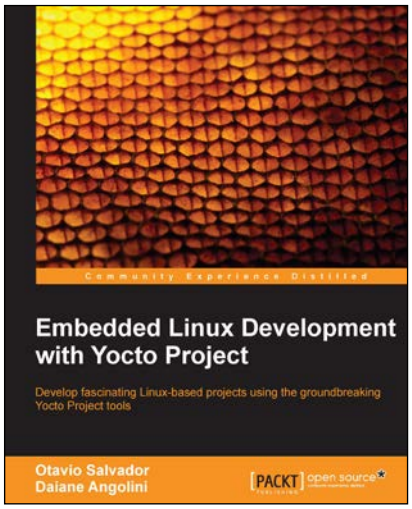
[PACKT] open source
community experience distilled
PUBLISHING

Embedded Linux Development with Yocto Project

Otavio Salvador

Daiane Angolini



Chapter No. 1

"Meeting the Yocto Project"

In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter NO.1 "Meeting the Yocto Project"

A synopsis of the book's content

Information on where to buy this book

About the Authors

Otavio Salvador loves to play video games and started his free software activities in 1999. In 2002, he founded O.S. Systems, a company focused on embedded system development services and consultancy worldwide, creating and maintaining customized BSPs and helping companies with their release management challenges. This resulted in him joining the OpenEmbedded community in 2008, when he became an active contributor to the OpenEmbedded project, culminating in his attribution as the maintainer of the Freescale ARM BSP layer in the Yocto Project in 2011.

Daiane Angolini has been focusing on embedded technologies for the past 8 years. Since 2008, she has been working on Freescale Semiconductors as an application engineer, on internal development and porting custom applications from Android to Freescale architectures, and on customer support for ARM processors of the i.MX family, while also participating in Freescale forums. She has been working with the Yocto Project tools through meta-fsl-arm, the BSP meta layer that provides board support for Freescale ARM machines, since 2012. The desire to become an expert in ice cream making has been keeping her busy in her spare time for the past year.

We initially want to thank our families. They provided lovely support and helped us to get on track for this project.

This project has only been possible because we had support from many people who provided insights, reviews, material, and guidance during the full period of conception and production of this book. We'd like to give special thanks to (in alphabetic order): Alex González, Alexandru Vaduva, Harsha Bharwani, Jeffrey Osier-Mixon, John Weber, Manan Badani, Paul Eggleton, Rogerio Nunes, Radek Dostál, Sageer Parkar, and Sankalp Pawar

- Otavio Salvador and Daiane Angolini

Embedded Linux Development with Yocto Project

Considering the current technology trend, Linux is the next big thing. Linux has consistently released cutting-edge open source products, and embedded systems have been added to the technological portfolio of mankind.

The Yocto Project is in an optimal position to be the choice for your projects; it provides a rich set of tools to help you to use most of your energy and resources in your product development, instead of reinventing the wheel.

The usual tasks and requirements for embedded Linux-based products and development teams were the guidelines for this book's conception. Written by active community members with a practical and straightforward approach, it is a stepping stone for both your learning curve and your product's project.

What This Book Covers

Chapter 1, Meeting the Yocto Project, presents the history of the Yocto Project, showing the parts that compose it.

Chapter 2, Baking Our Poky-based System, introduces the environment needed for the first build.

Chapter 3, Using Hob to Bake an Image, shows the user-friendly graphical interface that can be used as a wrapper for configuration and as a build tool.

Chapter 4, Grasping the BitBake Tool, presents the first concepts and premises of the tool used to control all other pieces of the Yocto Project.

Chapter 5, Detailing the Temporary Build Directory, details the output directory tree of a build with focus on the tmp directory.

Chapter 6, Assimilating Packaging Support, introduces the package concepts and details the packaging support used by the Yocto Project.

Chapter 7, Diving into BitBake Metadata, details the concepts and syntaxes used by the Yocto Project metadata, both in recipes and configuration files.

Chapter 8, Developing with the Yocto Project, details how to use the Yocto Project to generate a custom development environment.

Chapter 9, Debugging with the Yocto Project, details which debug tools the Yocto Project provides and how to use them.

Chapter 10, Exploring External Layers, explores one of the most important concepts of the Yocto Project, which is the flexibility of using external layers.

Chapter 11, Creating Custom Layers, practices the steps of creation of layers.

Chapter 12, Customizing Existing Recipes, lists the common use cases of recipe customization and how to achieve them properly.

Chapter 13, Achieving GPL Compliance, summarizes the tasks and concepts involved in a copyleft compliance product.

Chapter 14, Booting Our Custom Embedded Linux, uses a real hardware machine together with the Yocto Project's tools.

Appendix, References, lists the references used in the book.

1
Meeting the Yocto Project

In this chapter, we will be introduced to the **Yocto Project**. The main concepts of the project, which are constantly used throughout the book, are discussed here. We will discuss the Yocto Project history, OpenEmbedded, Poky, BitBake, and Metadata in brief, so fasten your seat belt and welcome aboard!

What is the Yocto Project?

The Yocto Project is a Linux Foundation workgroup defined as:

> *"The Yocto Project provides open source, high-quality infrastructure and tools to help developers create their own custom Linux distributions for any hardware architecture, across multiple market segments. The Yocto Project is intended to provide a helpful starting point for developers."*

The Yocto Project is an open source collaboration project that provides templates, tools, and methods to help us create custom Linux-based systems for embedded products regardless of the hardware architecture. Being managed by a Linux Foundation fellow, the project remains independent of its member organizations that participate in various ways and provide resources to the project.

It was founded in 2010 as a collaboration of many hardware manufacturers, open source operating systems, vendors, and electronics companies in an effort to reduce their work duplication, providing resources and information catering to both new and experienced users.

Among these resources is OpenEmbedded-Core, the core system component, provided by the OpenEmbedded project.

The Yocto Project is, therefore, a community open source project that aggregates several companies, communities, projects, and tools, gathering people with the same purpose to build a Linux-based embedded product; all these components are in the same boat, being driven by its community needs to work together.

Delineating the Yocto Project

To ease our understanding of the duties and outcomes provided by the Yocto Project, we can use the analogy of a computing machine. The input is a set of data that describes what we want, that is, our specification. As an output, we have the desired Linux-based embedded product.

If the output is a product running a Linux-based operating system, the result generated is the pieces that compose the operating system, such as the Linux kernel, bootloader, and the root filesystem (`rootfs`) bundle, which are properly organized.

To produce the resultant `rootfs` bundle and other deliverables, the Yocto Project's tools are present in all intermediary steps. The reuse of previously built utilities and other software components are maximized while building other applications, libraries, and any other software components in the right order and with the desired configuration, including the fetching of the required source code from their respective repositories such as The Linux Kernel Archives (`www.kernel.org`), GitHub, and `www.SourceForge.net`.

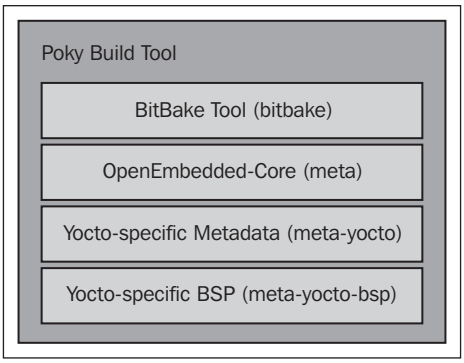
Preparing its own build environment, utilities, and toolchain, the amount of host software dependency is reduced, but a more important implication is that the determinism is considerably increased. The utilities, versions, and configuration options are the same, minimizing the number of host utilities to rely on.

We can list some projects, such as Poky, BitBake, and OpenEmbedded-Core, under the Yocto Project umbrella, all of them being complimentary and playing specific roles in the system. We will understand exactly how they work together in this chapter and throughout the book.

Understanding Poky

Poky is the Yocto Project reference system and is composed of a collection of tools and metadata. It is platform-independent and performs cross-compiling, using the **BitBake** tool, OpenEmbedded Core, and a default set of metadata, as shown in the following figure. It provides the mechanism to build and combine thousands of distributed open source projects to form a fully customizable, complete, and coherent Linux software stack.

Poky's main objective is to provide all the features an embedded developer needs.



Using BitBake

BitBake is a task scheduler that parses Python and Shell Script mixed code. The code parsed generates and runs tasks, which are basically a set of steps ordered according to the code's dependencies.

It evaluates all available configuration files and recipe data (known as **metadata**), managing dynamic variable expansion, dependencies, and code generation. It keeps track of all tasks being processed in order to ensure completion, maximizing the use of processing resources to reduce build time and being predictable. The development of BitBake is centralized in the `bitbake-devel@lists.openembedded.org` mailing list, and its code can be found in the `bitbake` subdirectory of Poky.

OpenEmbedded-Core

The **OpenEmbedded-Core** metadata collection provides the engine of the Poky build tool. It is designed to provide the core features and needs to be as clean as possible. It provides support for five different processor architectures (**ARM**, **x86**, **x86-64**, **PowerPC**, **MIPS** and **MIPS64**), supporting only QEMU-emulated machines.

The development is centralized in the `openembedded-core@lists.openembedded.org` mailing list, and houses its metadata inside the `meta` subdirectory of Poky.

Metadata

The metadata, which is composed of a mix of Python and Shell Script text files, provides a tremendously flexible system. Poky uses this to extend OpenEmbedded-Core and includes two different layers, which are another metadata subset shown as follows:

- `meta-yocto`: This layer provides the default and supported distributions, visual branding, and metadata tracking information (maintainers, upstream status, and so on)
- `meta-yocto-bsp`: This layer, on top of it, provides the hardware reference boards support for use in Poky

Chapter 7, *Diving into BitBake Metadata*, explores the metadata in more detail and serves as a reference when we write our own recipes.

The alliance of OpenEmbedded Project and Yocto Project

The **OpenEmbedded** project was created around January 2003 when some core developers from the **OpenZaurus** project started to work with the new build system. The OpenEmbedded build system has been, since its beginning, a tasks scheduler inspired and based on the **Gentoo Portage** package system named BitBake. The project has grown its software collection, and a number of supported machines at a fast pace.

As consequence of uncoordinated development, it is difficult to use OpenEmbedded in products that demand a more stable and polished code base, which is why Poky was born. Poky started as a subset of OpenEmbedded and had a more polished and stable code base across a limited set of architectures. This reduced size allowed Poky to start to develop highlighting technologies, such as IDE plugins and QEMU integration, which are still being used today.

Around November 2010, the Yocto Project was announced by the Linux Foundation to continue this work under a Linux Foundation-sponsored project. The Yocto Project and OpenEmbedded Project consolidated their efforts on a core build system called OpenEmbedded-Core, using the best of both Poky and OpenEmbedded, emphasizing an increased use of additional components, metadata, and subsets.

Summary

This first chapter provided an overview on how the OpenEmbedded Project is related to the Yocto Project, the components which form Poky, and how it was created. In the next chapter, we will be introduced to the Poky workflow with steps to download, configure, and prepare the Poky build environment, and how to have the very first image built and running using QEMU.

Where to buy this book

You can buy Embedded Linux Development with Yocto Project from the Packt Publishing website: `http://www.packtpub.com/embedded-linux-development-with-yocto-project/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

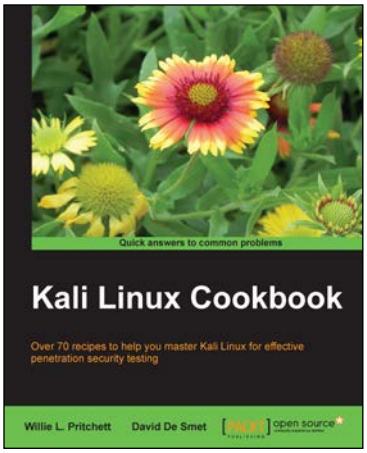
www.PacktPub.com

For More Information:
www.packtpub.com/embedded-linux-development-with-yocto-project/book

[PACKT] open source
community experience distilled
PUBLISHING

Kali Linux Cookbook

Willie L. Pritchett
David De Smet



Chapter No. 9
"Wireless Attacks"

In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter NO.9 "Wireless Attacks"

A synopsis of the book’s content

Information on where to buy this book

About the Authors

Willie L. Pritchett has a Master's in Business Administration. He is a seasoned developer and security enthusiast who has over 20 years of experience in the IT field. He is currently the Chief Executive at Mega Input Data Services, Inc., a full service database management firm specializing in secure, data-driven, application development, and staffing services. He has worked with state and local government agencies as well as helping many small businesses reach their goals through technology. Willie has several industry certifications and currently trains students on various topics including ethical hacking and penetration testing.

I would like to thank my wife Shavon for being by my side and supporting me as I undertook this endeavor. To my children, Sierra and Josiah, for helping me to understand the meaning of quality time. To my parents, Willie and Sarah, I thank you for providing a work ethic and core set of values that guide me through the roughest days. A special thanks to all of my colleagues, associates, and business partners who gave me a chance when I first started in the IT field; through you a vision of business ownership wasn't destroyed, but allowed to flourish. Finally, I would like to thank all of the reviewers and technical consultants who provided exceptional insight and feedback throughout the course of writing this book.

David De Smet has worked in the software industry since 2007 and is the founder and CEO of iSoftDev Co., where he is responsible for many varying tasks, including but not limited to consultant, customer requirements specification analysis, software design, software implementation, software testing, software maintenance, database development, and web design. He is so passionate about what he does that he spends inordinate amounts of time in the software development area. He also has a keen interest in the hacking and network security field and provides network security assessments to several companies.

I would like to extend my thanks to Usha Iyer for giving me the opportunity to get involved in this book, as well as my project coordinator Sai Gamare and the whole team behind the book. I thank my family and especially my girlfriend Paola Janahaní for the support, encouragement, and most importantly the patience while I was working on the book in the middle of the night.

Kali Linux Cookbook

Kali Linux is a Linux-based penetration testing arsenal that aids security professionals in performing assessments in a purely native environment dedicated to hacking. Kali Linux is a distribution based on the Debian GNU/Linux distribution aimed at digital forensics and penetration testing use. It is a successor to the popular BackTrack distribution.

Kali Linux Cookbook provides you with practical recipes featuring many popular tools that cover the basics of a penetration test: information gathering, vulnerability identification, exploitation, privilege escalation, and covering your tracks.

The book begins by covering the installation of Kali Linux and setting up a virtual environment to perform your tests. We then explore recipes involving the basic principles of a penetration test such as information gathering, vulnerability identification, and exploitation. You will learn about privilege escalation, radio network analysis, voice over IP, password cracking, and Kali Linux forensics.

Kali Linux Cookbook will serve as an excellent source of information for the security professional and novice alike. The book offers detailed descriptions and example recipes that allow you to quickly get up to speed on both Kali Linux and its usage in the penetration testing field.

We hope you enjoy reading the book!

What This Book Covers

Chapter 1, Up and Running with Kali Linux, shows you how to set up Kali Linux in your testing environment and configure Kali Linux to work within your network.

Chapter 2, Customizing Kali Linux, walks you through installing and configuring drivers for some of the popular video and wireless cards.

Chapter 3, Advanced Testing Lab, covers tools that can be used to set up more advanced simulations and test cases.

Chapter 4, Information Gathering, covers tools that can be used during the information gathering phase including Maltego and Nmap.

Chapter 5, Vulnerability Assessment, walks you through the usage of the Nessus and OpenVAS vulnerability scanners.

Chapter 6, Exploiting Vulnerabilities, covers the use of Metasploit through attacks on commonly used services.

Chapter 7, Escalating Privileges, explains the usage of tools such as Ettercap, SET, and Meterpreter.

Chapter 8, Password Attacks, walks you through the use of tools to crack password hashes and user accounts.

Chapter 9, Wireless Attacks, walks you through how to use various tools to exploit the wireless network.

9
Wireless Attacks

In this chapter, we will cover:

- Wireless network WEP cracking
- Wireless network WPA/WPA2 cracking
- Automating wireless network cracking
- Accessing clients using a fake AP
- URL traffic manipulation
- Port redirection
- Sniffing network traffic

Introduction

These days, wireless networks are everywhere. With users being on the go like never before, having to remain stationary because of having to plug into an Ethernet cable to gain Internet access is not feasible. For this convenience, there is a price to be paid; wireless connections are not as secure as Ethernet connections. In this chapter, we will explore various methods for manipulating radio network traffic including mobile phones and wireless networks.

Wireless network WEP cracking

Wireless Equivalent Privacy, or **WEP** as it's commonly referred to, has been around since 1999 and is an older security standard that was used to secure wireless networks. In 2003, WEP was replaced by WPA and later by WPA2. Due to having more secure protocols available, WEP encryption is rarely used. As a matter of fact, it is *highly* recommended that you never use WEP encryption to secure your network! There are many known ways to exploit WEP encryption and we will explore one of those ways in this recipe.

In this recipe, we will use the AirCrack suite to crack a WEP key. The AirCrack suite (or AirCrack NG as it's commonly referred to) is a WEP and WPA key cracking program that captures network packets, analyzes them, and uses this data to crack the WEP key.

Getting ready

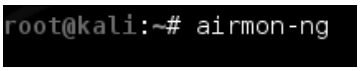
In order to perform the tasks of this recipe, experience with the Kali terminal window is required. A supported wireless card configured for packet injection will also be required. In case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties. Please ensure your wireless card allows for packet injection as this is not something that all wireless cards support.

How to do it...

Let's begin the process of using AirCrack to crack a network session secured by WEP.

1. Open a terminal window and bring up a list of wireless network interfaces:

   ```
   airmon-ng
   ```

2. Under the `interface` column, select one of your interfaces. In this case, we will use `wlan0`. If you have a different interface, such as `mon0`, please substitute it at every location where `wlan0` is mentioned.
3. Next, we need to stop the `wlan0` interface and take it down so that we can change our MAC address in the next step.

   ```
   airmon-ng stop
   ifconfig wlan0 down
   ```

4. Next, we need to change the MAC address of our interface. Since the MAC address of your machine identifies you on any network, changing the identity of our machine allows us to keep our true MAC address hidden. In this case, we will use `00:11:22:33:44:55`.

   ```
   macchanger --mac 00:11:22:33:44:55 wlan0
   ```

5. Now we need to restart `airmon-ng`.

   ```
   airmon-ng start wlan0
   ```

6. Next, we will use `airodump` to locate the available wireless networks nearby.

   ```
   airodump-ng wlan0
   ```

7. A listing of available networks will begin to appear. Once you find the one you want to attack, press *Ctrl* + *C* to stop the search. Highlight the MAC address in the `BSSID` column, right click your mouse, and select copy. Also, make note of the channel that the network is transmitting its signal upon. You will find this information in the `Channel` column. In this case, the channel is `10`.

8. Now we run `airodump` and copy the information for the selected BSSID to a file. We will utilize the following options:

   - `–c` allows us to select our channel. In this case, we use `10`.
   - `–w` allows us to select the name of our file. In this case, we have chosen `wirelessattack`.
   - `–bssid` allows us to select our BSSID. In this case, we will paste `09:AC:90:AB:78` from the clipboard.

   ```
   airodump-ng –c 10 –w wirelessattack --bssid 09:AC:90:AB:78 wlan0
   ```

9. A new terminal window will open displaying the output from the previous command. Leave this window open.

10. Open another terminal window; to attempt to make an association, we will run `aireplay`, which has the following syntax: `aireplay-ng -1 0 –a [BSSID] –h [our chosen MAC address] –e [ESSID] [Interface]`

    ```
    aireplay-ng -1 0 -a 09:AC:90:AB:78 –h 00:11:22:33:44:55 –e
    backtrack wlan0
    ```

11. Next, we send some traffic to the router so that we have some data to capture. We use `aireplay` again in the following format: `aireplay-ng -3 –b [BSSID] – h [Our chosen MAC address] [Interface]`

    ```
    aireplay-ng -3 –b 09:AC:90:AB:78 –h 00:11:22:33:44:55 wlan0
    ```

12. Your screen will begin to fill with traffic. Let this process run for a minute or two until we have information to run the crack.

