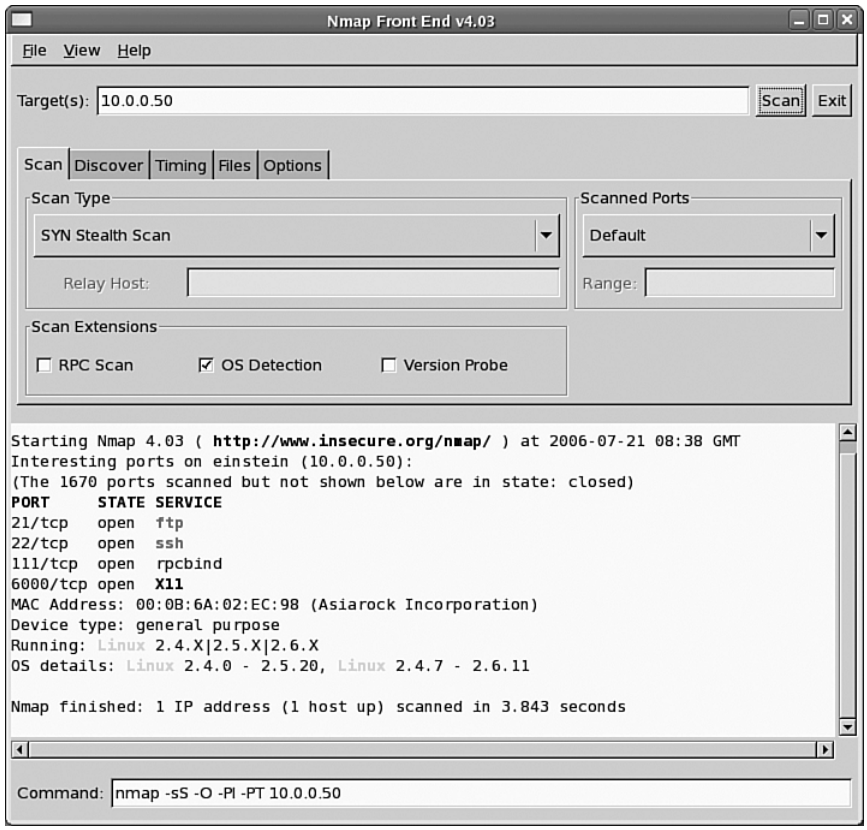

FIGURE 9-5 Determine operating system information and scan ports with `nmapfe`.

Finding Windows (Samba) Shares

Tools for getting information about shared folders from Windows systems using SMB protocol and from Samba servers enable you to look up a range of information about those shared folders. You can select examples of these tools from the SMB menu (K menu → Enumeration → KDE). Scan a range of IP addresses with the `nbtscan` command (www.inetcat.org/software/nbtscan.html) to determine which of those machines is capable of sharing files Windows folders and printers.

The `nmblookup`, `net`, and `smbclient` commands are all part of the Samba suite (www.samba.org). To find the IP addresses of computers using NetBIOS over TCP/IP (as Samba typically does), you can use the `nmblookup` command. The `net` command can be used to administer Samba servers and is similar to the `net` command for Windows and DOS. For example, you can view the status of sessions and shares on your computer using the `net` command.

You can use the `smbclient` command to create an interactive connection to services on an SMB server. With `smbclient`, you interact with an SMB server much as you would talk to an FTP server using the `ftp` command. For example, you can copy files across an `smbclient` connection.

For a graphical interface to SMB features, you can use the SMB/CIFS Share Browser (`smb4k` command). Using the SMB/CIFS Share Browser, you can scan your network for SMB servers, look for available shares, and connect to those shares in various ways. You can find out more about SMB4k by visiting its home page (http://smb4k.berlios.de) .

Finding E-Mail Server Information

Two tools for accessing information about an e-mail server are available by selecting Enumeration → SMTP from the BackTrack menu. Select Relay Scanner to run the `RelayScanner.pl` command to scan selected e-mail servers to see if mail relaying is open. Select SMTP Vrfy against a selected mail server to run the `verfy.pl` command to try to verify the existence of selected usernames. The command comes with a `names.txt` file, which includes nearly 9,000 possible usernames to test for. (You would use this tool to confirm that a spammer couldn't verify the existence of any of the usernames from your mail server.)