13. Finally, we run AirCrack to crack the WEP key.

```
aircrack-ng -b 09:AC:90:AB:78 wirelessattack.cap
```

That's it!

How it works...

In this recipe, we used the AirCrack suite to crack the WEP key of a wireless network. AirCrack is one of the most popular programs for cracking WEP. AirCrack works by gathering packets from a wireless connection over WEP and then mathematically analyzing the data to crack the WEP encrypted key. We began the recipe by starting AirCrack and selecting our desired interface. Next, we changed our MAC address which allowed us to change our identity on the network and then searched for available wireless networks to attack using `airodump`. Once we found the network we wanted to attack, we used `aireplay` to associate our machine with the MAC address of the wireless device we were attacking. We concluded by gathering some traffic and then brute-forced the generated CAP file in order to get the wireless password.

Wireless network WPA/WPA2 cracking

WiFi Protected Access, or **WPA** as it's commonly referred to, has been around since 2003 and was created to secure wireless networks and replace the outdated previous standard, WEP encryption. In 2003, WEP was replaced by WPA and later by WPA2. Due to having more secure protocols available, WEP encryption is rarely used.

In this recipe, we will use the AirCrack suite to crack a WPA key. The AirCrack suite (or AirCrack NG as it's commonly referred) is a WEP and WPA key cracking program that captures network packets, analyzes them, and uses this data to crack the WPA key.

Getting ready

In order to perform the tasks of this recipe, experience with the Kali Linux terminal windows is required. A supported wireless card configured for packet injection will also be required. In the case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties.

How to do it...

Let's begin the process of using AirCrack to crack a network session secured by WPA.

1. Open a terminal window and bring up a list of wireless network interfaces.

   airmon-ng

2. Under the interface column, select one of your interfaces. In this case, we will use `wlan0`. If you have a different interface, such as `mon0`, please substitute it at every location where `wlan0` is mentioned.
3. Next, we need to stop the `wlan0` interface and take it down.

   airmon-ng stop wlan0

   ifconfig wlan0 down

4. Next, we need to change the MAC address of our interface. In this case, we will use `00:11:22:33:44:55`.

   macchanger --mac 00:11:22:33:44:55 wlan0

5. Now we need to restart `airmon-ng`.

   airmon-ng start wlan0

6. Next, we will use `airodump` to locate the available wireless networks nearby.

   airodump-ng wlan0

7. A listing of available networks will begin to appear. Once you find the one you want to attack, press *Ctrl* + *C* to stop the search. Highlight the MAC address in the `BSSID` column, right-click, and select copy. Also, make note of the channel that the network is transmitting its signal upon. You will find this information in the `Channel` column. In this case, the channel is `10`.
8. Now we run `airodump` and copy the information for the selected BSSID to a file. We will utilize the following options:
   - `-c` allows us to select our channel. In this case, we use 10.
   - `-w` allows us to select the name of our file. In this case, we have chosen `wirelessattack`.
   - `-bssid` allows us to select our BSSID. In this case, we will paste `09:AC:90:AB:78` from the clipboard.

   airodump-ng -c 10 -w wirelessattack --bssid 09:AC:90:AB:78 wlan0

9. A new terminal window will open displaying the output from the previous command. Leave this window open.
10. Open another terminal window; to attempt to make an association, we will run `aireplay`, which has the following syntax: `aireplay-ng –dauth 1 –a [BSSID] –c [our chosen MAC address] [Interface]`. This process may take a few moments.

    ```
    Aireplay-ng --deauth 1 –a 09:AC:90:AB:78 –c 00:11:22:33:44:55
    wlan0
    ```

11. Finally, we run AirCrack to crack the WPA key. The `–w` option allows us to specify the location of our wordlist. We will use the `.cap` file that we named earlier. In this case, the file's name is `wirelessattack.cap`.

    ```
    Aircrack-ng –w ./wordlist.lst wirelessattack.cap
    ```

    That's it!

How it works...

In this recipe, we used the AirCrack suite to crack the WPA key of a wireless network. AirCrack is one of the most popular programs for cracking WPA. AirCrack works by gathering packets from a wireless connection over WPA and then brute-forcing passwords against the gathered data until a successful handshake is established. We began the recipe by starting AirCrack and selecting our desired interface. Next, we changed our MAC address which allowed us to change our identity on the network and then searched for available wireless networks to attack using `airodump`. Once we found the network we wanted to attack, we used `aireplay` to associate our machine with the MAC address of the wireless device we were attacking. We concluded by gathering some traffic and then brute forced the generated CAP file in order to get the wireless password.

Automating wireless network cracking

In this recipe we will use Gerix to automate a wireless network attack. Gerix is an automated GUI for AirCrack. Gerix comes installed by default on Kali Linux and will speed up your wireless network cracking efforts.

Getting ready

A supported wireless card configured for packet injection will be required to complete this recipe. In the case of a wireless card, packet injection involves sending a packet, or injecting it, onto an already established connection between two parties.

How to do it...

Let's begin the process of performing an automated wireless network crack with Gerix by downloading it.

1. Using `wget`, navigate to the following website to download Gerix.

   ```
   wget https://bitbucket.org/Skin36/gerix-wifi-cracker-pyqt4/
   downloads/gerix-wifi-cracker-master.rar
   ```

2. Once the file has been downloaded, we now need to extract the data from the RAR file.

   ```
   unrar x gerix-wifi-cracker-master.rar
   ```

3. Now, to keep things consistent, let's move the Gerix folder to the `/usr/share` directory with the other penetration testing tools.

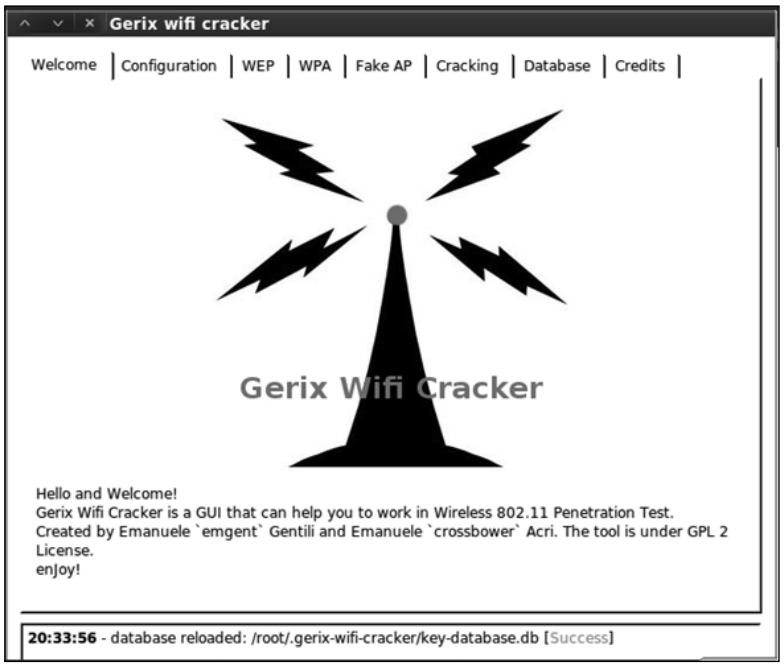
   ```
   mv gerix-wifi-cracker-master /usr/share/gerix-wifi-cracker
   ```

4. Let's navigate to the directory where Gerix is located.

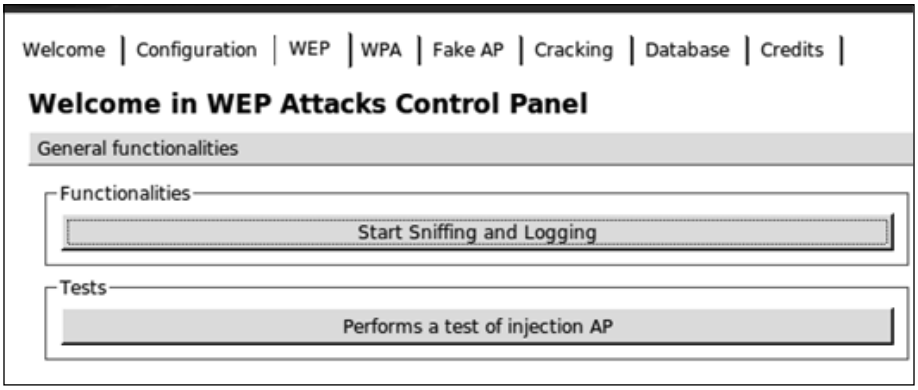
   ```
   cd /usr/share/gerix-wifi-cracker
   ```

5. To begin using Gerix, we issue the following command:

   ```
   python gerix.py
   ```



6. Click on the **Configuration** tab.
7. On the **Configuration** tab, select your wireless interface.
8. Click on the **Enable/Disable Monitor Mode** button.
9. Once Monitor mode has been enabled successfully, under **Select Target Network**, click on the **Rescan Networks** button.
10. The list of targeted networks will begin to fill. Select a wireless network to target. In this case, we select a WEP encrypted network.
11. Click on the **WEP** tab.



12. Under **Functionalities**, click on the **Start Sniffing and Logging** button.
13. Click on the subtab **WEP Attacks (No Client)**.
14. Click on the **Start false access point authentication on victim** button.
15. Click on the **Start the ChopChop attack** button.
16. In the terminal window that opens, answer **Y** to the **Use this packet** question.
17. Once completed, copy the `.cap` file generated.
18. Click on the **Create the ARP packet to be injected on the victim access point** button.
19. Click on the **Inject the created packet on victim access point** button.
20. In the terminal window that opens, answer **Y** to the **Use this packet** question.
21. Once you have gathered approximately 20,000 packets, click on the **Cracking** tab.
22. Click on the **Aircrack-ng – Decrypt WEP Password** button.

That's it!

How it works...

In this recipe, we used Gerix to automate a crack on a wireless network in order to obtain the WEP key. We began the recipe by launching Gerix and enabling the monitoring mode interface. Next, we selected our victim from a list of attack targets provided by Gerix. After we started sniffing the network traffic, we then used Chop Chop to generate the CAP file. We concluded the recipe by gathering 20,000 packets and brute-forced the CAP file with AirCrack.

With Gerix, we were able to automate the steps to crack a WEP key without having to manually type commands in a terminal window. This is an excellent way to quickly and efficiently break into a WEP secured network.

Accessing clients using a fake AP

In this recipe, we will use Gerix to create and set up a fake **access point** (**AP**). Setting up a fake access point gives us the ability to gather information on each of the computers that access it. People in this day and age will often sacrifice security for convenience. Connecting to an open wireless access point to send a quick e-mail or to quickly log into a social network is rather convenient. Gerix is an automated GUI for AirCrack.

Getting ready

A supported wireless card configured for packet injection will be required to complete this recipe. In the case of a wireless card, packet injection involves sending a packet, or injecting it onto an already established connection between two parties.

How to do it...

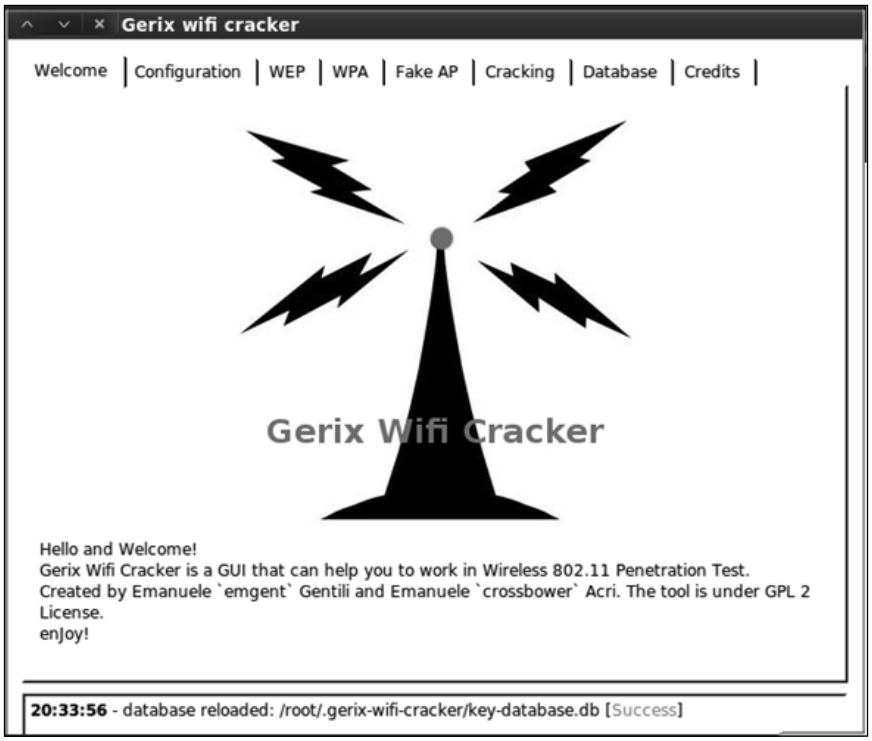
Let's begin the process of creating a fake AP with Gerix.

1. Let's navigate to the directory where Gerix is located:

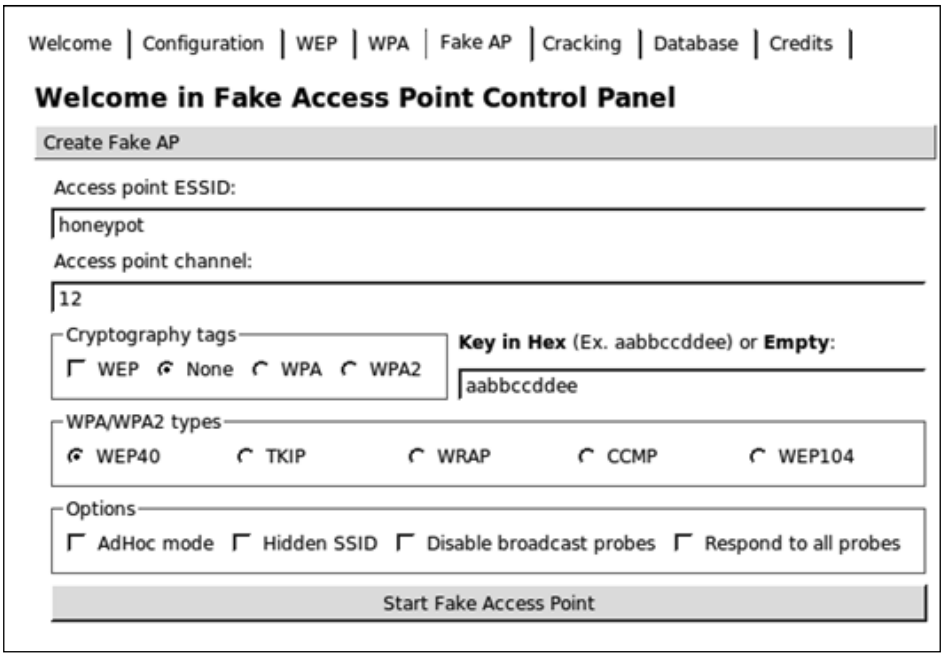
```
cd /usr/share/gerix-wifi-cracker
```

2. To begin using Gerix, we issue the following command:

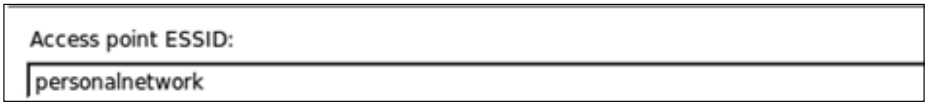
```
python gerix.py
```



3. Click on the **Configuration** tab.
4. On the **Configuration** tab, select your wireless interface.
5. Click on the **Enable/Disable Monitor Mode** button.
6. Once Monitor mode has been enabled successfully, under **Select Target Network**, press the **Rescan Networks** button.
7. The list of targeted networks will begin to fill. Select a wireless network to target. In this case, we select a WEP encrypted network.
8. Click on the **Fake AP** tab.



9. Change the **Access Point ESSID** from `honeypot` to something less suspicious. In this case, we are going to use `personalnetwork`.



10. We will use the defaults on each of the other options. To start the fake access point, click on the **Start Face Access Point** button.

That's it!

How it works...

In this recipe, we used Gerix to create a fake AP. Creating a fake AP is an excellent way of collecting information from unsuspecting users. The reason fake access points are a great tool to use is that to your victim, they appear to be a legitimate access point, thus making it trusted by the user. Using Gerix, we were able to automate the creation of setting up a fake access point in a few short clicks.

URL traffic manipulation

In this recipe, we will perform a URL traffic manipulation attack. URL traffic manipulation is very similar to a Man In The Middle attack, in that we will route traffic destined for the Internet to pass through our machine first. We will perform this attack through ARP poisoning. ARP poisoning is a technique that allows you to send spoofed ARP messages to a victim on the local network. We will execute this recipe using arpspoof.

How to do it...

Let's begin the process of URL traffic manipulation.

1. Open a terminal window and execute the following command to configure IP tables that will allow our machine to route traffic:

   ```
   sudo echo 1 >> /proc/sys/net/ipv4/ip_forward
   ```

2. Next, we launch arpspoof to poison traffic going from our victim's machine to the default gateway. In this example, we will use a Windows 7 machine on my local network with an address of `192.168.10.115`. Arpspoof has a couple of options that we will select and they include:

   - `-i` allows us to select our target interface. In this case, we will select `wlan0`.
   - `-t` allows us to specify our target.

   The syntax for completing this command is `arpspoof -i [interface] -t [target IP address] [destination IP address]`.

   ```
   sudo arpspoof -i wlan0 -t 192.168.10.115 192.168.10.1
   ```

3. Next, we will execute another arpspoof command that will take traffic from the destination in the previous command (which was the default gateway) and route that traffic back to our Kali machine. In this example our IP address is `192.168.10.110`.

   ```
   sudo arpspoof -i wlan0     -t 192.168.10.1 192.168.10.110
   ```

   That's it!

How it works...

In this recipe, we used ARP poisoning with arpspoof to manipulate traffic on our victim's machine to ultimately route back through our Kali Linux machine. Once traffic has been rerouted, there are other attacks that you can run against the victim, including recording their keystrokes, following websites they have visited, and much more!

Port redirection

In this recipe, we will use Kali to perform port redirection, also known as port forwarding or port mapping. Port redirection involves the process of accepting a packet destined for one port, say port 80, and redirecting its traffic to a different port, such as 8080. The benefits of being able to perform this type of attack are endless because with it you can redirect secure ports to unsecure ports, redirect traffic to a specific port on a specific device, and so on.

How to do it...

Let's begin the process of port redirection/forwarding.

1. Open a terminal window and execute the following command to configure IP tables that will allow our machine to route traffic:

   ```
   Sudo echo 1 >> /proc/sys/net/ipv4/ip_forward
   ```

2. Next, we launch arpspoof to poison traffic going to our default gateway. In this example, the IP address of our default gateway is `192.168.10.1`. Arpspoof has a couple of options that we will select and they include:

   - `-i` allows us to select our target interface. In this case, we will select `wlan0`.

   The syntax for completing this command is `arpspoof -i [interface] [destination IP address]`.

   ```
   sudo arpspoof -i wlan0 192.168.10.1
   ```

3. Next, we will execute another arpspoof command that will take traffic from our destination in the previous command (which was the default gateway) and route that traffic back to our Kali Linux machine. In this example our IP address is `192.168.10.110`.

   ```
   iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j
   REDIRECT --to-port 8080
   ```

   That's it!

How it works...

In this recipe, we used ARP poisoning with arpspoof and IPTables routing to manipulate traffic on our network destined for port 80 to be redirected to port 8080. The benefits of being able to perform this type of attack are endless because with it you can redirect secure ports to unsecure ports, redirect traffic to a specific port on a specific device, and so on.

Sniffing network traffic

In this recipe, we will examine the process of sniffing network traffic. Sniffing network traffic involves the process of intercepting network packets, analyzing it, and then decoding the traffic (if necessary) displaying the information contained within the packet. Sniffing traffic is particularly useful in gathering information from a target, because depending on the websites visited, you will be able to see the URLs visited, usernames, passwords, and other details that you can use against them.

We will use Ettercap for this recipe, but you could also use Wireshark. For demonstration purposes, Ettercap is a lot easier to understand and apply sniffing principles. Once an understanding of the sniffing process is established, Wireshark can be utilized to provide more detailed analysis.

Getting ready

A wireless card configured for packet injection is required to complete this recipe although you can perform the same steps over a wired network. In case of a wireless card, packet injection involves sending a packet, or injecting it, onto an already established connection between two parties.

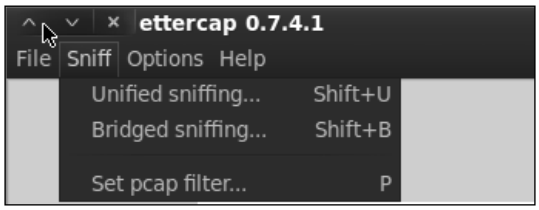
How to do it...

Let's begin the process of sniffing network traffic by launching Ettercap.

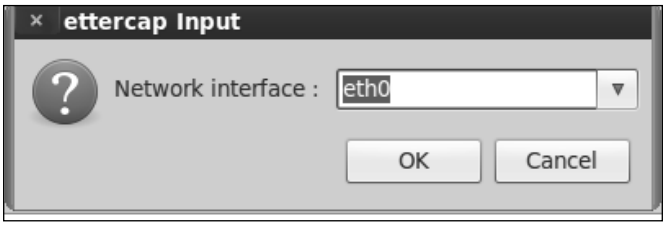
1. Open a terminal window and start Ettercap. Using the `-G` option, launch the GUI:

```
ettercap -G
```

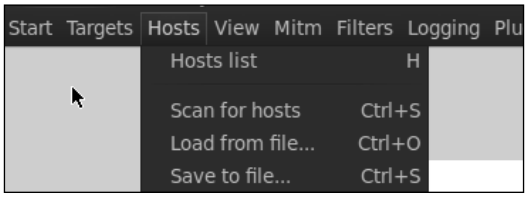
2. We begin the process by turning on **Unified sniffing**. You can press *Shift + U* or use the menu and navigate to **Sniff** | **Unified sniffing**.



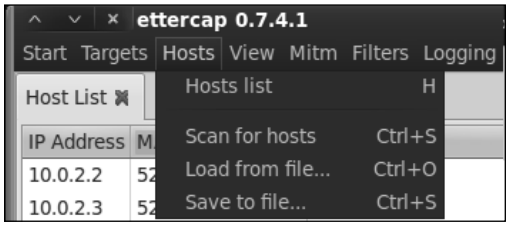
3. Select the network interface. In case of using a MITM attack, we should select our wireless interface.



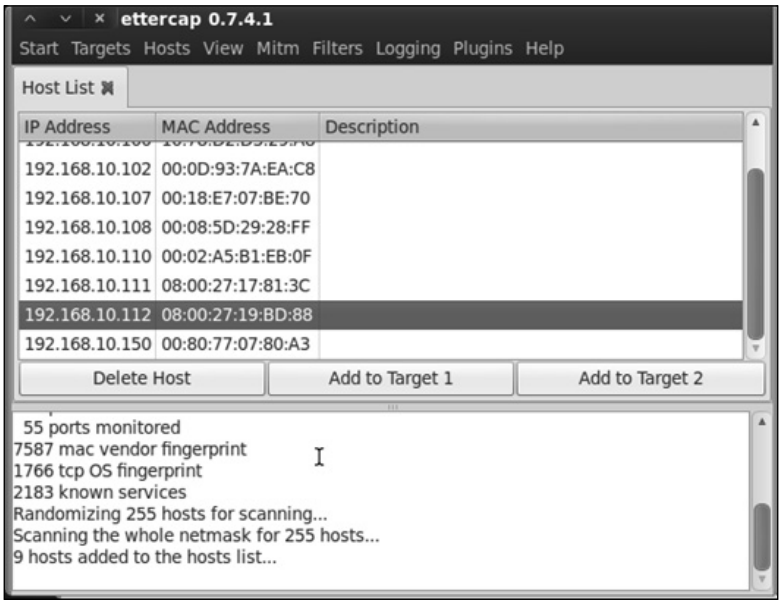
4. Next, we turn on **Scan for hosts**. This can be accomplished by pressing *Ctrl* + S or use the menu and navigate to **Hosts** | **Scan for hosts**.



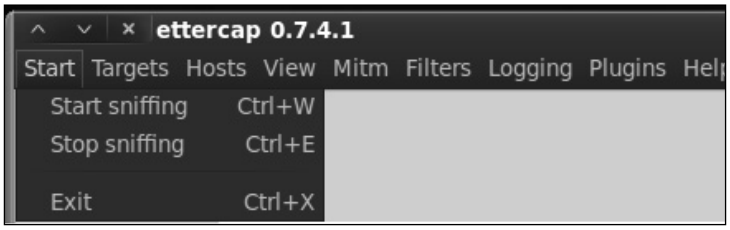
5. Next, we bring up the **Host List**. You can either press *H* or use the menu and navigate to **Hosts** | **Host List**.



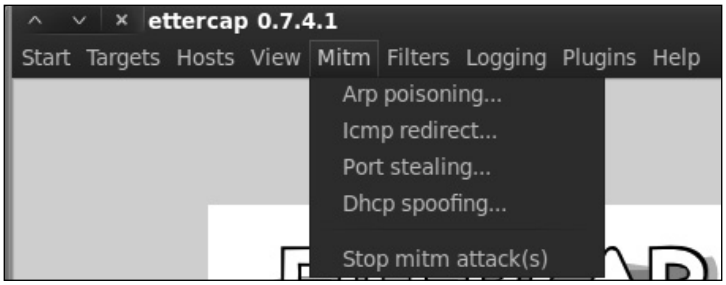
6. We next need to select and set our targets. In our case, we will select `192.168.10.111` as our Target 1 by highlighting its IP address and pressing the **Add To Target 1** button.



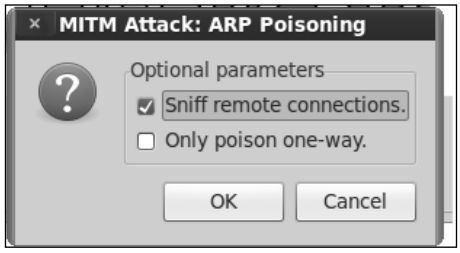
7. Now we are able to allow Ettercap to begin sniffing. You can either press *Ctrl + W* or use the menu and navigate to **Start | Start sniffing**.



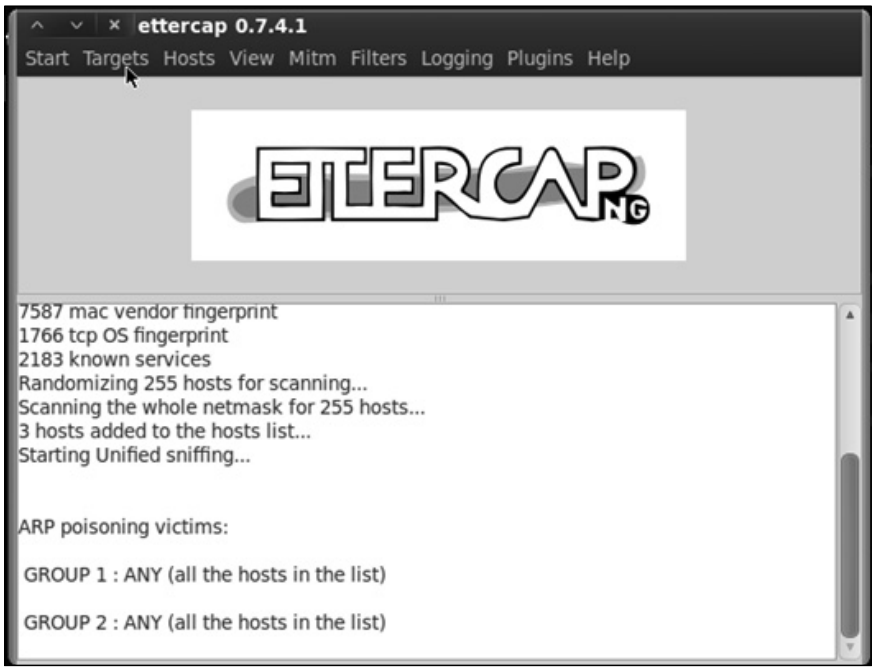
8. Finally, we begin the ARP poisoning process. From the menu, navigate to **Mitm | Arp poisoning...**.



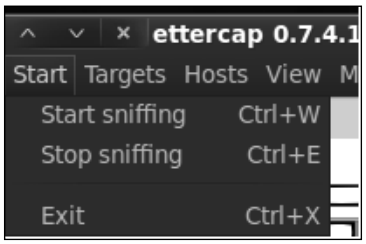
9. In the window that appears, check the optional parameter for **Sniff remote connections**.



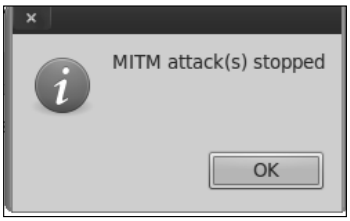
10. Depending on the network traffic, we will begin to see information.



11. Once we have found what we are looking for (usernames and passwords). We will turn off Ettercap. You can do this by either pressing *Ctrl + E* or by using the menu and navigating to **Start** | **Stop sniffing**.



12. Now we need to turn off ARP poisoning and return the network to normal.



How it works...

This recipe included an MITM attack that works by using ARP packet poisoning to eavesdrop on wireless communications transmitted by a user. We began the recipe by launching Ettercap and scanning for our hosts. We then began the process of ARP poisoning the network. ARP poisoning is a technique that allows you to send spoofed ARP messages to a victim on the local network.

We concluded the recipe by starting the packet sniffer and demonstrated a way to stop ARP poisoning and return the network back to normal. This step is key in the detection process as it allows you to not leave the network down once you have stopped poisoning the network.

This process is useful for gathering information as it's being transmitted across the wireless network. Depending on the traffic, you will be able to gather usernames, passwords, bank account details, and other information your targets send across the network. This information can also be used as a springboard for larger attacks.

Where to buy this book

You can buy Kali Linux Cookbook from the Packt Publishing website:
`http://www.packtpub.com/kali-linux-cookbook/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/kali-linux-cookbook/book

Linux E-mail Second Edition

Set up, maintain, and secure a small office e-mail server

Ian Haycox

Ralf Hildebrandt

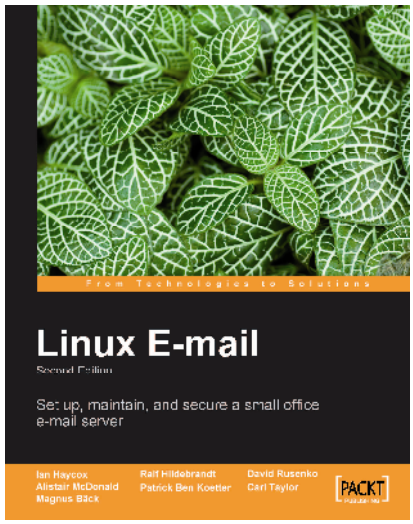
David Rusenko

Alistair McDonald

Patrick Ben Koetter

Carl Taylor

Magnus Bäck



Chapter No. 4

"Providing Webmail Access"

In this package, you will find:

A Biography of the authors of the book

A preview chapter from the book, Chapter NO.4 "Providing Webmail Access"

A synopsis of the book's content

Information on where to buy this book

About the Authors

Ian Haycox is a freelance IT consultant based in France and actively contributes to open source projects. He has twenty-five years of software development experience in the enterprise integration, telecommunications, banking, and television sectors.

Ian has a degree in Computer Science from the University of Hertfordshire, UK, and now runs his own web design company (`http://www.ianhaycox.com/`) and Linux programming consultancy.

My thanks to Debbie for supplying me with copious amount of coffee and cheese sandwiches.

Alistair McDonald is a software developer and IT consultant. He has worked as a freelancer in the UK for 15 years, developing cross-platform software systems in C, C++, Perl, Java, and SQL. He has been using open source software for over 20 years and implementing systems using it for the past 10 years.

Last year, he gave up his freelance career and joined JDA Software, working in a technical role in their Service Industries division.

Alistair is also the author of the book *SpamAssassin: A practical guide to integration and configuration*, published by Packt .

I would like to thank my wife Louise for the support she has given me throughout the writing of all my books.

Magnus Bäck has been playing and working with computers since his childhood days. He is interested in everything in the computer field, from digital typography and compilers, to relational databases and UNIX. His interests also include e-mail services, and he is an active contributor to the Postfix mailing list. Besides computers, he enjoys photography, cars, and bicycling.

Magnus holds a Master's degree in Computer Science and Engineering from Lund Institute of Technology, Sweden, and currently works with software configuration management for mobile phone software at Sony Ericsson Mobile Communications.

Ralf Hildebrandt is an active and well-known figure in the Postfix community, working as a Systems Engineer for T-Systems, a German telecommunications company.

He speaks about Postfix at industry conferences and hacker conventions, and contributes regularly to a number of open source mailing lists. Ralf Hildebrandt is the co-author of *The Book of Postfix*.

Patrick Ben Koetter is an active and well-known figure in the Postfix community, working as an Information Architect. Patrick Koetter runs his own company, consulting and developing corporate communication for customers in Europe and Africa.

He speaks about Postfix at industry conferences and hacker conventions, and contributes regularly to a number of open source mailing lists. Patrick Koetter is the co-author of *The Book of Postfix*.

David Rusenko was born in Paris, France, and spent most of his childhood overseas. He began working as a freelance Web Designer in 1996 and had his first experience with open source, a box copy of Red Hat 5.2, shortly after in 1999. After six years and as many versions of Red Hat, he now creates appealing web pages and devises solutions implementing high availability through clustering and alternate security models.

He founded Aderes (`http://www.aderes.net`) in 2001, a company that provides e-mail and web-based security solutions. His search for an appropriate Webmail Platform for the company led him to SquirrelMail. Initially managing all aspects of the business—from the technical concerns to customer support—gave him the experience that he now contributes to the Webmail chapter of this book.

David has studied both, Information Sciences and Technology (IST) and Management Information Systems (MIS) at the Pennsylvania State University. He speaks English and French fluently, and is conversational in Arabic. During his free time and vacations, he enjoys scuba diving, backpacking, playing racquetball, and playing electronic music records.

Carl Taylor has worked over 20 years in the IT industry and has spent the majority of that time working on UNIX type systems, mainly communications or office automation projects. He was an early user of the UseNet network and taught himself to program in C through working on a variety of open source software. His experience covers roles including pre and post sales support, product development, end user training and management.

Carl now runs his own web solutions development company "Adepteo", where they specialize in intranet and workflow products building on the best open source applications available. Whilst not working or looking after his children, Carl is something of a dance addict and is currently learning Latin Ballroom and Salsa.

Linux E-mail Second Edition

Set up, maintain, and secure a small office e-mail server

Many businesses want to run their e-mail servers on Linux for greater control and flexibility of corporate communications, but getting started can be complicated. The attractiveness of a free-to-use and robust e-mail service running on Linux can be undermined by the apparent technical challenges involved. Some of the complexity arises from the fact that an e-mail server consists of several components that must be installed and configured separately, then integrated together.

This book gives you just what you need to know to set up and maintain an e-mail server. Unlike other approaches that deal with one component at a time, this book delivers a step-by-step approach across all the server components, leaving you with a complete working e-mail server for your small business network.

What This Book Covers

Chapter 1: Linux and E-mail Basics takes you through the essential elements of a Linux e-mail server and the network and mail protocols that make e-mail possible. Like it or not, running a Linux e-mail server does require some understanding of the underlying networking, and this chapter is where you will start to get that understanding. This chapter explains the benefits and disadvantages of running your own e-mail server and provides some guidance on hardware sizing for a typical organization.

Chapter 2: Setting Up Postfix speaks about basic Postfix setup. Postfix is our chosen Mail Transfer Agent (MTA), which forms the heart of any e-mail server. The MTA is responsible, among other things, for moving messages between the various mail servers on the Internet.

Chapter 3: Incoming mail with POP and IMAP covers what to do with incoming e-mails. It will show you how to set up IMAP and POP access to mailboxes. This means users will be able to send and receive messages using their familiar e-mail clients.

Chapter 4: Providing Webmail Access shows how to set up webmail access using SquirrelMail. This will give users an easy, out-of-office access to their e-mail.

Chapter 5: Securing Your Installation looks at how your installation can be secured to prevent misuse of your users' data and the e-mail facility itself.

Chapter 6: Getting Started with Procmail discusses the basics of Procmail and gets you familiar with the various fi les that Procmail uses to load recipes, the core principles of filtering, and the options available.

Chapter 7: Advanced Procmail explores Procmail and explains a large number of services and a large amount of functionality that it can provide in getting mail under control. It also discusses the advanced features of Procmail and their benefits.

Chapter 8: Busting Spam with SpamAssassin shows the use of SpamAssassin in conjunction with Procmail to filter out the wide range of spam that afflicts the modern e-mail user.

Chapter 9: Antivirus Protection shows another way to protect users from rogue e-mail—this time the spread of e-mail viruses. Using ClamAV you can scan mail for viruses and schedule tasks to maintain an up-to-date antivirus database.

Chapter 10: Backing up your System will show you how to protect all your hardwork by backing up not only the e-mail itself, but also all of the configuration options that make up your e-mail server. Examples are provided to create an automated backup schedule to minimize data loss. Of course, you'll also learn how to restore data from these backups.

4
Providing Webmail Access

You learned how to set up and configure an e-mail server in the previous chapters. Now that your e-mail server is ready to serve, how will your users access it? In this chapter, you will learn about the following:

- The benefits and disadvantages of a webmail access solution
- The SquirrelMail webmail package
- Setting up and configuring SquirrelMail
- What SquirrelMail plugins are and what they can do
- How to make SquirrelMail more secure

In the next section, we will introduce the SquirrelMail software package and examine the pros and cons of this and other webmail access solutions. After that, we will follow the installation and configuration of SquirrelMail step by step. Next, we will examine the installation of plugins and include a reference of useful plugins. Finally, we'll include some tips on how to secure SquirrelMail.

The webmail solution

A **webmail solution** is a program or a series of scripts that is run on a server, is accessible over the web, and provides access to e-mail functions similar to a conventional mail client. It is used by Yahoo! Mail, Microsoft Hotmail, Microsoft Outlook Web Access, and Gmail as the primary interface to their e-mail solutions. You may already be familiar with various forms of webmail.

Though we will be examining the SquirrelMail webmail solution specifically, the benefits and drawbacks of SquirrelMail apply to most webmail systems in the market. From this point of view, we will approach the issue from a general perspective, and then in detail for the SquirrelMail package.

The benefits

This section will focus on the advantages offered by installing and maintaining a webmail solution. As with any list, it is not entirely comprehensive. Many benefits will be specific to a particular case; it is important to carefully examine and consider how the following qualities impact your individual situation.