Finding Simple Network Management Protocol (SNMP) Information

Utilities for checking SNMP resources are available from the SNMP menu (select K menu → BackTrack → Enumeration → SNMP). The Mibble MIB Browser (www.mibble.org) is a GUI tool for parsing Manageable Information Base (MIB) files to interpret their contents. It supports SNMP versions 1, 2c, and 3. You can display the results in tree form and sort them in different ways.

You can use the `onesixtyone` command (www.phreedom.org/solar/onesixtyone) to scan SNMP enabled devices to glean information about those servers. The command does this by sending many SNMP requests at a time to many different systems. It then logs the responses as they come in. The `snmpwalk` command (www.net-snmp.org) can retrieve a tree of values under a selected variable from an SNMP MIB object.

Finding Web Server Information

You can glean a lot of information from Web servers, drawing on many different enumeration tools. The WWW menu (select K menu → BackTrack → Enumeration → WWW) includes 11 tools you can use to draw useful information from a Web server.

Using `curl` (http://curl.haxx.se), you can grab files from a Web server (as well as from FTP, TELNET, and gopher servers). Running `curl` with multiple Web locations

(URLs), you can download multiple files over the HTTP protocol from a single command line. Using the `dimitry` command, you can gather a whole lot of information at once about a particular Web server (or other type of system). Besides telling you the system's name and IP address, this command searches a variety of information services to find out about the host, including the `whois` database (Inet and Inic), Netcraft, DNS (to find subdomains), and e-mail service.

The `httprint` command (http://net-square.com/httprint) is a tool for fingerprinting Web servers. Instead of relying on what the Web server reports about itself (which might be masked by modified banner strings and plug-ins), `httprint` uses techniques such the Web server's responses to various requests (such as HTTP `DELETE` and `GET` requests) to determine the type of Web server. Select Httprint GUI to use a graphical version of the `httprint` tool.

To find all the URLS on a Web page, select List URLs to run the `listurls.py` command. Choose Paros Proxy (www.parosproxy.org) to test the security of your Web applications. The Paros graphical utility can intercept and modify HTTP and HTTPS data transferred between the client and server to test for vulnerabilities.

Using Exploit Archives

BackTrack includes several archives of exploits that you can use to test the vulnerability of your systems. It also includes the Metasploit Framework for running exploit modules to test systems for vulnerabilities to selected exploits:

- **SecurityFocus Archive (www.securityfocus.com)**—To access an archive of exploits gathered by Securityfocus.com, select the K menu → Backtrack → SecurityFocus Archive. To update the archive, select Update SecurityFocus from that same menu. This updates and adds exploit information and regenerates the local exploit database. If you have an Internet connection, you can be sure to have the up-to-the-minute information on known exploits.
- **Milw0rm Archive (www.milw0rm.com)**—To access an archive of exploits gathered by Milw0rm.com, select the K menu → Backtrack → Milw0rm Archive. Just as with the SecurityFocus archive, you can update the Milw0rm Archive from the K menu by selecting Update Milw0rm from the Milw0rm menu. This brings up your Milw0rm database to include the latest exploit information from the Milw0rm archives.
- **Metasploit Project (www.metasploit.com)**—The Metasploit Project was set up to help exploit research, as well as assist those who do IDS signature development and penetration testing. In BackTrack, you can access the Metaspolit Web interface from the K menu by selecting Backtrack → Exploit

Archives → Metasploit Framework → Metasploit 2.6 Web-Gui. Other interfaces for using Metasploit are available as well (as described shortly).

By including the Metasploit Framework on the BackTrack live CD, BackTrack enables users to check for vulnerability to known exploits on a variety of systems, including BSD, HPUX, Irix, Linux, OSX, Solaris, Windows 2000, Windows 2003, Windows NT, and Windows XP. You can choose which exploits to run based on application (Apache, Samba, and so on), operating system, or computer architecture (PPC, Sparc, or X86). Figure 9-6 shows the Metasploit Framework Web Console.

The Metasploit Framework Web Console can run as a local service from your Web browser (http://127.0.0.1:55555). By default, you see a list of all available exploits (with updates, BackTrack currently includes 143 exploits and 75 payloads). Figure 9-6 shows the beginning of the list of exploits that appear when Samba was selected. You can see the first three entries in that list, which show the name of the exploit and the operating systems the exploits can be tested against.