The main benefits we will explore in this section are as follows:

- Easy and quick access with little or no setup
- Easy remote access
- No need to maintain client software or configuration
- Provision of a user interface to configure mail server options
- Possible security benefits

Easy and quick access

Although well suited to certain situations, traditional mail access solutions can often be difficult to set up and maintain. Generally, this involves installing software on a client's local computer and configuring it. This can be difficult, especially in cases where users need to set up the software themselves. Configuration can often be even more problematic as some users may not be competent enough to follow even a very detailed set of instructions. These instructions also need to be provided and maintained for many different mail clients on many different platforms.

However, a webmail solution does not have most of these problems. All of the user's settings can be configured on the server as the application itself resides on the server. This translates to almost zero set up time for the user. Once they have received their login credentials, they can visit the webmail site and instantly have access to all of their mail. The user is able to access the site instantly to send and receive e-mail.

As the Internet is so common now, many users will be familiar with webmail sites such as Google Mail and Windows Live Hotmail, which offer free e-mail services. However, the user interface provided by an open source package may be more primitive and lack some visual features. Squirrelmail provides access to e-mail, including the ability to send and receive attachments, and offers a good user interface.

It is also worth mentioning that a webmail solution can offer what certain traditional mail clients call **groupware** features. These features let groups communicate and coordinate in ways that complement e-mail communication. Examples of groupware components are private calendars, shared calendars, meeting scheduling, To-do lists, and other similar tools.

These applications can be preconfigured so that a user can instantly begin using them without having to configure them on their own. Several SquirrelMail plugins which implement these features are available from the SquirrelMail website.

Easy remote access

Another problem with traditional mail access software is that it is not portable, as an e-mail client needs to be installed and configured on a computer. Once it has been downloaded, installed, and configured on a particular computer, it is accessible only on that computer. Without webmail, users on the road will not be able to access e-mail from friends' computers, mobile devices, or Internet booths at airports.

However, in a webmail solution, e-mail can be accessed from any location with an Internet connection. Employees can access their work e-mail from any computer with an Internet connection and a suitable browser.

As the administrator, you can choose to permit or deny users from accessing e-mail in insecure situations. By requiring the connection to be encrypted, you can ensure that when a user is in a remote location, their communication with the server is secure.

No need to maintain clients

Even if software mail clients have been installed and properly configured, they must be maintained. When a new version is released, all clients must be updated. This is not necessarily an easy task. Software that does not work as expected can result in a large number of support-desk calls.

Updating the software on each client can be a very large administrative burden. In fact, many expensive software packages are designed for the specific purpose of updating software on individual machines automatically. Despite this, problems specific to each local machine often arise and must be solved individually. It may also be difficult to convey instructions or notifications to remote branch locations or remote workers. With a webmail solution, this is not necessary.

In contrast to this, a webmail solution is centrally maintained and administered. The webmail application resides on the server. With webmail, only the web server and the webmail package need to be upgraded. Any exceptions or problems that arise can be dealt with before or during the upgrade. The software upgrade itself can be run through on a test system before it is deployed on a live system. Although changes in settings are rare with SquirrelMail, it is possible to update a user's settings to make them compatible with the changes introduced in an updated version.

Additionally, while upgrading or changing a mail server platform, testing effort can be greatly reduced as only supported browser versions need to be tested. It is advisable to mandate particular browser versions for corporate computers. In contrast with e-mail clients, there is no need to test on all of the possible clients and software platforms.

Configuring mail server interface via the user interface

Many traditional desktop e-mail clients provide only e-mail functionality and nothing more. Often there is no support for other essential tasks (such as changing the access password) that are performed on behalf of a mail user. Certain configuration options that reside on the server may require additional software applications or external solutions to provide for these needs. Examples of mail server options that may need to be configured include each user's password and junk mail filtering settings.

In the case of the SquirrelMail webmail application, many plugins have been developed that provide these features. For example, a user is able to change his/her password directly from the webmail interface. Also, there are plugins and systems that allow users to easily sign up without any direct human intervention. This may be useful if you are interested in providing a service where users can sign up without needing an administrative overhead.

Possible security benefits

This issue can be seen in two different ways—it is for this reason that the title is listed as "*Possible*" *security benefits*. Nonetheless, this is still an interesting point to examine.

In the software client access model, e-mail is traditionally downloaded onto the local user's computer, being stored in one or more personal folders. From a security perspective, this may be a bad thing. Users of the system may not be as conscientious or knowledgeable about computer security as a trained computer administrator might be. It is often much easier to gain unauthorized access to an end user's computer than a properly configured and secured server. The implication is that someone who stole a company laptop might be able to access all the e-mail stored on that computer.

There is one more disadvantage associated with the client access model. Even if an employee is terminated, he/she may still have access to all of the e-mail that resides on his/her local office computer. It may take a certain amount of time before important information may be secured. A disgruntled worker might easily connect an external storage source to their local office computer and download any data they desire.

It is also worth noting that in a webmail model, all e-mail is centrally stored. If an attacker were to gain access to the central e-mail server, he/she might access all the e-mail stored on that server. However, it is possible that an attacker will gain access to all the e-mail if the central mail server is compromised even if a webmail system is not used.

The disadvantages

This section focuses on the disadvantages resulting from providing and supporting a webmail solution. The warning given in the previous section applies: This list is not entirely comprehensive. Each situation is unique, and may bring its unique disadvantages.

We will go over the following disadvantages of a webmail solution:

- Performance issues
- Compatibility with large e-mail volumes
- Compatibility with e-mail attachments
- Security issues

Performance

The traditional e-mail client is designed in the client-server model. One mail server accepts and delivers e-mail to and from other mail servers. However, a desktop mail client can offer many additional productivity-enhancing features such as message sorting, searching, contact list management, attachment handling, along with more recent ones such as spam filtering and message encryption.

Each of these features may require a certain amount of processing power. The required level of processing power may be negligible when it comes to storing one user's e-mail on a desktop computer, but providing these features may be problematic when applied on a larger scale to a single server.

When examining the performance issue, it is important to consider the number of potential users that will access the webmail application and size a server accordingly. A single server may be able to easily handle something like 300 users, but if the number of users increases significantly, server load may become an issue.

For example, searching through several years' archived mail may take a few seconds on a client's computer. When one user performs this task using webmail, the load will be similar. However, if many clients request this operation at short intervals or concurrently, it may be difficult for the server to process all the requests in a timely manner. This may result in pages being served at a slower rate or, in extreme circumstances, the server failing to respond.

Optimally, load testing in the appropriate conditions should be performed if there is any concern that a server may not be able to handle a particular load of users.

Compatibility with large e-mail volumes

The webmail solution is not well suited to large mail volumes. This disadvantage is related to the previous one, but is more related to the amount of data sent. Even with a relatively low number of users, a large volume of e-mails may be difficult to manage in a webmail application. There are mainly the following two reasons for this:

- Firstly, every e-mail viewed and every folder listed must be sent from the server each time. With a traditional e-mail client, the client software can manage e-mail messages, creating lists and views to suit the user. However, with a webmail solution, this is performed on the server. So, if there are many users, this overhead may use a significant proportion of the server's resources.
- Secondly, each interaction with the webmail application requires a **Hypertext Transfer Protocol** (**HTTP**) request and response. These messages will typically be larger than those between an e-mail server and a desktop e-mail client. There may also be less parallelism when using a webmail client, in other words, fewer things going on at the same time. A desktop e-mail client may be able to check for new e-mails in several folders at the same time, but a webmail client will typically perform these tasks one after the other, if they occur automatically at all.

Compatibility with e-mail attachments

The webmail solution is not well suited to e-mail attachments. By virtue of the fact that a webmail application resides on a remote server, any and all e-mail attachments must first be uploaded onto that server. For a couple of reasons, it may be difficult or impossible to accomplish this operation with too many attachments or with attachments that are large in size.

Difficulties uploading large attachments may arise due to limited storage space on the webmail server. Large attachments may take a long time to upload over the HTTP protocol and even longer over HTTPS. Additionally, many file size limits may be imposed on uploaded files. PHP, the programming language used with SquirrelMail, imposes a 2MB limit on uploaded files in its default configuration.

The solution to the above problem may lie in the nature of the webmail access solution—e-mail and the mail access software reside on the server. In a traditional mail client, e-mail is often downloaded before the user is aware of the contents or size of the particular e-mail message. As opposed to this, in the case of webmail, the user is able to view e-mail with large attachments without downloading the attachments—a particular benefit to those without high-speed internet connections.

Finally, downloading and uploading large e-mail attachments from the server may cause a performance issue with the user interface. Many users are frustrated by an attachment's upload time in the webmail application, especially as the message cannot be sent until the attachment is uploaded. In a traditional mail client, the attachment is attached instantly, while the message takes time to send.

Security issues

The last issue we will examine is the potential for security shortcomings. One important feature of a webmail access solution also creates a potential problem. The benefit of remote access gives way to the potential insecurity of the local machines upon which the user accesses his/her mail.

A computer that is not directly under your control may be controlled by a third-party intent on accessing your information. Normally, a computer does not record a user's individual keystrokes. Internet cafes and kiosks, and even the home computers of employee's could be running malicious software. This malicious software may monitor keystrokes and websites visited. A user must type in his/her password or login credentials to gain access to the system. When these credentials are captured and stored on the computer with malicious software, they can be intercepted and used by third parties for unauthorized access.

Even if we take malicious intent out of the picture, there are still certain situations that may prove to pose security risks. For example, many modern web browsers offer the option of saving a password whenever it is entered. This password is stored on the local computer where the website is visited. If a user logs in to the webmail application and accidentally saves their password on the local computer, this password may be accessible to any user with access to that local computer.

Finally, users may inadvertently leave themselves logged in to the webmail application. Without logging out, any user with access to that specific computer might be able to gain access to the user's mail account.

The SquirrelMail webmail package

The following screenshot shows the SquirrelMail login screen:

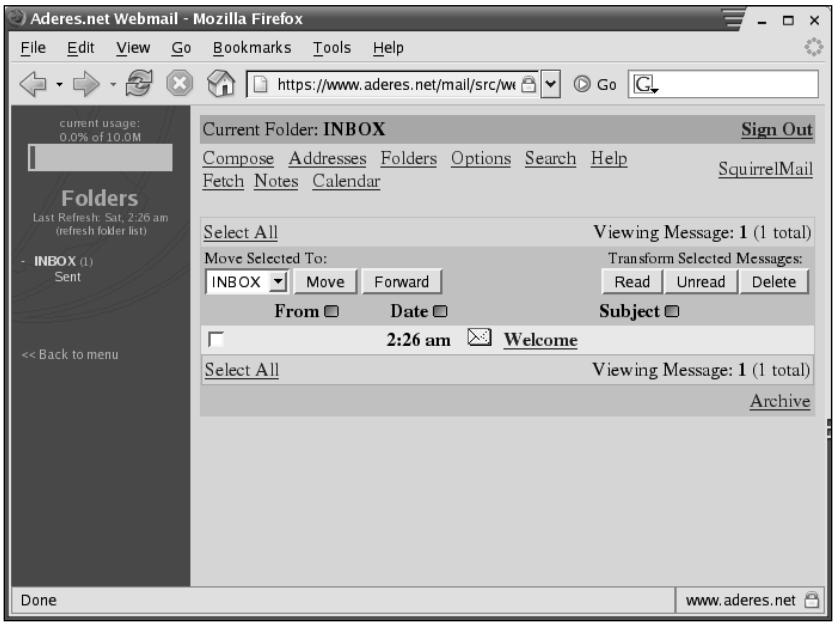
SquirrelMail was chosen based on the combination of the following features it provides:

- It is a proven, stable, and mature webmail platform.
- It has been downloaded over two million times.
- It is standards-based and renders pages in pure HTML 4.0 without requiring the use of JavaScript.

SquirrelMail also includes the following features (and many more, via the flexible plugin system):

- Strong MIME support
- Address book functionality
- A spell checker
- Support for sending and receiving HTML e-mail
- Template and theme support
- Virtual host support

The following screenshot shows an inbox where you can see some of these features:



SquirrelMail installation and configuration

SquirrelMail installation and configuration may seem daunting if you are not familiar with installing web applications. But by following the instructions to be discussed next, SquirrelMail can be installed without difficulty.

Prerequisites to installation

SquirrelMail requires both PHP and a web server that supports PHP scripts to be installed before proceeding. In our case, we will be using the Apache2 web server, although others will work as well.

First, we will go over the basic requirements, and what to do if you do not meet them. Then, we will go over some more advanced requirements that may impact on certain features within SquirrelMail.

Basic requirements

At the time of writing, the most current stable version of SquirrelMail available is 1.4.19. The following instructions apply to this version. There are two basic requirements for a SquirrelMail installation.

Installing Apache2

Any modern version of Apache that supports PHP, either the 1.x or 2.x series, will do the trick. Here we provide instructions for using Apache2. To query for an Apache installation on an RPM package management-based system, issue the following command at the prompt:

```
$ rpm -q apache
    apache-1.3.20-16
```

If, as in the example just seen, a version of Apache is returned, then the Apache web server is installed on your system.

To query for an Apache installation on a Debian package management-based system, issue the following command at the prompt:

```
$ apt-cache search --installed apache2 | grep HTTP
    libapache2-mod-evasive - evasive module to minimize HTTP DoS or brute
    force attacks
    libpoe-component-server-http-perl - foundation of a POE HTTP Daemon
    libserf-0-0 - high-performance asynchronous HTTP client library
    libserf-0-0-dbg - high-performance asynchronous HTTP client library
    debugging symbols
    libserf-0-0-dev - high-performance asynchronous HTTP client library
    headers
    nanoweb - HTTP server written in PHP
    php-auth-http - HTTP authentication
    apache2 - Apache HTTP Server metapackage
    apache2-doc - Apache HTTP Server documentation
    apache2-mpm-event - Apache HTTP Server - event driven model
    apache2-mpm-prefork - Apache HTTP Server - traditional non-threaded
    model
    apache2-mpm-worker - Apache HTTP Server - high speed threaded model
    apache2.2-common - Apache HTTP Server common files
```

Similar commands are available for other distributions using other package management systems.

If you do not have an Apache installation present, it is best to first look into your distribution for a copy of Apache—such as on your operating system installation CDs or using an online package repository. Alternatively, you may visit the home page for the Apache foundation at `http://www.apache.org`.

PHP

The PHP programming language (version 4.1.0 or greater, including all PHP 5 versions) is required in order to install SquirrelMail. To check if your system has PHP installed, simply attempt to run it with the following command:

```
$ php -v
```

If the command succeeds, you will see a message describing the version of PHP that is installed. If PHP version 4.1.0 or higher is present, then your system has the required software. Otherwise, you will need to install or upgrade your current installation. As with Apache, it is best to look to your distribution for a copy to install. Alternatively, you may also visit `http://www.php.net`.

Perl

The Perl programming environment is not required for SquirrelMail, but having it available makes configuration of SquirrelMail much simpler. In this chapter, we assume that you will have Perl accessible to enable easy configuration of SquirrelMail.

To query for a Perl installation on an RPM-based system, simply attempt to run it with the following command:

```
$ perl -v
```

If the command succeeds, you will see a message describing the version of Perl that is installed.

If any version of Perl is present, your system has the required software. Otherwise, you will need to install or upgrade your current installation. As with Apache, it is best to look into your distribution for a copy to install. Alternatively, you may also visit `http://www.perl.com/get.html`.

Review configuration

You will need to review the PHP configuration file `php.ini` to ensure that settings are correct. On most Linux systems, this file may be found at `/etc/php.ini`.

`php.ini` is a text file and can be edited with a text editor such as Emacs or vi. Firstly, if you want users to be able to upload attachments, make sure that the option `file_uploads` is set to `On`:

```
; Whether to allow HTTP file uploads.
file_uploads = On
```

The next option within the `php.ini` file you may want to change is `upload_max_filesize`. This setting applies to uploaded attachments and determines the maximum file size of an uploaded file. It may be helpful to change this to something reasonable, such as `10M`.

```
; Maximum allowed size for uploaded files.
upload_max_filesize = 10M
```

Installing SquirrelMail

SquirrelMail may be installed either though a package or directly from source. While no source code compilation takes place in either method, upgrades are made easier using the packages.

Many of the various Linux and Unix distributions include the SquirrelMail package. Install the appropriate package from your distribution to use the binary method. On many Linux distributions, this may be an RPM file that begins with `squirrelmail`....

However, an updated version of SquirrelMail may not be included or available for your specific distribution.

The following are the advantages of using the version of SquirrelMail provided with a Linux distribution:

- It will be very simple to install SquirrelMail.
- It will require much less configuration as it will be configured to use the standard locations chosen by your Linux distributer.
- Updates will be very easy to apply, and migration issues may be dealt with by the package management system.

The following are the disadvantages of using the version of SquirrelMail provided with a Linux distribution:

- It may not be the latest version. For example, a more recent version that may fix a security vulnerability may have been released, but Linux distributors may not have created a new package yet.

- Sometimes Linux distributions alter packages by applying patches. These patches may affect the operation of the package, and may make getting support or help more difficult.

Source installation

If you do not install SquirrelMail through your distribution, you will need to obtain the appropriate tarball. To do so, visit the SquirrelMail website at `http://www.squirrelmail.org`, and click **download it here**. At the time of writing, this link is `http://www.squirrelmail.org/download.php`.

There are two versions available for download, a **stable version** and a **development version**. Unless you have specific reasons for choosing otherwise, it is generally best to choose the stable version. Download and save this file to an intermediate location.

```
$ cd /tmp
$ wget http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fprdownloads.sourceforge.net%2Fsquirrelmail%2Fsquirrelmail-1.4.19.tar.gz
```

Next, unpack the tarball (`.tar.gz`) file. You may use the following command:

```
$ tar xfz squirrelmail-1.4.19.tar.gz
```

Move the folder just created to your web root folder. This is the directory from which Apache serves pages. In this case, we will assume that `/var/www/html` is your web root. We will also rename the clumsy `squirrelmail-1.4.3a` folder to a more simple `mail` folder. You will need to have superuser `root` privileges in order to do this on most systems.

```
# mv squirrelmail-1.4.19 /var/www/html/mail
# cd /var/www/html/mail
```

Here we have used the name `mail`, so the URL that users will use will be `http://www.sitename.com/mail`. You can choose another name, such as `webmail`, and use that directory name instead of `mail` in the commands that you enter.

It is also useful and secure to create a `data` directory for SquirrelMail that is outside the main web root, so that this folder will be inaccessible from the Web.

```
# mv /var/www/html/mail/data /var/www/sqmdata
```

It is important to make this newly created folder writable by the web server. To be able to do this, you must know the user and group that your web server runs under. This may be `nobody` and `nobody`, `apache` and `apache`, or something else. You will want to verify this; it will be listed in your `httpd.conf` file as the `User` and `Group` entries.

```
# chown -R nobody:nobody /var/www/sqmdata
```

Finally, we will create a directory to store attachments. This directory is special in that, although the web server should have write access to write the attachments, it should not have read access. We create this directory and assign the correct permissions with the following commands:

```
# mkdir /var/www/sqmdata/attachments
# chgrp -R nobody /var/www/sqmdata/attachments
# chmod 730 /var/www/sqmdata/attachments
```

SquirrelMail has now been properly installed. All of the folders have been set up with correct permissions that will secure intermediate files from prying eyes.

If a user aborts a message that contains an uploaded attachment, the attachment file on the web server will not be removed. It is a good practice to create a cron job on the server that erases excess files from the attachment directory. For example, create a file called `remove_orphaned_attachments` and place it in the `/etc/cron.daily` directory. Edit the file to have these lines:

```
#!/bin/sh
#!/bin/sh
rm `find /var/www/sqmdata/attachments -atime +2 | grep -v
"\."| grep -v _`
```

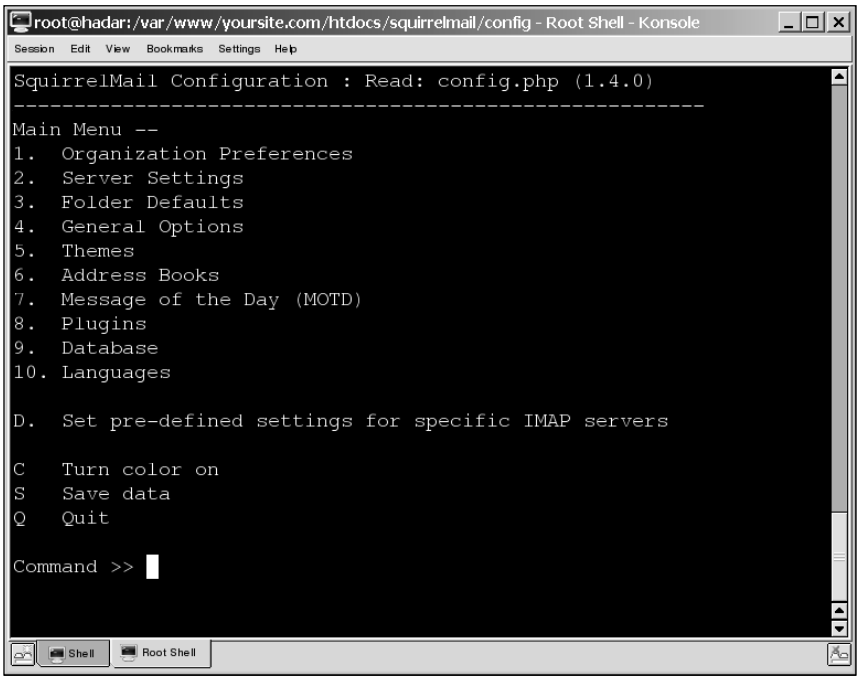
This will run daily and search the SquirrelMail attachments directory for files which are orphaned, and delete them.

Configuring SquirrelMail

SquirrelMail is configured through the `config.php` file. To aid the configuration, a `conf.pl` Perl script has also been provided. These files are located within the `config/` directory in the base installation directory.

```
# cd /var/www/html/mail/config
# ./conf.pl
```

Once you have run this command, you should see the following menu:



To select an item from the menu, enter the appropriate letter or number, followed by the *Enter* key. As SquirrelMail has been developed, it has been noticed that IMAP servers don't always behave in the same way. To get the most out of your setup, you should tell SquirrelMail which IMAP server you are using. To load a default configuration for your IMAP server, enter the **D** option and type the name of the IMAP server that you have installed. This book covers the Courier IMAP server, so you should choose that. Press *Enter* again, and you will return to the main menu.

We will be moving through the various subsections of the menu and configuring the appropriate options.

Type *1* and then press *Enter* to select the **Organization Preferences**. You will get a list of items you can change. You may wish to edit the **Organization Name**, **Organization Logo**, and **Organization Title** fields. Once you have modified these to your satisfaction, enter *R* to return to the main menu.

After this, type *2* to visit the **Server Settings**. This allows you to set the IMAP server settings. It is important that you update the **Domain** field to the proper value.

In our case, the **Update IMAP Settings** and **Update SMTP Settings** values should be correct. If you would like to use an IMAP or SMTP server that is located on a different machine, you may wish to update these values.

Press *R* followed by the *Enter* key to return to the main menu.

Next, type *4* to visit the **General Options**. You will need to modify two options in this section.

- Data Directory to be `/var/www/sqmdata`.
- Attachment Directory to be `/var/www/sqmdata/attachments`.
- Type in *R* followed by the *Enter* key to return to the main menu. Enter *S* followed by the *Enter* key twice to save the settings to the configuration file. Finally, enter *Q* followed by the *Enter* key to exit the configuration application.

We have finished configuring the SquirrelMail settings needed for basic operation. You may return to this script at any time to update any settings you have set. There are many other options to set, including those regarding themes and plugins.

SquirrelMail plugins

Plugins are pieces of software that extend or add functionality to a software package. SquirrelMail was designed from the ground up to be very extensible, and includes a powerful plugin system. Currently, there are over 200 different plugins available on the SquirrelMail website. They may be obtained at `http://www.squirrelmail.org/plugins.php`.

The functionality they provide includes administration tools, visual additions, user interface tweaks, security enhancements, and even weather forecasts. In the following section, we will first go over how to install and configure a plugin. After that, we'll go over some useful plugins, what they do, how to install them, and more.

Installing plugins

These SquirrelMail additions were designed to be simple to set up and configure. In fact, the majority of them follow exactly the same installation procedure. However, a few require custom setup instructions. For all plugins, the installation process is as follows:

1. Download and unpack the plugin.
2. Perform custom installation if needed.
3. Enable the plugin in `conf.pl`.

Example plugin installation

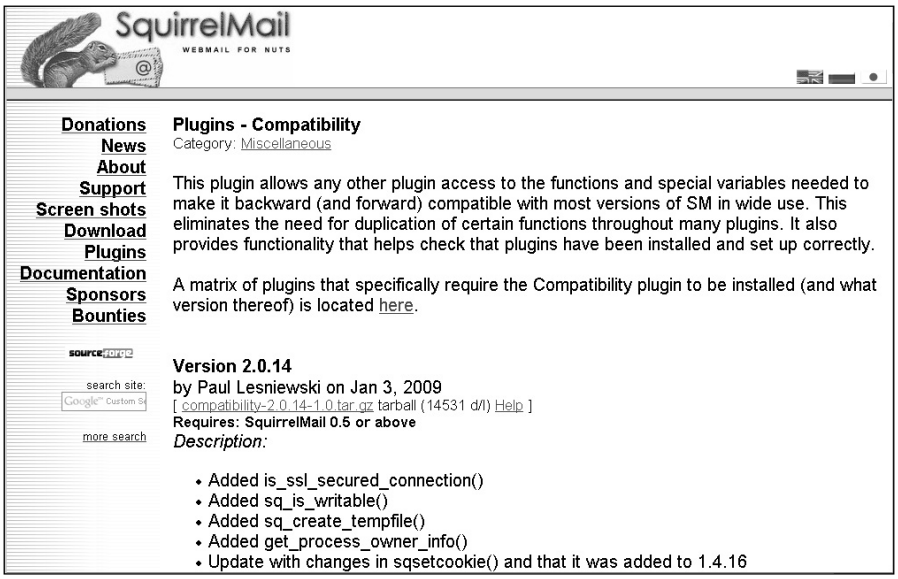
In this section, we will go over the installation of the **Compatibility plugin**. This plugin is required in order to install plugins created for older versions of SquirrelMail. No matter how bare-bones your installation, the Compatibility plugin will most likely be part of your setup.

Downloading and unpacking the plugin

All available plugins for SquirrelMail are listed on the SquirrelMail website at `http://www.squirrelmail.org/plugins.php`.

Certain plugins may require a specific version of SquirrelMail. Verify that you have this version installed. Once you have located a plugin, download it to the `plugins/` directory within the SquirrelMail root folder.

You may locate the Compatibility plugin by clicking on the **Miscellaneous** category in the plugins page on the SquirrelMail plugins web page. This page has a list of plugins in the **Miscellaneous** category. Locate Compatibility and click on **Details and downloads**, and then download the latest version.



Download tarball to your SquirrelMail plugin directory.

```
# cd /var/www/mail/plugins
# wget http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fwww.
squirrelmail.org%2Fplugins%2Fcompatibility-2.0.14-1.0.tar.gz
```

Once you have downloaded the plugin to the `plugins` directory, unpack it using the following command:

```
# tar zxvf compatibility-2.0.14-1.0.tar.gz
```

If a plugin of the same name has already been installed, its files may be overwritten. Verify that you either do not have a plugin of the same name, or save the files before you unpack the tarball.

Performing custom installation

The current version of the Compatibility plugin does not require any additional configuration. However, you should always check the documentation for a plugin, as certain other plugins may require custom installation. Once you have unpacked the plugin package, the installation instructions will be listed in the `INSTALL` file within the newly created `plugin` directory. It is advisable to check the installation instructions before enabling the plugin in the configuration manager, as some plugins may require custom configuration.

Enabling the plugin in conf.pl

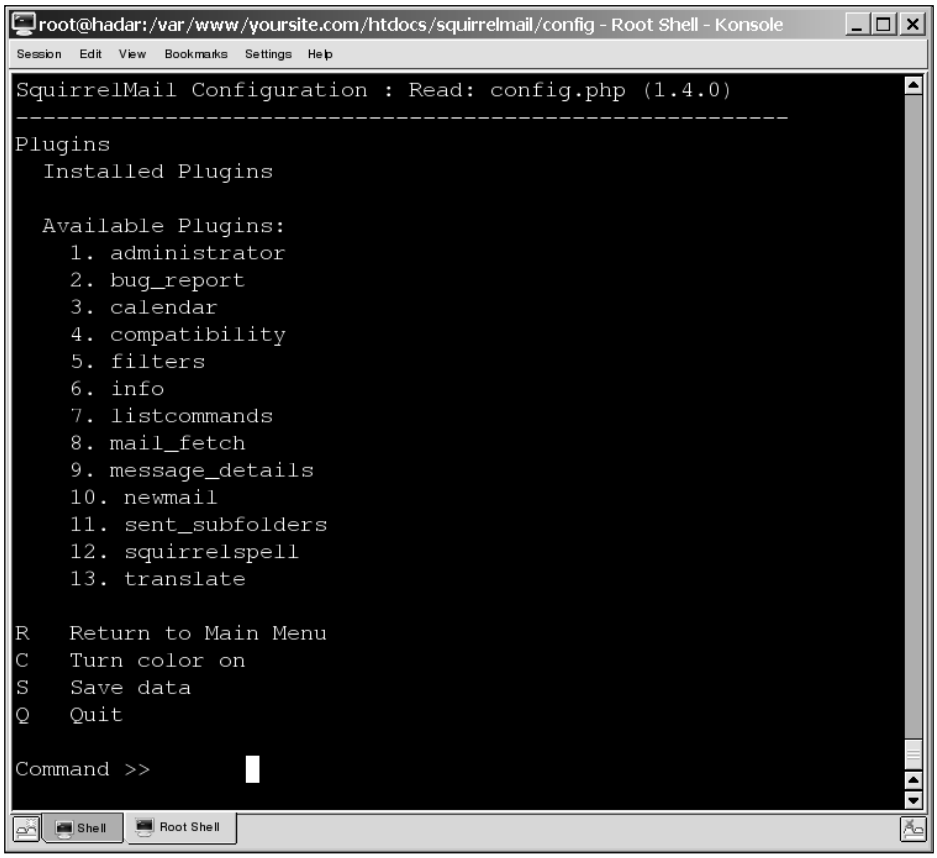
Within the main menu of the configuration editor, option number 8 is used to configure and enable plugins. Start `conf.pl` and select option **8**.

```
# cd /var/www/mail/plugins
# cd ../config
# ./conf.pl
```

```
SquirrelMail Configuration : Read: config_default.php (1.4.0)
---------------------------------------------------------
Main Menu --
[...]
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
[...]

Command >>
```

You should get the following display when you select this option for the first time:



All the plugins that have been installed and enabled are listed under the **Installed Plugins** list. All the plugins that have been installed but not enabled are listed under the **Available Plugins** list.

Once you have unpacked a plugin within the `plugins/` directory, it will show up under **Available Plugins**. As you can see in the previous figure, there are a number of installed plugins, but none of them are enabled. As a malfunctioning or wrongly configured plugin can cause SquirrelMail to stop functioning properly, it is advisable to enable plugins one by one, and verify that SquirrelMail works after each one.
To enable the Compatibility plugin, locate it in the list **Available Plugins** (in this case, number **4**) and press the *Enter* key. The Compatibility plugin is now installed. Plugins can be disabled by locating them in the **Installed Plugins** list and entering their number and pressing *Enter*.

Useful plugins

We'll now see some useful SquirrelMail plugins that you may consider installing.

The information has been compiled to provide a helpful reference while deciding whether to install a plugin. Each plugin contains four specific categories:

- **Category**: The category in which the plugin is listed on the SquirrelMail site
- **Authors**: Authors who wrote the plugin, in chronological order
- **Description**: A short description of the plugin's functionality
- **Requirement**: A list of prerequisites for the plugin's successful installation

Plugin name	Category	Author(s)	Description	Requirement
Compatibility plugin	Miscellaneous	Paul Lesneiwski	This plugin allows any other plugin access to the functions and special variables needed to make it backward (and forward) compatible with most versions of SM in wide use. This eliminates the need for duplication of certain functions throughout many plugins. It also provides functionality that helps in checking whether the plugins have been installed and set up correctly.	Nothing
Secure login	Logging in	Graham Norbury, Paul Lesneiwski	This plugin automatically enables a secure HTTPS/SSL-encrypted connection for the SquirrelMail login page if it hasn't already been requested by the referring hyperlink or bookmark. Optionally, the secure connection can be turned off again after successful login.	SquirrelMail version 1.2.8 or above, HTTPS/SSL-capable web server with encryption already working on your SquirrelMail installation.

Plugin name	Category	Author(s)	Description	Requirement
HTTP authentication	Logging in	Tyler Akins, Paul Lesniewski	If you keep SquirrelMail behind a password-protected directory on your web server and if PHP has access to the username and password used by the web server, this plugin will bypass the login screen and use that username/password pair.	SquirrelMail >= 1.4.0
Password forget	Logging in	Tyler Akins, Paul Lesneiwski	This plugin provides a workaround for the potential vulnerability of browsers, automatically storing usernames and passwords entered into web pages.	SquirrelMail >= 1.0.1
HTML mail	Compose	Paul Lesneiwski	This plugin allows users with IE 5.5 (and up) and newer Mozilla (Gecko-based browsers such as Firefox) browsers to compose and send their e-mail in HTML format.	SquirrelMail >= 1.4.0
Quick save	Compose	Ray Black III, Paul Lesneiwski	This plugin automatically saves messages as they are being composed, in order to prevent accidental loss of message content due to having browsed away from the compose screen or more serious problems such as browser or computer crashes.	SquirrelMail >= 1.2.9, the Compatibility plugin, JavaScript-capable browser

Plugin name	Category	Author(s)	Description	Requirement
Check quota usage (v)	Visual additions	Kerem Erkan	This plugin will check and display users' mail quota status.	SquirrelMail 1.4.0+; Compatibility plugin, version 2.0.7+, UNIX, IMAP or cPanel quotas installed and configured
Sent confirmation	Miscellaneous	Paul Lesneiwski	Displays a confirmation message after a message is successfully sent, as well as other features.	SquirrelMail >= 1.2.0, the Compatibility plugin
Timeout user	Miscellaneous	Ray Black III, Paul Lesneiwski	Automatically logs out a user if they are idle for a specified amount of time.	The Compatibility plugin
E-mail footer	Miscellaneous	Ray Black III, Paul Lesneiwski	This plugin automatically appends a custom footer onto the end of messages sent using SquirrelMail.	SquirrelMail >= 1.4.2
Change password	Change password	Tyler Akins, Seth E. Randall	Allows a user to change their password using PAM or Courier authentication modules.	SquirrelMail >= 1.4.0
Address book import-export	Address book	Lewis Bergman, Dustin Anders, Christian Sauer, Tomas Kuliavas	Allows the importing of address books from a **CSV** (**comma separated values**) file.	SquirrelMail >= 1.4.4
Plugin updates (v0.7)	Administrator's Relief	Jimmy Conner	Checks for updates to your currently running plugins.	SquirrelMail >= 1.4.2

Many other plugins exist that handle vacation messages, calendars, shared calendars, notes, to-do lists, exchange server integration, bookmarks, weather information, and much more. Check the **Plugins** section in the SquirrelMail website for all of the available plugins.

Securing SquirrelMail

The SquirrelMail package, in and of itself, is fairly secure. It is well written and does not require JavaScript to function. However, there are a few precautions that may be taken to allow SquirrelMail to run as a secured mail handling solution.

- **Have an SSL connection**: By using an SSL connection, you may be certain that all communications will be encrypted, and so usernames, passwords, and confidential data cannot be intercepted during transmission. This may be accomplished through the installation of the **Secure Login plugin**. Obviously a web server configured for secure SSL access will also be required; certificates will most likely need to be generated or acquired.
- **Time out inactive users**: Users may leave themselves logged in and neglect to log out once they are finished. To fight this, inactive users should be logged out after a certain amount of time. The **Timeout User plug-in** accomplishes this.
- **Fight "Remembered Passwords"**: Many modern-day browsers offer to remember a user's password. Although a convenience, this may be a large security vulnerability, especially if the user is located at a public terminal. To fight this, install the **Password Forget plugin**. This plugin will change the names in the username and password input fields, to make it more difficult for a browser to suggest them to future users.
- **Do not install security-compromising plugins**: Plugins such as **Quick Save**, **HTML Mail**, and **View As HTML** may compromise security.

Summary

Now that you've finished this chapter, you should have a working SquirrelMail installation as well as a greater understanding of the benefits and disadvantages of a webmail solution. You should be familiar with the benefits and drawbacks of a webmail solution. The benefits include remote access, a single central point to be maintained, and simpler testing; while disadvantages include potential performance problems and the security risk of allowing remote access from potentially compromised computers.

You are now aware of the main features of SquirrelMail, including its flexibility and the availability of plugins, along with what the prerequisites for installing SquirrelMail are, and how to identify if they are already installed.

You also have learned how to configure SquirrelMail, including locating, installing, and configuring plugins. You have been walked through the installation of a key plugin; the Compatibility plugin. Several other useful plugins have also been introduced. Finally, you have learned about some ways to improve the security of SquirrelMail, including web server configuration and some appropriate plugins.

Where to buy this book

You can buy Linux E-mail Second Edition from the Packt Publishing website:
`http://www.packtpub.com/linux-email-2nd-edition/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

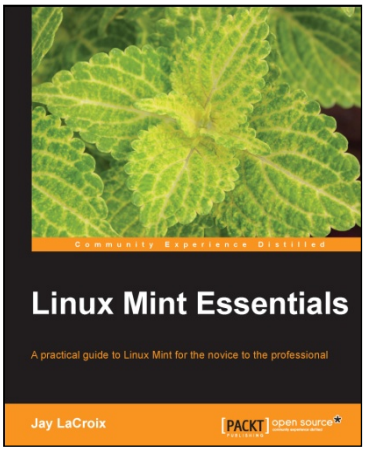
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

[PACKT] open source
community experience distilled
PUBLISHING

Linux Mint Essentials

Jay LaCroix



Chapter No. 3
"Getting Acquainted with Cinnamon"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.3 "Getting Acquainted with Cinnamon"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Jay LaCroix is a Linux Administrator with over 12 years of experience and nine certifications. He is a technologist who enjoys all things tech, including (but not limited to) hardware, software, servers, networking, and development. When Jay is not buried in a plethora of computer books, he enjoys photography, music, gaming, and writing. Jay is passionate about open source software, especially Linux, and its long-term adoption.

Jay is also the proud author of the self-published Sci-Fi novel, *Escape to Planet 55*.

> To my dad, Bill; my sons, Alan and Johnny; my brother, Gordon; my sisters, Cheri, April, and Christina; as well as their children; my dear friends, Krys and Jim; and all of the men and women who spend countless hours volunteering their time to make open source the best software on Earth.