FIGURE 9-6 Metasploit lets you choose exploit modules to test for system vulnerabilities.

The advantage of including Metasploit on a live CD is that you can reboot a computer on a LAN with the live CD and test for exploits on that LAN without connecting that LAN to the Internet. You can select each exploit module to see a

description of the vulnerability it attacks. Then links lead you through the process of running each module (selecting options, such as remote systems and port numbers and starting the module).

Besides being a tool for checking exploits, Metasploit is a framework for developing your own security tools. To find out more about how to create and integrate your own tools into Metasploit, a good place to start is the Metasploit documentation page, which includes the project's FAQ: www.metasploit.com/projects/Framework/documentation.html.

Scanning Network Ports

Besides tools such as `nmap` and `nmapfe`, described earlier, you can use several tools to find information about open ports for various protocols (such as TCP and UDP) on remote systems. The `amap` command can scan selected ports on target machines to try to determine what services (if any) are running on the selected ports.

AutoScan (http://autoscan.free.fr) is a graphical tool for exploring and managing your network. When you launch AutoScan, you can select subnetworks to scan in the background. Servers and workstations that are found appear on a list in the left column. You can display available services for each system by selecting that system. You can even start applications to connect to available services (such as `ssh`, `nmap`, and `nessus`). For users who are uncomfortable with Linux shell commands, AutoScan provides an easy-to-use and attractive interface to managing ports and services on your network.

The `pbnj` command (http://pbnj.sourceforge.net) uses the `nmap` and `amap` commands to gather information scanned from ports on selected systems and direct the results to an output file. Using that output file later, you can run the `pbnj` command again to determine changes to the services in the systems and ports you are scanning.

VPN Scanning Tools

Although virtual private networks (VPNs) were designed to provide secure connections over a public network (usually the Internet) between users and remote systems, VPNs can be exploited. Several tools on the VPN Scanners menu (select K menu → BackTrack → Scanners → VPN Scanners) can be used to test the characteristics of a VPN server for potential vulnerabilities.

By selecting Ike-Scan, you can run the `ike-scan` command (www.nta-monitor.com/tools/ike-scan) to try to discover the identity of an active VPN server. By watching for transport characteristics during Internet Key Exchange, `ikescan` can learn about the server being used to set up a connection between a server and

remote client. Using that information, an intruder could try attacking any vulnerabilities that might be known about that VPN service.

Select the PSK-Crack entry (`psk-crack` command) from the menu to use a dictionary file to try to crack a VPN connection during an Internet Key Exchange. By default, the dictionary used is `/usr/local/share/ike-scan/psk-crack-dictionary`. A brute-force cracking technique is also available with the `psk-crack` command (`–bruteforce` option).

Vulnerability Scanning Tools

Network scanners can check for a variety of vulnerabilities. From the Vulnerability Scanners menu (K menu → BackTrack → Scanners → Vulnerability Scanners), you can choose from several different general vulnerability scanners.

The GFI LANguard Network Scanner (www.gfi.com) is a graphical tool for testing a selected system or a range of IP addresses for general information about each computer, specific information about open ports, and alerts related to any of the services that are open on those ports. Figure 9-7 shows an example of the LANguard Network Scanner.

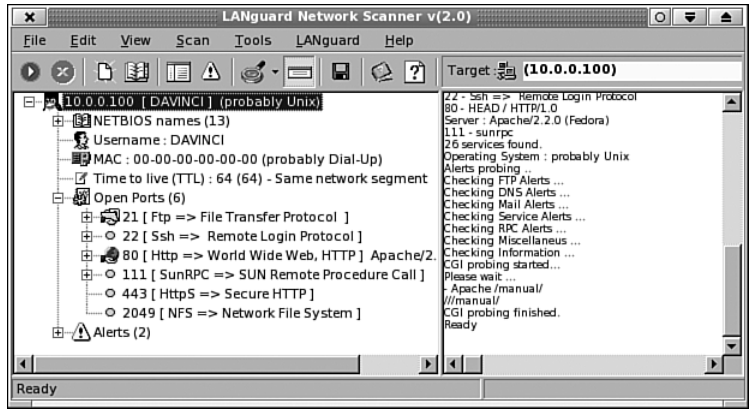

FIGURE 9-7 Scan for a range of vulnerabilities with LANguard Network Scanner.