Linux Mint Essentials

Welcome to the world of Linux Mint! With this book as your guide, you'll explore this exciting Linux distribution from its installation all the way to its administration and maintenance. Geared toward the Linux novice, this book will build skills that will not only help you use Linux Mint for your day-to-day computing tasks, but also build a foundation on which you can expand your knowledge. Whether you simply want to benefit from a bird's-eye view of Linux Mint or get started on the road to becoming a Linux admin, this book will help you get there. Along the way, we'll work through how to complete day-to-day tasks such as creating/managing files and documents, and we'll also work on configuring our Mint installation, managing packages, connecting to networks, increasing security, adding/removing users, troubleshooting, and more!

What This Book Covers

Chapter 1, *Meet Linux Mint*, discusses what Linux Mint is and what sets it apart from other distributions. We'll also talk about some reasons you'd want to choose Linux in the first place.

Chapter 2, *Creating Boot Media and Installing Linux Mint*, will walk you through the process of installing Linux Mint on your computer. Several methods of installation, such as bootable DVD and bootable flash drive, are covered in this book, and you'll also learn about some of the best practices for the installation of Linux Mint, including tips on partitioning your hard disk.

Chapter 3, *Getting Acquainted with Cinnamon*, discusses Cinnamon, a fresh and exciting desktop environment (a graphical user interface) that is taking the Linux community by storm. In this chapter, we'll tackle this interface head-on.

Chapter 4, *An Introduction to the Terminal*, will explain how to navigate the filesystem, execute commands, search for fi les, and even work through an introduction to scripting. Although using a Terminal is not required in order to use Mint, learning the basics of the terminal will further empower your skills.

Chapter 5, *Utilizing Storage and Media*, discusses how to work through the examples of accessing various types of media in Mint. The examples shown in this chapter include formatting and mounting removable storage, along with analyzing disk usage, burning CDs and DVDs, and utilizing Mint's USB Image Writer.

Chapter 6, *Installing and Removing Software*, discusses how to work through the examples of installing and removing software on our Mint installation, as it features a large repository of free software packages. Also, several different methods of software management will be covered, with examples of both graphical programs and terminal commands.

Chapter 7, *Enjoying Multimedia on Mint*, is all about enjoying multimedia on Mint. This chapter covers features such as listening to MP3s, ripping audio CDs, editing audio tags, watching DVDs, and more!

Chapter 8, *Managing Users and Permissions*, talks about users and permissions. You'll learn how to create/remove users and groups, as well as how to configure user access to administrative commands with sudo.

Chapter 9, *Connecting to Networks*, is all about networking. Concepts such as wired and wireless networking will be covered, as well as accessing your machine via SSH and also how to share files.

Chapter 10, *Securing Linux Mint*, will work on hardening our Linux Mint system with concepts such as choosing strong passwords, encrypting your home folder, blocking access to specific websites, and even backing up and restoring important data.

Chapter 11, *Advanced Administration Techniques*, will cover advanced concepts for managing your installation. In this chapter, setting up cron jobs, moving to new Mint releases and killing processes, and monitoring resources will be covered.

Chapter 12, *Troubleshooting Linux Mint*, concludes our journey with Mint by providing certain tips and tricks for what to do when things go wrong. In this chapter, you'll learn about dealing with problems such as booting issues, audio and networking woes, as well as how to access system logs for troubleshooting.

Appendix A, *Reinstalling Mint while Retaining Data*, discusses a technique on how to move from one release of Linux Mint to another, as Linux doesn't really feature a direct utility for you to do this.

Appendix B, *Using the MATE edition of Linux Mint*, discusses another edition of Linux Mint, MATE. In this appendix, we'll explore the various specific features of the MATE edition, which runs better on older hardware.

Appendix C, *Using the KDE edition of Linux Mint*, discusses another popular desktop environment, and Mint features it as the default desktop edition. In our final appendix, we'll explore the KDE Mint flavor.

3
Getting Acquainted with Cinnamon

By now, you should have a fully functional installation of Linux Mint ready to do your bidding. Whether you have already installed the distribution or you are running it from live media, Linux Mint is at your command. Right out of the box, you can browse the web, create and manage files, listen to music, watch movies, and even connect to and administer other machines. In the default installation, Mint includes everything you need to be productive. In this chapter, we'll explore the most popular Mint desktop environment (**Cinnamon**) and how to use and customize it.

In this chapter, we will discuss the following topics:

- What is Cinnamon?
- Logging in to Cinnamon
- Launching programs
- Task management
- Workspaces
- Notifications
- Creating launchers
- Bundled applications
- File management with Nemo
- Configuring Cinnamon settings
- Changing the default search engine in Firefox
- Changing the themes of the desktop

Getting familiar with Cinnamon

Cinnamon is a **desktop environment**. This is the term that the Linux community uses to describe a user interface thrown on top of the Linux kernel. With Linux, you don't actually need a desktop environment. In the case of Linux servers, it's not uncommon to see them with no user interface at all; instead, the administrator would rely on shell commands to configure and interact with a system. In fact, it's even possible to perform all the basic desktop functions (such as modifying files, listening to music, and browsing the web) using shell commands. These commands call programs that can run without a user interface. However, when using Linux on your desktop or laptop, installing a desktop environment makes things much simpler. Most distributions (such as Mint) include a desktop environment in the default installation. Nowadays, Linux desktop environments have become so efficient that terminal commands are no longer a necessity; you can operate your computer with the comfort of your traditional mouse just like you would with Mac OSX or Windows.

Cinnamon is not the only desktop environment available for Linux. As mentioned earlier, there are others such as GNOME, KDE, MATE, and Xfce. Each desktop environment offers a different style of interacting with your computer graphically. Some may enjoy the eye candy that KDE provides; others may prefer the simplicity of Xfce, while those that use virtual workspaces heavily may enjoy GNOME. If you don't like one user interface, you can always try another one. Workspaces will be discussed later in this chapter.

Cinnamon is a desktop environment that tries to cater to all types of users. There is plenty of eye candy (such as KDE); it offers a great support for workspaces (such as GNOME), runs fast (like Xfce), and has a few tricks of its own. Due to its popularity, it's unofficially assumed to be the default desktop environment of Mint. Many of the same developers of Mint work on it even though Cinnamon is actually not exclusive to Mint.

In fact, Cinnamon is actually a fork of GNOME 3.x. When the 3.x series of the GNOME desktop was released for Linux, many users were displeased due to its radical departure from how the environment functioned in the 2.x series. Cinnamon was built on top of GNOME 3.x, but changed dramatically to become its own environment. As of Cinnamon 2.0, it's now completely separate from GNOME, though its origin remains.

The following screenshot shows off the Cinnamon desktop, which you'll see right after logging in. We will explore its various functions in the following sections of this chapter.

Logging in to Cinnamon

When your Linux Mint computer has completed the start-up procedure, the first thing you'll see is the **MDM** (**Mint Display Manager**), which will allow you to log in to the system by providing the username and password that you created during the installation.

If you choose the option for automatic login during installation or if you are running Mint from live media (such as a USB stick or DVD), the MDM screen will be bypassed and you'll immediately be logged in to Cinnamon. If this is the case, feel free to move on to the next section and come back to this one if you need a run through of how the MDM functions.

At first, the only user account you'll be able to log in with is the one that you created during installation. In *Chapter 8*, *Managing Users and Permissions*, the process of creating additional users will be explained.

When the MDM first appears, you will be shown a list of users on the left-hand side, and you will have an opportunity to type in your user name and then press *Enter* to begin the login process. If your hand is already on your mouse, it may be quicker just to click on the desired username on the left-hand side rather than typing in the username manually. Next, you will need to provide your password when the system will ask for it, and then you can either click on **OK** or press *Enter* to begin logging in to the system.

While this is all you really have to know in order to access your system, the MDM has a few additional features as well. As we've discussed, there are more desktop environments available other than just Cinnamon. However, one thing that is not yet mentioned is that you can actually install more than one environment at a time by simply installing the required packages to install another desktop environment. Installing additional programs is covered in *Chapter 6*, *Installing and Removing Software*.

If you have any additional desktop environments installed, you can choose the one that you'd like to use on the MDM screen prior to logging in. To do so, click on the middle icon on the lower-left side of the desktop in between the power icon and flag. When you do so, you will be given a selection of which desktop environment to use. For example, you could use Cinnamon as your main interface, but also install Xfce to use from time to time.

Launching programs

Once you're logged in to Cinnamon, you're able to launch applications and start working. On the bottom-left side of the Cinnamon desktop, you'll see **Menu**, titled appropriately enough, next to an icon that looks like a gear. Clicking on this will launch Cinnamon's application menu, as shown in the following screenshot:

If you have a Windows logo key on your keyboard, you can press this key to immediately launch the application menu without having to use your mouse.

The application menu in Cinnamon is not a simple menu; it's full of features designed to make it easy to find the items that you want. For example, if you already know the name of the application that you want to launch, you can start typing its name in the search box at the top of the window. This will narrow down a list of applications as you type. In addition, the **Recent Files** section will store the files that you've been working on lately, so you can get right back to work. Similarly, **Places** stores the most recent folders that you've accessed.

The middle section of the application menu is a list of applications broken down by category. The first entry, **All Applications**, shows every graphical application installed on the system, though other entries such as **Graphics** and **Office** show more specific results. For example, if you're looking for **Libre Office Writer** (a word processor) you'll find it either in **All Applications** or **Office**.

For advanced users, if you know the command for the application you'd like to launch, you can save some time by pressing *Alt* + *F2* on your keyboard. This will open a box in which you can type a command. Type the name of the application (for example, `firefox`), and it will open straight away. In most cases, the command for an application is its name in lowercase characters.

On the left-hand side of the menu, you'll see a list of icons. These are applications that have been saved as favorites, thus allowing you quick access to the programs that you use the most. By default, Firefox, Software Manager, System Settings, XChat, Terminal, and Nemo are saved as favorites and are immediately visible on the left-hand side of the menu. If you right-click on an application within the menu, you'll have an **Add to favorites** option to add it on the left pane of the menu along with the others. If you'd like to remove an application that is already listed as a favorite, locate that icon within the menu, right-click on it, and select **Remove from favorites**.

With each application that you add to your favorites, the application menu will grow taller. Keep this in mind as you add favorites, so the menu doesn't grow to an uncomfortable size.

Finally, the last three icons on the bottom-left corner of the desktop screen allow you to lock your session, log out, and shut down, respectively.

Monitoring tasks

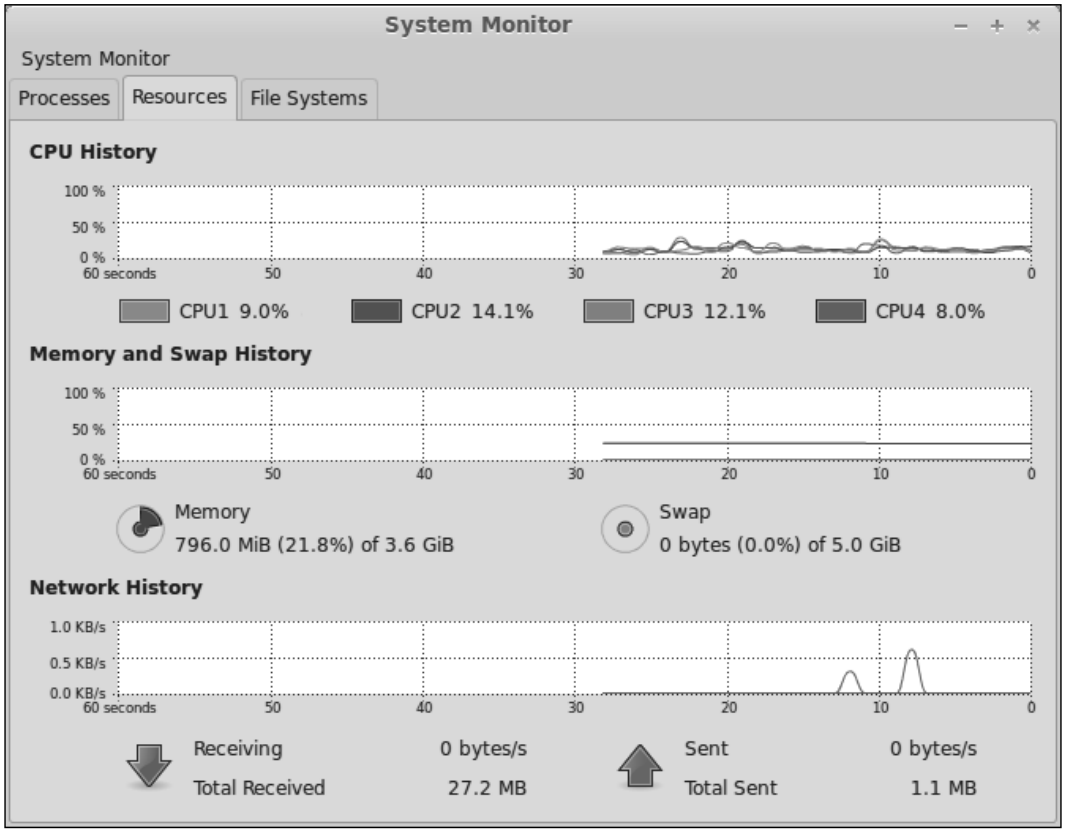
Managing running programs in Cinnamon is very similar to other user environments. Just like Mac OSX and Windows, there is a close, maximize, and minimize icon on the edge of the window border. On the bottom of the screen is a panel that shows a list of running applications as well as the date/time and messages from individual applications.

You may notice a few standalone icons on the left-hand side, next to the **Menu** icon. These are pinned applications similar to the quick-launch area of the Microsoft Windows taskbar. Here, you can store launchers for your favorite applications. By default, there is a Show Desktop button and program icons for Firefox, launching a Terminal, and opening Nemo. If you'd like to remove any of these, simply right-click on them and you'll have the option to do so. To add new pinned applications, right-click on the desired application within the application menu and click on **Add to panel**.

The typical use case of the Cinnamon desktop consists of a user launching an application from the application menu (and pinning it if desired). This creates an entry for the running program in the panel. Then, the user can minimize the application to free screen space or close it. The running applications will be listed in the panel in the order in which they were launched.

Another method of cycling through open applications is known as **Scale Mode**. To activate it, press *Alt* + *Ctrl* + the down arrow on your keyboard. When you do so, your desktop will zoom out showing you a bird's-eye view of the applications that are currently open. Next, you can either click on the application you'd like to bring to the front or press *Esc* to exit the menu.

From time to time, you may want to take a look at the applications that are running on your system and their impact on resources such as CPU or RAM usage. For this purpose, Mint includes **System Monitor** that you can use to not only check resources' usage but also to close misbehaving applications and see which programs are being the greediest. An example of the **System Monitor** is pictured in the following screenshot:



To access the **System Monitor**, open the application menu and you'll find it under **System Tools**. Feel free to pin it to the panel or the application menu for quick access later. One example of the usefulness of the **System Monitor** is the following scenario. Imagine you're not working with any intensive application, but mysteriously, the fan on your computer starts running abnormally high. You could then check the **System Monitor** to easily determine which application is using the most of your CPU. Then, you'll know which application to focus your troubleshooting on.

Utilizing workspaces

So, what do you do when you have too many applications open? One of the most popular elements of most Linux desktops is the concept of **workspaces**. When your screen becomes full of applications, it can become hard to manage. Thankfully, you can separate applications into different workspaces, which are essentially additional Cinnamon screens that you can work with.

By default, Cinnamon has two workspaces available. To see this concept in action, simply move your mouse to the upper-left corner of your screen. This activates **Expo Mode**, which allows you to view and switch between your workspaces. At first, you should see two workspaces. The first is the one that you've been using all along; however, you'll also see a blank Cinnamon interface ready for your use. If you click on the second (blank) interface, you're brought into an entirely different workspace that is a blank slate. You can then launch applications inside this second workspace. These applications are not shown on the same screen as those that were running on the first workspace. You can create additional workspaces by clicking on the **+** icon on the left-hand side of the **Expo** screen. You can close existing workspaces by pointing to them and clicking on the **x** icon that will appear.

You can also enter Expo mode by pressing *Ctrl* + *Alt* + the up arrow on your keyboard.

By default, the workspaces are displayed horizontally. This is fine if you only have a few workspaces to cycle through. However, once you start adding a bunch, you'll notice that it can be hard to see them all as the Expo screen zooms out with each workspace you create. To remedy this, try the following steps:

1. First, open **System Settings** (available in the application menu) and then switch to advanced mode by clicking on the link at the bottom of the window.
2. Next, click on the Workspaces icon and enable the **Display Expo view as a grid** option. You should notice the difference the next time you activate the Expo screen. If you plan on using a large number of workspaces, you may find this layout easier to follow.

Once an application is running, it's not glued to the workspace that you opened it in. If you'd like to move an application that is already open to another workspace, you can easily do so via one of the following two methods:

- The first method is to right-click on the open application's entry on the panel, and then you can click on **move to left workspace** or **move to right workspace**. This will immediately move the application one workspace to the left or right.
- Alternatively, you can also right-click on the window border (the top edge of the application window), which will have the same options as mentioned in the preceding method.

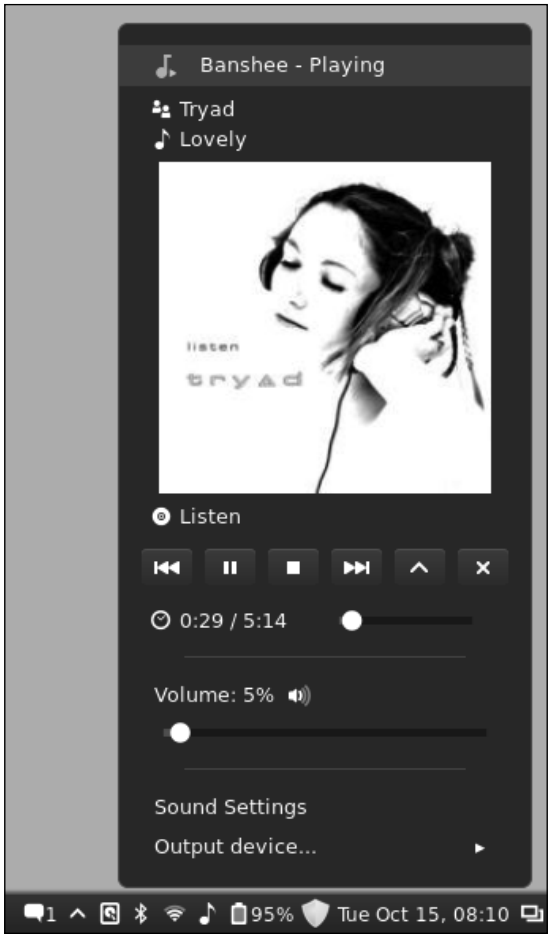
Notifications

At various times, you'll see several notifications while you use Mint. For example, you may see notifications such as updates are available to be installed, removable media has been inserted, how much battery power is remaining, or a wireless network has become available. Whenever an event occurs, the Cinnamon desktop will immediately display a notification in order to let you know.

Cinnamon will notify you in one of two places when a noteworthy event occurs. For example, you may see a notification bubble on the top-right corner of the screen when a wireless network becomes available or your machine is disconnected from a connection. If you miss a notification, don't worry. Each time a notification appears, it is stored in the panel for viewing later, underneath an icon that looks like a speech balloon, which is shown as follows:

Removable media notifications are handled a bit differently. If you insert media, such as a flash drive or DVD, a notification will not appear on screen but will be immediately available in the panel. By default, the contents of removable media will immediately open in the file manager (Nemo). Notifications for removable media are stored underneath a separate icon, shown as follows, which looks similar to a hard disk:

There is a series of notifications for audio as well. On the panel, there is an icon for controlling the volume, which you can adjust either by clicking on it and adjusting the slider or hovering your mouse pointer over it and moving the scroll wheel. If you are playing audio (for example, listening to MP3 files in Banshee), the icon will turn into a musical note instead, but you'll still be able to adjust the volume in the same way. However, when you click on the volume icon while the music is playing, you'll see a section used to control music in addition to the controls that are normally available. The following screenshot shows the Cinnamon notification area (with the audio icon clicked on) while the music is playing:



Creating launchers

Some users may desire to have their favorite applications available on the desktop in the form of shortcut icons. The Cinnamon interface features two ways of creating launchers. These allow you to create icons to launch applications or commands.

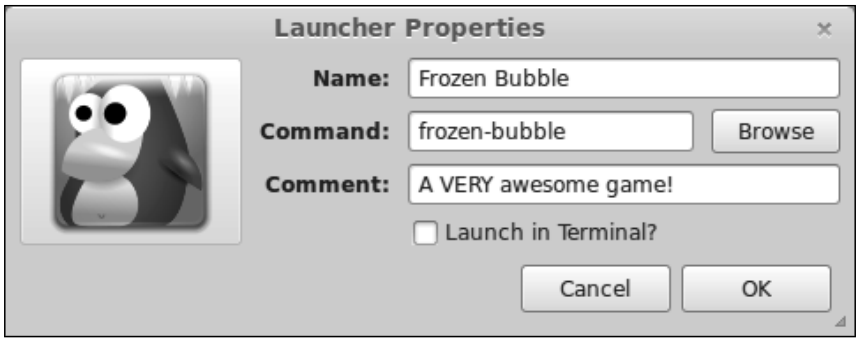
The easiest way to create a launcher is to find the application in the application menu and right-click on it. One of the options that appears in this menu is **Add to desktop**, which will create the icon for you. Then, you can drag the icons to arrange them as you like.

Additionally, you can also create a launcher manually. This is useful if you cannot find the application in the menu or you'd like to create a custom icon different from the one available in the menu; to do so, right-click on an empty portion of the desktop and click on **Create Launcher**. A window will appear with some fields for you to fill out in order to create a launcher. However, since you'll need to know the command used to launch the application, this may be a method catered more toward intermediate users. However, if you'd like to create a launcher to a file location (such as your `Documents` folder), this is best accomplished by this method.

The fields to fill out in order to manually create a launcher are as follows:

- **Type**: Choose whether the launcher will be an application, terminal application, or a location.
- **Name**: Provide the name of the application; you can type anything you want here.
- **Command**: Provide the command used to open the application. This is only visible while creating an application launcher.
- **Location**: Provide the location of the folder you want the launcher to point to. This is only visible while creating a location launcher.
- **Comment**: Provide a comment regarding the application or location. This is not required.

The following screenshot is an example of creating an application launcher:



Bundled applications

Although most of the applications bundled with Mint are not specific to Cinnamon, they are discussed here as each compliments the environment by providing a basic functionality. As discussed earlier, Mint includes just about everything you'll need to be productive immediately. Whether you want to browse the Web, check your e-mail, or watch movies, you're covered. In this section, we'll go through some of the noteworthy applications included out of the box. In *Chapter 6*, *Installing and Removing Software*, we'll run through the process of installing new applications, so you will get a chance to install some additional applications, discussed as follows, that will make your experience even more complete.

- **Firefox**: The default web browser in Mint is Mozilla Firefox, which is a great choice because it is cross platform (it's essentially the same Firefox that you can download for use with Windows and Mac OSX) and recognized in the industry. The main difference in Mint's version is that the process of changing the default search engine has been customized. We will discuss how to change the default search engine in Firefox later in this chapter.
- **Thunderbird**: Thunderbird is a cross-platform e-mail client, which will allow you to consolidate your e-mail accounts into one application. Nowadays, cloud e-mail solutions (such as **Gmail**) have largely replaced standalone applications such as Thunderbird. However, it's still very useful for ISP e-mail services and even Gmail itself can be accessed with it. If you prefer a standalone e-mail solution over a cloud-based solution, Thunderbird is for you.

- **Pidgin**: Chatting with instant messaging services (such as AOL Instant Messenger or Yahoo Chat) is a snap with Pidgin. Pidgin allows you to connect to all of your chat services in one application with a single contact list. Like Firefox and Thunderbird, Pidgin is also a cross-platform application. It's available on Windows as well.
- **Transmission**: Transmission is a client of **Bit Torrent**, one of the best services available for Linux. Bit Torrent itself is a very useful service that facilitates the transfer of large downloads. The Linux community uses Bit Torrent heavily for downloading large distribution ISO files (for example, the Crunchbang distribution can only be obtained this way). However, like most services created for the purposes of good, Bit Torrent is often abused in order to distribute illegal copies of paid applications and media as well. It's important to use responsibility and good judgment while using it.
- **XChat**: XChat is a full-featured client of **IRC** chat. While some may see IRC as an archaic technology, it's still very popular in the Linux community, so using it is recommended. For most (if not all) of the major Linux distributions, an IRC channel is available.
- **Libre Office**: Libre Office is a cross-platform productivity suite featuring a Word processor (**Writer**) as well as a spreadsheet application (**Calc**) and presentation application (**Impress**). Libre Office is a very capable Office suite on all the platforms; it's available on Linux, Mac OSX, and Windows, so learning it is highly recommended. By default, Libre Office saves files in open formats, though you can save files in Microsoft formats should you need to send documents to someone who uses Microsoft's Office suite.
- **GIMP**: **GIMP** (**GNU Image Manipulation Program**) is a free alternative to Adobe Photoshop. GIMP is very useful for editing, cropping, and manipulating photos and is a welcome addition to any graphic designer's tool set.
- **Simple Scan**: If you own a scanner, Simple Scan will facilitate your document-scanning needs. Simple Scan is easy to use, thus making things such as creating multi-page PDF files a cinch.
- **Banshee**: For those of you who have a collection of MP3 files, Banshee is a very capable music manager. With Banshee, you can not only listen to your MP3 files but also edit metadata, create playlists, listen to podcasts, and so on.

- **Brasero**: Brasero is a multipurpose media creator. If your computer has a rewritable DVD or CD drive, you'll be able to create music and data discs with this program. Brasero also allows you to create bootable CDs and DVDs from downloaded ISO files, so it is an important part of any Linux administrator's tool kit.
- **Software Manager**: A Mint-specific application, Software Manager is your gateway to discovering new applications. Although installing and removing applications is covered in *Chapter 6*, *Installing and Removing Software*, feel free to have a look around at the various categories of applications available. In addition, although Software Manager was developed by the Linux Mint team, it has found its way to other distributions since its debut.
- **Synaptic**: Synaptic is an application that does essentially the same thing as the Software Manager, but is catered more toward power-users. Synaptic is a tried-and-true package manager, having existed for well over 10 years. Intermediate to advanced users will likely prefer Synaptic over Mint's Software Manager.
- **Update Manager**: During the time in which a version of Mint is within its support cycle, security and feature updates are regularly released. Updates may include new versions of applications such as Firefox or even the Linux kernel itself. Although Linux is inherently secure, keeping it up to date is the best security practice recommended on any platform. Keeping your system up to date is discussed in *Chapter 6*, *Installing and Removing Software*.
- **Videos**: Videos is a generic video player application with a generic name. By default, all video files (clips, movies, and so on) stored on your hard disk will open with this program.
- **VLC**: VLC is also included for viewing video files. It's very similar to the Videos application, but much more capable and available on just about every platform in existence. There are few types of video files that won't open with VLC. In many ways, VLC is actually superior to the default Videos application.
- **Document Viewer**: Document Viewer allows you to view PDF files, which you would normally view using Adobe Reader on other platforms.
- **gThumb**: gThumb comes to the rescue when you need to view images. Not only does gThumb handle the viewing of images currently in your collection, it allows you to import new photos from a digital camera if you have one.

File management with Nemo

Every operating system has its method of managing saved files and browsing the filesystem; in Windows, it's File Explorer; in Mac OSX, Finder is used for this purpose; and in the case of Linux, there are many file managers. Each desktop environment has its own file management application. For example, the Xfce environment uses Thunar, KDE ships with Dolphin, GNOME features Files, and Cinnamon includes Nemo. There are others; however, we have a choice with Linux. In fact, it's not uncommon to see Linux users mix file managers or even install one that is completely separate, such as **Krusader** or **Midnight Commander**.

Nemo, Cinnamon's preference for file management, is a very capable file manager. With it, you can complete any task you'd normally perform in any other file manager. Copying, moving, renaming, and deleting files and folders is a breeze. In addition, you can browse network locations within Nemo as well. The following screenshot shows the Nemo file manager:

Browsing your filesystem within Nemo is as easy as clicking on objects to open them. In the main section of the window, you're presented with the contents of the current folder. Along the top of the window, you'll see the path you've navigated to, and on the left-hand side is a pane that shows the shortcuts to various locations.

If you insert a removable media (such as a disc or USB disk), it will automatically mount and show up in the left-hand pane of Nemo. To safely remove the attached media, click on the Eject icon that appears next to its heading on the left pane.

In the preferences menu, you can customize Nemo to your liking; perhaps, the default icon view isn't your favorite layout. You can view the content of folders as a list instead, which is similar to the Detailed List view in Windows Explorer. To access the preferences menu, click on **Edit** in the file menu and then select **Preferences**. The options here are self-explanatory, so feel free to adjust them to your liking and see what effect each setting has.

On the top-right side of each Nemo window are three additional icons you can use to adjust your view. The icons are shown as follows:

The first, which looks like a curved arrow, changes the location bar from an icon view (also known as **breadcrumbs**) to a text path that allows for keyboard input similar to an address bar in a web browser. The magnifying glass opens a menu that allows you to search folders for specific files should you forget where something is. Finally, the remaining three icons allow you to switch views without having to access the preferences menu.

Feel free to navigate around the filesystem and do some exploring; however, don't worry too much about what each of the individual folders is for just yet. We'll explore the filesystem in greater detail in the following chapter.

Configuring the settings of Cinnamon

Cinnamon is highly configurable; it's very easy to make it your own. You can customize everything from the theme all the way to power events such as choosing what happens when you close your lid (if you're using a laptop). Just about everything is customizable, thus making your installation of Mint truly your own. To get started with customizing your installation, open the **System Settings** application. You'll find it in the application menu under **System Tools**. By default, Mint has **System Settings** pinned on the left-hand side of the application menu for easy access. The following screenshot shows the Cinnamon **System Settings** application:

By default, not all categories are shown as **System Settings** will open in normal mode the first time you open it. In order to be able to access the complete array of settings, it's recommended that you switch to an advanced mode right away. To do so, click on the **Switch to Advanced Mode** link on the lower-left side of the **System Settings** window. You will see more categories appear instantly.

Next, we'll go through the most useful modules within **System Settings**, which are described as follows. Feel free to experiment to create your own perfect desktop.

- **Backgrounds**: Here, you can select a wallpaper for your desktop. A nice set of default backgrounds are included. To disable wallpapers altogether, expand **Advanced options** and change the picture aspect to **No picture**.
- **Effects**: By default, some of your video card resources are utilized to provide flashy effects during transitions. For example, with the effects enabled, minimizing a window will show it fading away rather than just simply disappearing from view. If you are on a slower system and need to conserve resources, disable this feature.
- **Themes**: Your entire desktop can be themed, thus changing its appearance completely. There are several items that can be individually themed to create your own look for the desktop. This process is explained later in this chapter.
- **Desktop**: Here, you can configure which icons are visible on the desktop. By default, the Computer and Home icons are visible as well as the icons for any removable media you may have inserted. In addition, you can also choose to show icons of the trash folder and available network servers.
- **Hot corners**: In this menu, you can configure what happens when you move the mouse into any corner of the screen. By default, the upper-left corner is configured to access the Expo mode. If you find yourself accidentally enabling the Expo mode frequently, you can disable it here (or simply use *Ctrl* + *Alt* + the up arrow instead). You can configure the other corners as well to activate Expo, Scale, or even activate a command if you wish. For example, you could configure Cinnamon to launch Firefox each time you move your mouse to the upper-right corner of the screen.
- **Networking**: In most cases, you won't access this module often. Here, you can configure networking (both wired and wireless) if you need custom settings. To connect to a wireless network, it's much easier to click on the wireless icon in the tray. However, in *Chapter 9*, *Connecting to Networks*, we will go over the networking functions in more detail, so it's a good idea to at least know where the settings can be found.
- **Power Management**: In this module, you can configure when to suspend the system. This is especially useful if you are using a laptop. For example, you may want the laptop to suspend (sleep) when the lid is closed.

Be very careful with the sleep settings on laptops. While it's a good idea to configure your laptop to sleep while not in use or when the lid is closed, make sure you also exercise good judgment. For example, only stow your laptop in your bag if you are *absolutely sure* that it has entered a suspended state. You can typically tell if a laptop is suspended by the activity of the LEDs, which may be in the form of a sleep indicator or a blinking power LED depending on the model. Placing a non-suspended laptop in your bag can easily cause it to overheat and suffer hardware failure as there is no airflow inside laptop bags. Not all laptops will turn themselves off when the temperature gets too hot.

- **Device Drivers**: Most of the time, Mint finds drivers that it needs for your specific hardware. In some cases, proprietary drivers may be available that may offer improved performance. A typical example of this is video cards. While support for video cards in Linux is great nowadays, sometimes the open source drivers may not be as functional as those available from the manufacturer. As a general rule, don't fix it if it's not broke.

  If you are having issues with your system (low frame rates in games, unable to access wireless networks, and so on), then you may try accessing this module to see if you have proprietary drivers available that will provide you with added functionality. Whenever possible, it's recommended to either use the drivers that ship with the distribution, as they have been thoroughly tested, or the open-source drivers, as the developers have access to the source code and so they can fix bugs.

  In regards to proprietary drivers, being closed source means that the Linux community has no visibility into the code to fix potential issues.

  Regardless of the overall opinion of proprietary drivers, it's important to make the decision that's best for you. If you need such drivers to make full use of your hardware, there's no reason why you shouldn't do so. This is especially true nowadays as resource-intensive gaming applications (such as Steam) have become available on the Linux desktop.

Changing the default search engine in Firefox

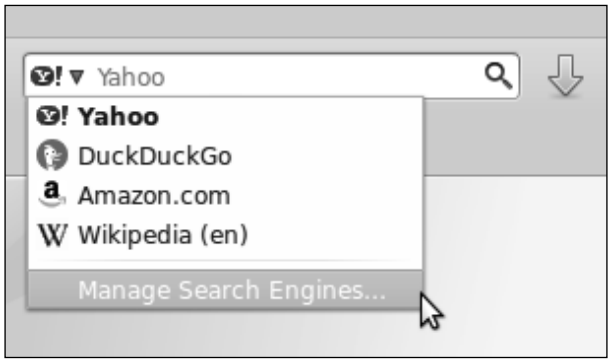
In the preceding section, it was mentioned that the version of Firefox included with Mint differs from the others in the way that the search engine settings are configured. In this section, we'll walk you through the process.

If you are already satisfied with Mint's default search engine (Yahoo), then there is no need for you to complete this activity. Feel free to skip it if you wish.

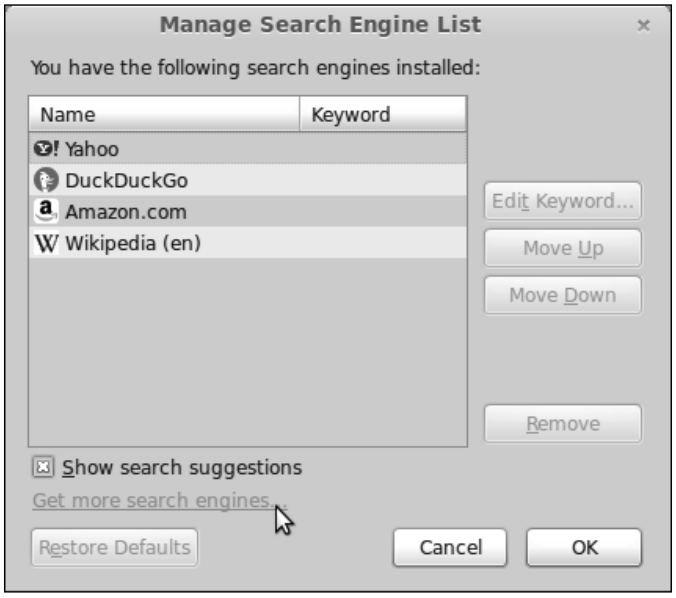
So, why does Mint use Yahoo as its default search engine? The main reason is revenue. Maintaining and developing a Linux distribution is a very expensive job. A lot of bandwidth is regularly consumed not only by those who are downloading the ISO image itself but also by the many updates that are released and regularly downloaded by its users. Yahoo shares revenue, which its users generate while searching online, with Linux Mint. This is one of the ways that Mint generates funding to keep itself going.

The process of changing the default search engine in Firefox is as follows:

1. First, locate the search box next to the address bar. There is a down arrow located in the search box. If you click on it, you'll see an option to **Manage Search Engines…**.



2. In the **Manage Search Engine List** window that appears, click on the **Get more search engines** link at the bottom of the window, as shown in the following screenshot:



3. A new tab will open to a customized Mint URL that explains the rationality behind changing the way in which search engines are managed in Mint. If you scroll down, you'll find icons for other search engines, such as Google, eBay, YouTube. Click on the one you want to add.
4. Another new tab will open. While this page is displayed, click on the down arrow in the search box again like you did in step 1. A new option to add the selected search engine will appear that was not there earlier.
5. The selection to the search provider you added will then be displayed in the search box. From this point forward, you can use this newly added search engine to conduct searches online.
6. If you'd also like to change the default search provider for address bar searches, the process is different. To do this, type `about:config` into the address bar.
7. Click on the button labeled **I'll be careful, I promise!**

8. In the search box, type `keyword.url`.
9. Double-click on **keyword.url**.
10. Change the search string in the dialog box to the one that matches the one for the provider you'd like to use. If you don't know what the search engine string is, you can find a list online. For example, type the following to make Google handle address bar searches:

    ```
    http://www.google.com/search?&q=
    ```

Changing themes

One of the greatest aspects of Cinnamon is how customizable it is. Nearly every aspect of the environment can be changed, including (but not limited to) the colors of applications, desktop wallpaper, and even the theme of the Cinnamon interface itself.