The example in Figure 9-7 shows the results of scanning a single IP address (10.0.0.100) for a host named DAVINCI. After determining some general information about the systems, such as its NetBIOS names and MAC address, LANguard searches for open ports. When it finds open ports, it tests known vulnerabilities for the active services found on those ports.

In this case, six open ports were found. After the services on those ports were tested, two vulnerabilities were found relating to those ports. You can see the stored

information about the scan on the left, with a running output of messages on the right.

SuperScan is another graphical tool for scanning one or more systems over a network. SuperScan is a Windows executable that can be run in Linux using WINE software. Using SuperScan, you can set up a list of ports you want to scan and set timeouts, check ports, and resolve host names.

Nikto (www.cirt.net) is a command-line vulnerability tester that tests for a range of issues that are most useful for evaluating the vulnerability of Web servers. It incorporates testing features using plug-ins. By running the `nikto` command on a name or IP address for a Web server, Nikto first determines general information about the server (such as operating system, Web server type, host name, and port number). Then it goes on to test the server to determine whether a later version of the server software is available and whether the current version unnecessarily exposes too much information about itself when queried.

Password Attack Tools

Trying to steal the passwords for one user (or a whole lot of users) is one of the oldest techniques for breaking into computer systems. Some password-attack tools are designed to be run on live (online) systems, and others are meant to be run offline (for example, when a password file is stolen and the cracker can run the attack at leisure).

Because BackTrack includes so many tools for cracking passwords, I focus on only a few of them. When you open the Password Attacks menu (K menu → BackTrack → Password Attacks), most of the password attacks are divided into offline and online attacks.

Offline Password Attacks

If you have access to a computer's password file (typically `/etc/passwd`) and shadow file (typically `/etc/shadow`), you can run commands on those files to try to crack the passwords they contain. As someone trying to protect your own systems, you can use these commands to uncover weak passwords entered by those who use your computers.

Check the Offline Attacks menu (K menu → BackTrack → Password Attacks → Offline Attacks) to see a list of password attack commands available with BackTrack. One example of such a command is `john`. Create a combined password and shadow file. Then run `john` on that file:

```
# unshadow /etc/passwd /etc/shadow > mypasswd
# john mypasswd
Created directory: root/.john
```

```
Loaded 3 password hashes with 3 different salts (FreeBSD MD5 [32/32])
jake     (jake)
toor     (root)
```

In this example, the password for the user jake was discovered to simply be `jake`. The password for root was found to be `toor`. If you were able to crack users' passwords on a running system, you should tell those users to change their passwords. Be sure to remove the `mypasswd` file and `.john` directory when you are done checking passwords.

Other offline password cracking tools include Rainbow Crack tools `BKHive`, `OPHCrack`, `Rcrack`, `Rtdump`, `Rtsort`, `SAMDump2`, and `WebCrack`. `RainbowCrack` uses prebuilt rainbow tables to speed password cracking. (See www.antsight.com/zsl/rainbowcrack site for more information on the RainbowCrack project.)

Online Password Attacks

Online password attacks can be more difficult to carry out. Repeated brute force attacks can easily be detected. A system can be configured to simply prevent someone from repeatedly trying and failing to login to a system.

Hydra (www.thc.segfault.net) is a paralyzed login hacking program. It can be used to try to guess login/password pairs using more than a dozen different service types (telnet, ftp, pop3, imap, and so on). CowPatty is a brute-force password crackdr made for cracking WIFI login/password pairs. Medusa (www.foofus.net) is another parallel password cracking tool, including modules for cracking passwords in CVS, FTP, HTTP, IMAP, MySQL, and other services.

Running Fuzzers

A fuzzer is a program that injects input into an application, hoping that, as a result, the application will crash or cause an exception. By getting programs to crash, the fuzzer hopes to find vulnerabilities in the system that can be exploited by such things as buffer overflows and denial-of-service attacks. You can find fuzzer applications on the Fuzzers menu (select K menu → Backtrack → Fuzzers).

The Bruteforce Exploit Detector (`bed` command) sends commands to server daemons in an attempt to get those commands to cause the daemons to crash. This can help determine whether the server is susceptible to buffer overflow attacks. The CIRT Fuzzer (`fuzz.pl` command) lets you identify a host and port to attack. You can include a `template.txt` file on the `fuzz.pl` command line.

You can use the `clfuzz.py` program to help audit binaries that have `setuid` set. The `fuzzer.py` program can help find buffer overflow and SQL injection.

BackTrack also comes with the SPIKE Fuzzer Creation Kit. Tools that come with this kit include `webfuzz` (a combination of small tools for Web application

fuzzing) and `msrpcfuzz` (used to send random arguments with the intent of finding bugs that will cause a port to close down).

Doing Spoofing

One way to break into networks or systems is to run programs in which the program pretends to be something it's not, such as a legitimate DNS server or DHCP server, in hopes of fooling an unsuspecting application or user into giving up critical information. BackTrack includes a whole set of spoofing tools (K menu → Backtrack → Spoofing) that you can try out to test for vulnerabilities in your systems.

The `arpspoof` program (part of the `dsniff` package from www.monkey.org/~dugsong/dsniff) tries to trick another computer into believing that it is the gateway machine. It does this by constantly sending the victim machine its own IP address as that of the gateway. When the victim machine eventually caches the `arpspoof` machine's MAC address as that of the gateway, the victim starts sending its IP packets to the `arpspoof` machine instead of the real gateway.

The `dhcpx` program (www.phenoelit.de/irpas) is a DHCP flooder. This technique is also referred to as a DHCP exhaustion attack. The `dnsspoof` command tries to convince other systems on the LAN that it is the DNS server by faking replies to various DNS address queries and pointer queries.

Using the `fragroute` command, you can try to get, change, and rewrite traffic in the LAN that is destined for locations outside the LAN. This type of network traffic is referred to as egress traffic. Although `fragroute` has an effect on only packets that originate on the local machine, the `fragrouter` command can work on other machines as well.

Check the Spoofing menu for other types of spoofing you can do as well. For example, some commands try to redirect ICMP traffic, and others try to gain control by replaying TCP/IP packets.

Network Sniffers

As its name implies, a network sniffer watches a network, trying to "sniff" for information being passed on that network. That information can be used for evil (in the case of someone trying to steal data or connection information) or good (to uncover a potential problem on the network). In some cases, employers use sniffers to make sure employees aren't doing anything against company policy with their computers.

BackTrack includes more than a dozen sniffers that you can use to watch network traffic of various kinds. Check the Sniffers menu to see which ones BackTrack includes (K menu → BackTrack → Sniffers).

You can use the AIM Sniffer (`aimsniffer` command) to log and capture traffic from AOL Instant Messenger sessions. Driftnet (www.ex-parrot.com/~chris/driftnet) is a graphical utility that listens for images in TCP traffic and displays those it finds in the Driftnet window. Figure 9-8 shows the images that appeared on the Driftnet window after selecting Wikipedia.org from a local Web browser.

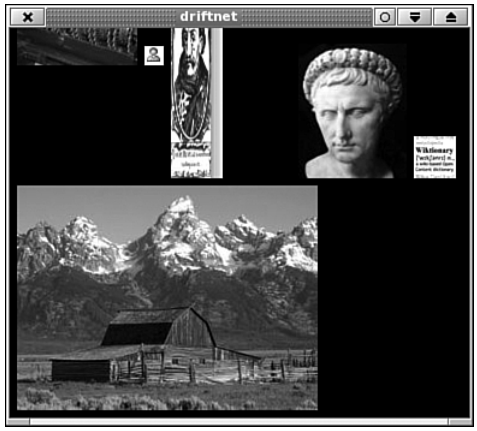

FIGURE 9-8
Display images encountered in TCP traffic with `driftnet`.

The `dsniff` command (http://monkey.org/~dugong/dsniff) is a password-sniffing command that can listen for passwords on a variety of services. Those services include FTP, TELNET, SMTP, HTTP, POP, X11, AIM, ICQ, and others. With `dsniff` running, it automatically detects the protocols it is sniffing, picks out only relevant password information, and saves that information to an output file in Berkeley DB format.

Ethereal (www.ethereal.com) is a graphical tool for capturing Ethernet traffic on selected interfaces. The Ethereal window contains tools for filtering, sorting, and analyzing network traffic. Ethereal also allows the use of plug-ins, which can be used for many purposes, including outputting the data gathered by Ethereal in various ways.

Iptraf (http://iptraf.seul.org) is an IP network statistics utility you can use to monitor the traffic on your network. The utility is `ncurses`-based, so it can run in any shell (no need for a GUI). Using the `iptraf` command, you can watch for specific types of traffic or watch only on selected interfaces. Log information is placed in `/var/log/iptraf` unless otherwise specified.

You can use other specialized sniffers in addition to those just mentioned. The SMB Sniffer (`start-smb-sniffer` command) watches for traffic relating to Windows file- and printer-sharing activity. The `mailsnarf` command captures mail traffic based on interfaces you choose and specific patterns you are looking for. The `PHoss` utility (www.phenoelit.de/phoss) watches network interfaces for clear-text passwords.

To sniff secure shell (SSH) traffic, the `sshmitm` utility (part of the `dsniff` package) can be placed between an `ssh` client and server. By acting as the real `ssh` server, `sshmitm` can grab the password information from an unsuspecting client and then complete the transaction for the client with the real server without the client suspecting that anything has gone wrong. The `ettercap` utility (http://ettercap.sourceforge.net) is a graphical utility that can similarly do man-in-the-middle attacks, but `ettercap` can grab whole sessions as well as the password information.

As the name implies, URLsnarf (`urlsnarf` command) can grab Web address information requested on a selected interface (matching any pattern you like). XSpy (`xspy` command) can track everything typed into an X display and echo that information back to the shell where `xspy` is running.

To save files that are transferred using the NFS protocol, you can use the `filesnarf` command. Using the `msgsnarf` command, you can save messages and chat sessions from many different IRC and messaging protocols. With `mailsnarf`, you can sniff SMTP and POP traffic to save e-mail messages. All three, `mailsnarf`, `filesnarf`, and `msgsnarf`, are part of the `dsniff` package.

Wireless Network Tools

You can use tools for working with wireless networks to try to analyze traffic, crack passwords, forge packets, and switch to different driver configurations on your wireless LAN cards. You can see tools available in BackTrack for working with wireless LANs on the Wireless Tools menu (K menu → BackTrack → Wireless Tools).

Kismet (www.kismetwireless.net) is a popular tool for working with wireless LAN interfaces. With Kismet, you can detect intrusions on 802.11 wireless networks (802.11b, 802.11a, and 802.11g). Kismet collects packets passively to detect hidden networks. It detects networks using standard names and can decloak hidden networks over time. Data you log with Kismet is compatible with Ethereal and `tcpdump` logs.

To test password strength of the nodes on a wireless LAN, you can use tools such as WepAttack (http://wepattack.sourceforge.net), Wep_crack and Wep_decrypt (www.lava.net/~newsham/wlan, as part of the `wep_tools` package), and Weplab (http://weeplab.sourceforge.net). Each of these tools provides a different set of options for testing the security of the passwords used on your wireless networks.

Some miscellaneous wireless tools are available from the Miscellaneous submenu. The `macchanger` utility (www.alobbs.com/macchanger) lets you view and change the MAC addresses of network interfaces. If you have a GPS device

attached to your system, you can start and stop the GPS tracking daemon by selecting Start GPS Daemon or Stop GPS Daemon, respectively. Several utilities also exist for changing the drivers used on your wireless cards.

For forging packets on wireless networks, a handful of tools are available on the Packet Forging menu. The `aircrack-ng` package (www.aircrack0-ng.org/doku.php) contains a set of tools for auditing the reliability of your wireless network, including `aireplay-ng` (for injecting frames that generate traffic for cracking WEP and WPA-PSK keys). Airsnarf (www.shmoo.com) can be used to set up a rogue wireless access point. WifiTap (`wifitap` command) can be used to capture traffic and inject packets on an 802.11 network (see http://sid.rstack.org/index.php/wifitap_EN).

BlueTooth Device Tools

Tools for monitoring BlueTooth devices (www.bluetooth.org) are available from the BlueTooth menu (K menu → BackTrack → BlueTooth). Blue Snarfer (`bluesnarfer` command) downloads the phonebook of Bluetooth-enabled devices that are not secured against such intrusions. Carwhisperer (`carwhisperer` command) tests the vulnerability of BlueTooth car kits, particularly those that use standard passkeys (see http://trifinite.org/trifinite_stuff_carwhisperer.html).

You can use the HeloMoto utility (`helomoto` command) to exploit the handling of trusted devices by some Motorola phones and other devices. You can learn about HeloMoto attacks from the Trifinite.org site (http://trifinite.org/trifinite_stuff_helomoto.html).

Cisco Router Tools

Look under the Cisco Tools menu (K menu → BackTrack → Cisco Tools) to find tools for checking for vulnerabilities in Cisco Routers. The Cisco Global Exploiter (`cge.pl perl` script) can test for a range of known Cisco equipment vulnerabilities. Using this tool usually just requires a target and the name of an exploit to test for. You can find out about this script from the BlackAngels site (www.blackangels.it).

Cisco Torch is a mass Cisco vulnerability scanner (`cisco-torch.pl` Perl script). With it, you can scan for a variety of services on Cisco devices, including `telnetd`, `sshd`, `snmp`, and `tftp`. You can do fingerprinting scans on Network Time Protocol (NTP) and TFTP servers.

Yersinia (`yersinia` command) is a tool for analyzing network protocols, including the Cisco Discovery Protocol (CDP). You can find out more about Yersinia from its home page (www.yersinia.net). The `cisco-copy-config.pl` and `merge-copy-config.pl` Perl scripts are simple scripts for copying and merging Cisco configuration scripts, respectively, using SNMP.

Database-Analysis Tools

BackTrack divides its tools for analyzing database vulnerabilities into categories of generic database tools, MS-Sql, Mysql, and Oracle. Nearly a dozen tools exist for checking database vulnerabilities.

Absinthe (www.0x90.org/releases/absinthe) is a graphical tool for downloading database schema and content that might be vulnerable to blind SQL injection. It also includes the basics for MS SQL server error-based injection. The tool is used primarily to improve the speed of recovering data.

The SQL-Dict selection runs the `sqldict.exe` Windows binary for launching the SQL Server Dictionary Attacker (http://ntsecurity.nu/toolbox). The window lets you start attacks to test for SQL dictionary vulnerabilities based on target server IP address and target account. You can also load a password file.

For MySQL databases, BackTrack includes Blind SQL Injection (`bsqlbf.pl` script from www.unsec.net), SQL-Miner (`mysql-miner.pl` script), and Setup MySQL (`setup-mysql` Python script). The `setup-mysql` script lets you configure and start a MySQL server on which you can start testing.

Forensic Tools

To check out a system for problems after they occur, BackTrack includes a handful of forensic tools you can run on the system. The Autopsy Forensic Browser (www.sleuthkit.org/autopsy) lets you set up cases to do volume and file-system analysis. As forensics are run, data is stored in an Evidence Locker that is associated with a case.

Using the `pasco` command (www.foundstone.com/resources/proddesc/pasco.htm), you can browse the contents of cache files that Internet Explorer left behind. It takes an `index.dat` file as input and outputs field-delimited data that you can use in a spreadsheet.

Acquiring Tools

To be able to run forensic analysis tools on an infected system, you usually need to copy the file system to an image file to work on it from another location. The standard tool for copying an image file, and possibly converting it in various ways, is the `dd` command (which has been around since the old UNIX days). BackTrack includes several related commands that can also work with those file system images.

The `dcfldd` command is an enhanced version of the `dd` command that is intended specifically for doing forensics. This command can do such things as

hashing on-the-fly (to ensure data integrity) and performing flexible disk wipes (using any known pattern you choose). To learn more about `dcfldd`, refer to its SourceForge page (http://dcfldd.sourceforge.net).

The `ddrescue` command (www.gnu.org/software/ddrescue/ddrescue.html) is a data-recovery tool that does its best to rescue data in environments where there might be read errors. Because `ddrescue` runs automatically, it fills in data gaps instead of truncating output or stopping for errors.

A graphical tool for creating file-system image files, already discussed in this chapter, is the Automated Image & Restore (AIR) utility.

Running Services

Because network interfaces and services that run on those interfaces are off by default in BackTrack, the BackTrack Services menu (K menu → BackTrack → BackTrack Services) lets you select to start and stop various system services that can help you with rescue and security activities. You can select the following services from this menu:

- **HTTPD**—Starts and stops an Apache Web server
- **Postgres**—Starts and stops a Postgresql database server
- **SNORT**—Sets up and initializes Snort, and then starts Snort (along with any dependent processes)
- **SSH**—Sets up, starts, and stops an SSH remote-login daemon server
- **TFTPD**—Starts and stops the trivial FTP server
- **VNC Server**—Starts and stops a virtual network computing (VNC) server to allow remote displaying of local desktop displays
- **vsFTPD**—Starts and stops an FTP server

Depending on the types of services you want to be able to test from your live Linux CD, you might want to add other services as well.

Miscellaneous Services

A whole bunch of services that didn't fit into any other category are lumped into the Misc category on the BackTrack menu. Among the tools on that menu are QTParted (a graphical disk-partitioning tool based on the `parted` command), PDF Viewer (for viewing PDF files with the `kpdf` viewer), and Screen Capture (to take a screen shot with the `ksnapshot` command).

Adding Software to a BackTrack Live CD

Using the customization and remastering techniques described in Chapters 6, "Building a Custom Knoppix Live CD," through 8, "Building a Basic Gentoo Live CD," for Knoppix, Fedora, and Gentoo live CDs, you can add the components described in this chapter (as well as other components) as you choose. In some cases, software packages will already be available for your chosen live CD type (Deb, RPM, or Emerge packages). In other cases, you can download the software you want from the project Web sites mentioned with many of the components described.

Because BackTrack is based on a slackware (in particular, SLAX live CD) distribution, I next describe how to add SLAX-based packages to a BackTrack live CD. This way, you can download and run any packages (referred to a modules) available from the SLAX software repository.

The technique for adding software modules to a SLAX-based live CD such as BackTrack is much easier than the technique for remastering Knoppix (using a `chroot` environment) in Chapter 6. You can add SLAX modules to your BackTrack live CD in two ways:

- **Remaster**—Add a SLAX module to the `/modules` directory on the live CD. You copy the contents of your BackTrack live CD to a directory on your hard disk, adding the SLAX module to the modules directory, and re-creating the ISO image (using `mkisofs` command); the software in the module you added is installed when you boot that live CD.
- **Live**—You can download any SLAX module and install it live using the `uselivemod` command while BackTrack is running. The software in that module then is immediately available for use.

To find available software modules that will work with BackTrack and other SLAX-based live CDs, go to the SLAX modules page (www.slax.org/modules.php). There you will find 15 categories of software you can add to your running live CD or to the `/modules` directory when you remaster. To install a module immediately, find the software you want (either select the category or use the search box to find what you want). Then download the module you choose.

With the module you want to install in the current directory, you can use the `uselivemod` command to install the module. For example, you could type the following to install the `eMovix` package:

```
# uselivemod eMovix_0_9_0.mo
```

The software will be immediately available on your live CD to use.

You might need a variety of drivers and firmware to get certain hardware peripherals working properly on your live CD. SLAX comes with a bunch of drivers that might not have been included with the distribution but that might be legal for you to install on your personal live CD:

From the SLAX modules page, select Drivers. These are some examples of drivers you might want to add (depending on your hardware) that are available from the modules page:

- **ATI video drivers**—If your video card uses ATI video chipsets, you can install closed-source, proprietary drivers from the manufacturer that might work better for applications that require hardware acceleration than their open-source counterparts.
- **NVIDIA drivers**—As with ATI video cards, if you have NVIDIA chipsets in your video cards, proprietary drivers from NVIDIA will probably work better for you than open-source drivers for gaming and other software that performs better with hardware acceleration.
- **madwifi**—Several different wireless LAN cards might not work at all in Linux without drivers from the madwifi project.
- **Linux Webcam driver**—The drivers in this package can help the limited number of Webcams that are supported in Linux to work properly.

Besides those packages just mentioned, other drivers are available from this page. They include modem drivers for specific modem types and some firmware packages that are needed to get certain wireless cards working properly.

SUMMARY

With dozens of security- and rescue-oriented live Linux CDs available today, choosing one for yourself can be done based on the tools that are provided and the Linux distribution it is based on. For examples in this chapter, I used the BackTrack network security suite live CD.

Using BackTrack, you can evaluate literally hundreds of security- and rescue-oriented tools to work on Linux systems, networks, and a variety of systems you can reach from a network. BackTrack is based on SLAX (which is based on Slackware) and, therefore, can use many of the resources available to SLAX (such as extra software modules).

The chapter steps you through some of the design decisions in BackTrack and notes where you have opportunities to modify how BackTracks boots up and starts services. Because BackTrack uses KDE as its desktop, you have a wide range of desktop tools to support specialized security utilities that will run from that platform.