To start customizing your environment, use the following steps:

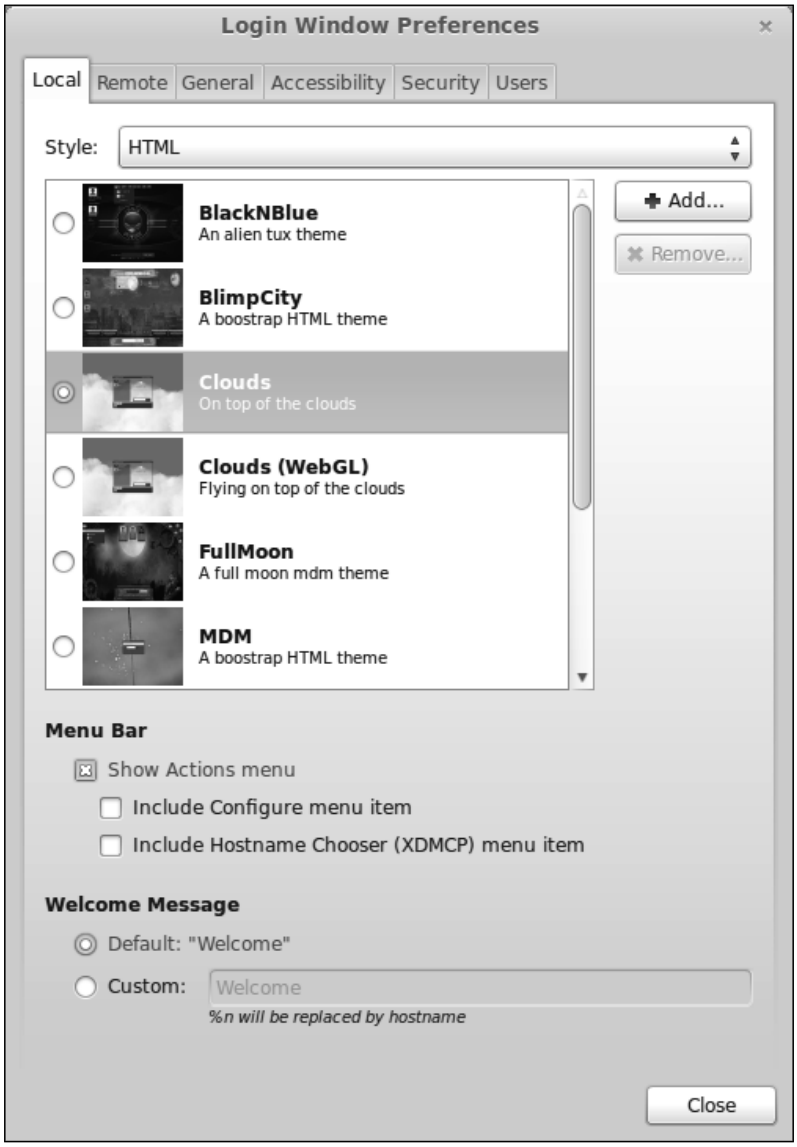
1. Open **System Settings** and locate the **Themes** section in the first row.
2. A new menu will appear with three tabs: **Installed**, **Get More Online**, and **Other Settings**.
3. The **Installed** tab shows which Cinnamon themes are currently installed. If you've never customized themes before, you'll only have the two default themes listed (the following two screenshots). There is a green check mark next to the currently active theme.
4. Feel free to switch to the **Cinnamon** theme to see the changes right away. The colors of the panel as well as the application menu will change. Make note of the fact that the color of application windows (such as Nemo) did not change.
5. In the next tab, **Get more online**, you can download new themes from Mint's `spices` repository. When it finishes refreshing, you'll see a list of new Cinnamon themes for you to download.
6. Feel free to download a few Cinnamon themes that look good to you. When you're done installing the themes, switch back to the **Installed** tab and you'll see your newly downloaded themes listed there.
7. Activate one of your newly downloaded themes by double-clicking on it. Notice that the Cinnamon interface is now using your newly downloaded theme.

The last section, **Other settings**, allows you to theme components other than Cinnamon. Unfortunately, there are no integrated means to download themes for other components like you can for Cinnamon. Although we'll cover installing new software in *Chapter 6*, *Installing and Removing Software*, you can find new themes in the **Software Manager** by simply searching for `themes`. The type of themes you're looking for are known as **GTK** themes, which set the themes for individual applications, and **Metacity** themes, which allow you to change the window borders of applications.

If you have downloaded additional **GTK** or **Metacity** themes, you can select them in the **Other settings** tab. To illustrate how different these themes can make applications appear, the following screenshots shows the screens before and after applying the theme. The following screenshot shows Nemo with the default Mint-X theme applied to it:

The following screenshot shows Nemo with a custom desktop theme applied to it:

In addition, the theme for the MDM display manager can be changed. As a reminder, the MDM is the login screen that you see when you first start your Mint computer. By default, there are quite a few themes for the MDM that you can choose. You can change the MDM theme by accessing the login screen section of **System Settings**. The following screenshot shows off this configuration menu:



Summary

In this jam-packed chapter, the Cinnamon desktop environment was covered in depth. First, we explored what Cinnamon is and how it fits in with other desktop environments such as GNOME and KDE. Next, we discussed logging in to the environment and how to launch applications. We also covered task management, switching between workspaces, and notifications. In addition, some of the bundled applications were listed, Nemo was featured, and then we configured the Cinnamon settings and themes.

In the following chapter, we'll get started with executing shell commands in the terminal to boost your knowledge even further. We'll cover how to access the Linux shell, manage files without a GUI, navigate the filesystem, and more!

Where to buy this book

You can buy Linux Mint Essentials from the Packt Publishing website: `http://www.packtpub.com/linux-mint-essentials/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

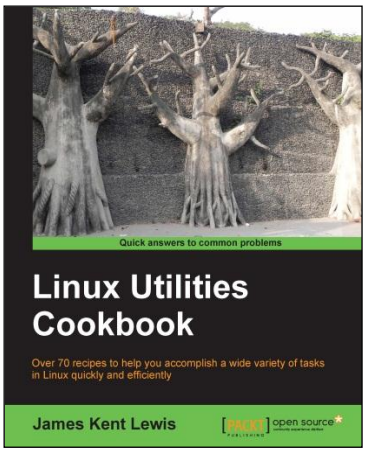
www.PacktPub.com

For More Information:
www.packtpub.com/linux-mint-essentials/book

[PACKT] open source
community experience distilled
PUBLISHING

Linux Utilities Cookbook

James Kent Lewis



Chapter No. 2
"The Desktop"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.2 "The Desktop"

A synopsis of the book's content

Information on where to buy this book

About the Author

James Kent Lewis has been in the computer industry for over 30 years. He started out writing BASIC programs in high school and used punch cards in college for his Pascal, Fortran, COBOL, and assembly language classes. Jim taught himself the C programming language by writing various utilities, including a fully-functional text editor, which he uses everyday. He started out using DOS and AIX, and then OS/2. Linux is now his operating system of choice.

Jim has worked in the past for several companies, including IBM, Texas Instruments, Tandem, Raytheon, Hewlett-Packard, and others. Most of these positions dealt with low-level device drivers and operating system internals. In his spare time, he likes to create video games in Java.

Jim has written articles for IBM Developer Works and has one patent.

First, I would like to thank Red Hat for creating a great OS. I used Fedora 17 to develop this book and it worked flawlessly. I would also like to thank my brother David for letting me bounce ideas off of him. Last, but certainly not least, I would like to thank my girlfriend, Gabriele. Her patience was greatly appreciated, and she also helped by lending me her Ubuntu laptop from time to time.

Linux Utilities Cookbook

Linux Utilities Cookbook shows how to solve typical problems on a Linux computer. The information is provided in a "recipe format" allowing the user to find desired topics quickly and efficiently. The steps to perform a task are clearly explained and have been tested for accuracy. There is also a section on shell scripting.

What This Book Covers

Chapter 1, *Using the Terminal / Command Line*, covers how to get the most out of the Linux command line.

Chapter 2, *The Desktop*, introduces some of the desktop environments available for Linux.

Chapter 3, *Files and Directories*, explains fi les, directories, and how to manage them.

Chapter 4, *Networking and the Internet*, covers connectivity and how to fix it when it goes down.

Chapter 5, *Permissions, Access, and Security*, gives a brief overview of Linux security features.

Chapter 6, *Processes*, explains how to manage processes in Linux.

Chapter 7, *Disks and Partitioning*, gives a brief insight into disk management.

Chapter 8, *Working with Scripts*, covers how to write scripts in Linux.

Chapter 9, *Automating Tasks Using Cron*, explains how to run jobs automatically.

Chapter 10, *The Kernel*, introduces how to make a custom kernel for your system.

Appendix A, *Linux Best Practices*, shows how to set up and run your systems like a pro.

Appendix B, *Finding Help*, covers locating the information you need quickly.

2
The Desktop

In this chapter we will cover these desktop environments:

- GNOME 2
- KDE desktop
- xfce
- LXDE
- Unity
- Mate

Introduction

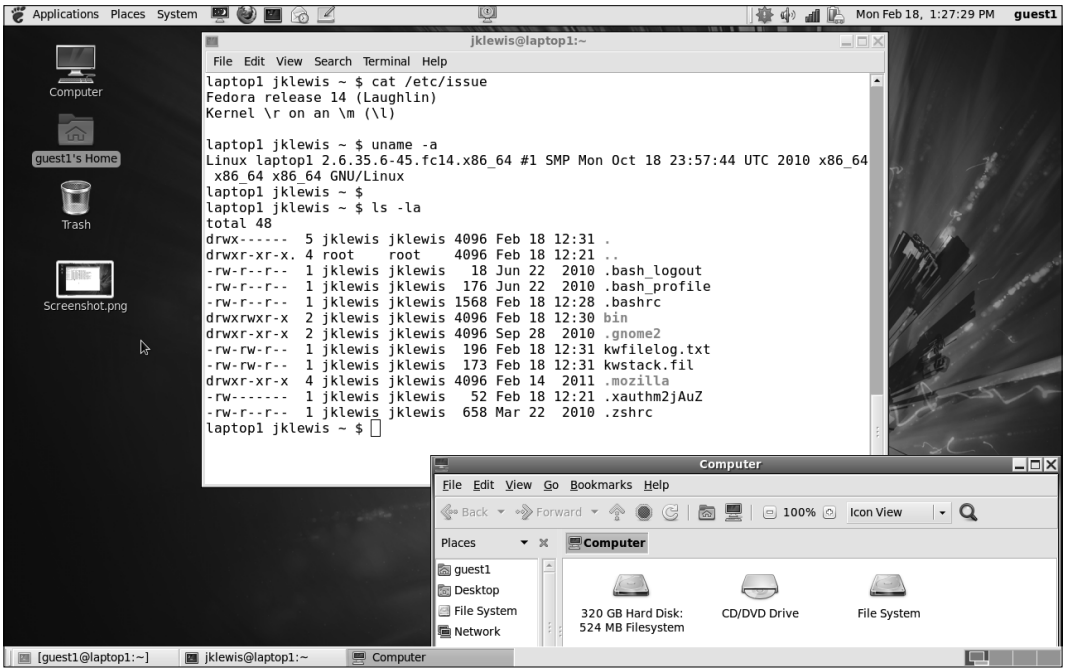
A computer desktop is normally composed of windows, icons, directories/folders, a toolbar, and some artwork. A window manager handles what the user sees and the tasks that are performed. A desktop is also sometimes referred to as a **graphical user interface** (**GUI**).

There are many different desktops available for Linux systems. Here is an overview of some of the more common ones.

GNOME 2

GNOME 2 is a desktop environment and GUI that is developed mainly by Red Hat, Inc. It provides a very powerful and conventional desktop interface. There is a launcher menu for quicker access to applications, and also taskbars (called **panels**). Note that in most cases these can be located on the screen where the user desires.

The screenshot of GNOME 2 running on Fedora 14 is as follows:

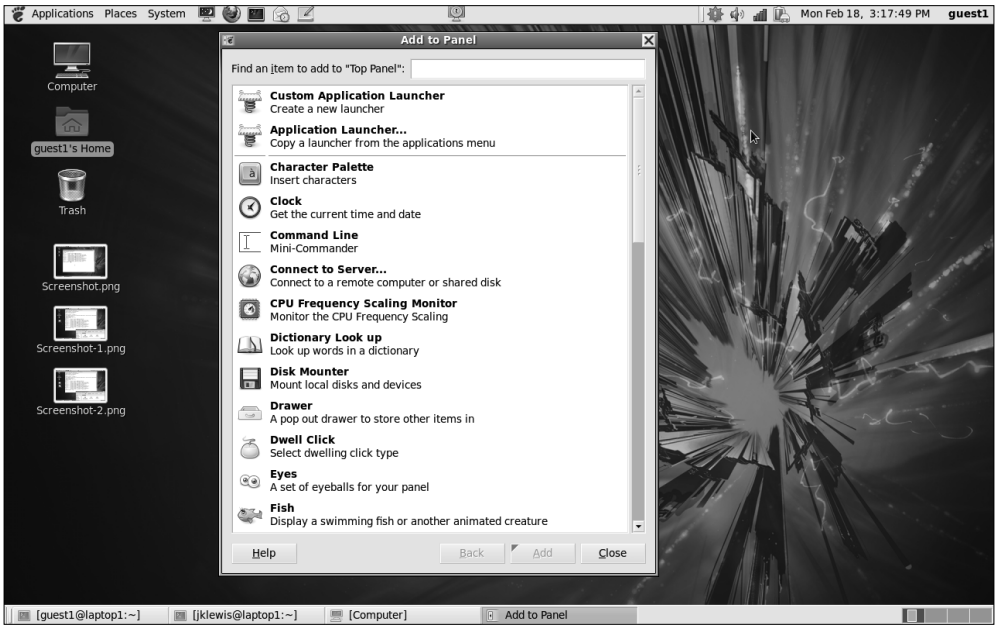


This shows the desktop, a command window, and the **Computer** folder. The top and bottom "rows" are the panels. From the top, starting on the left, are the **Applications**, **Places**, and **System** menus. I then have a screensaver, the Firefox browser, a terminal, **Evolution**, and a Notepad. In the middle is the lock-screen app, and on the far right is a notification about updates, the volume control, Wi-Fi strength, battery level, the date/time, and the current user. Note that I have customized several of these, for example, the clock.

Getting ready

If you have a computer running the GNOME 2 desktop, you may follow along in this section. A good way to do this is by running a Live Image, available from many different Linux distributions.

The screenshot showing the **Add to Panel** window is as follows:



How to do it...

Let's work with this desktop a bit:

1. Bring this dialog up by right-clicking on an empty location on the task bar.
2. Let's add something cool. Scroll down until you see **Weather Report**, click on it and then click on the **Add** button at the bottom.
3. On the panel you should now see something like **0 °F**. Right-click on it.
4. This will bring up a dialog, select **Preferences**.
5. You are now on the **General** tab. Feel free to change anything here you want, then select the **Location** tab, and put in your information.
6. When done, close the dialog. On my system the correct information was displayed instantly.
7. Now let's add something else that is even more cool. Open the **Add to Panel** dialog again and this time add **Workspace Switcher**.
8. The default number of workspaces is two, I would suggest adding two more. When done, close the dialog.
9. You will now see four little boxes on the bottom right of your screen. Clicking on one takes you to that workspace. This is a very handy feature of GNOME 2.

There's more...

I find GNOME 2 very intuitive and easy to use. It is powerful and can be customized extensively. It does have a few drawbacks, however. It tends to be somewhat "heavy" and may not perform well on less powerful machines. It also does not always report errors properly. For example, using Firefox open a local file that does not exist on your system (that is, `file:///tmp/LinuxBook.doc`). A **File Not Found** dialog should appear. Now try opening another local file that does exist, but which you do not have permissions for. It does not report an error, and in fact doesn't seem to do anything. Remember this if it happens to you.

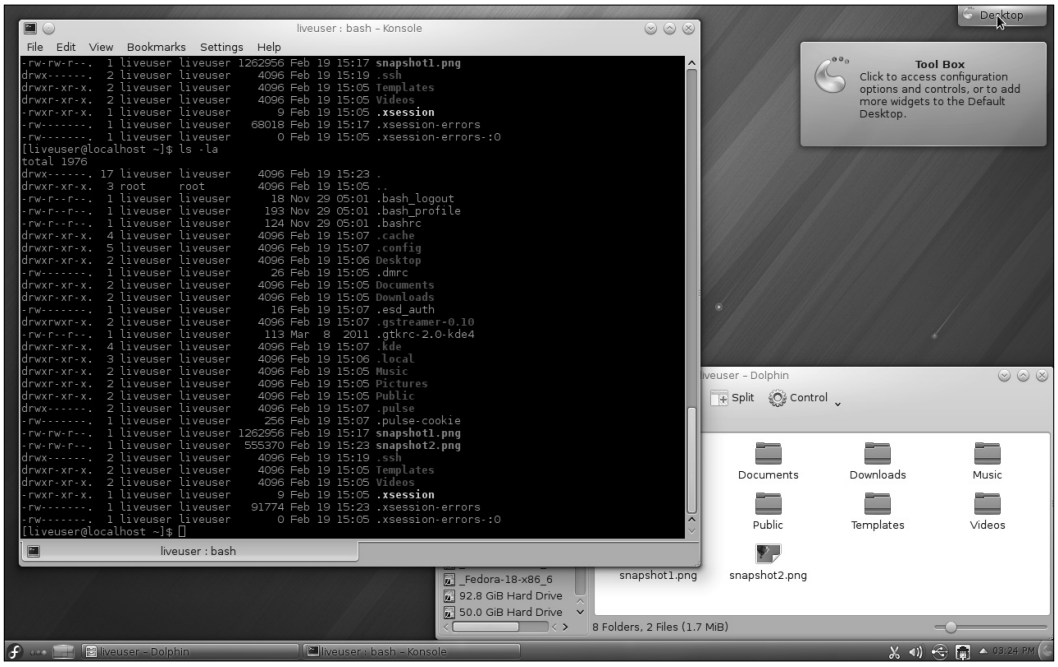
KDE desktop

The **KDE** desktop was designed for desktop PCs and powerful laptops. It allows for extensive customization and is available on many different platforms. The following is a description of some of its features.

Getting ready

If you have a Linux machine running the KDE desktop you can follow along. These screenshots are from KDE running on a Live Media image of Fedora 18.

The desktop icon on the far right allows the user to access **Tool Box**:

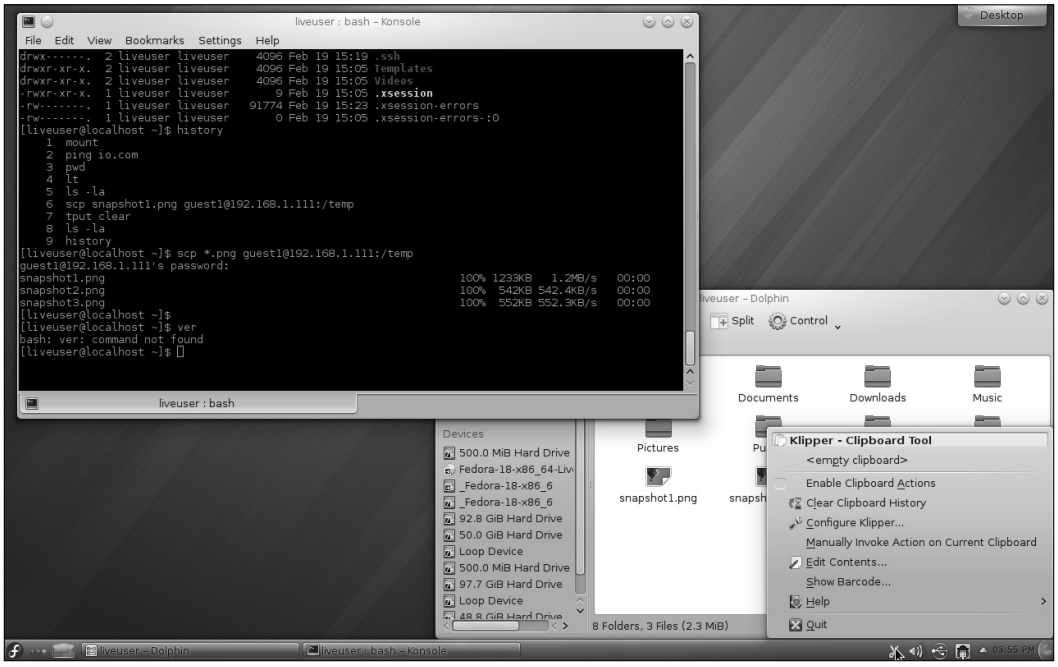


You can add panels, widgets, activities, shortcuts, lock the screen, and add a lot more using this dialog.

The default panel on the bottom begins with a Fedora icon. This icon is called a **Kickoff Application Launcher** and allows the user to access certain items quickly. These include **Favorites**, **Applications**, a **Computer** folder, a **Recently Used** folder, and a **Leave** button.

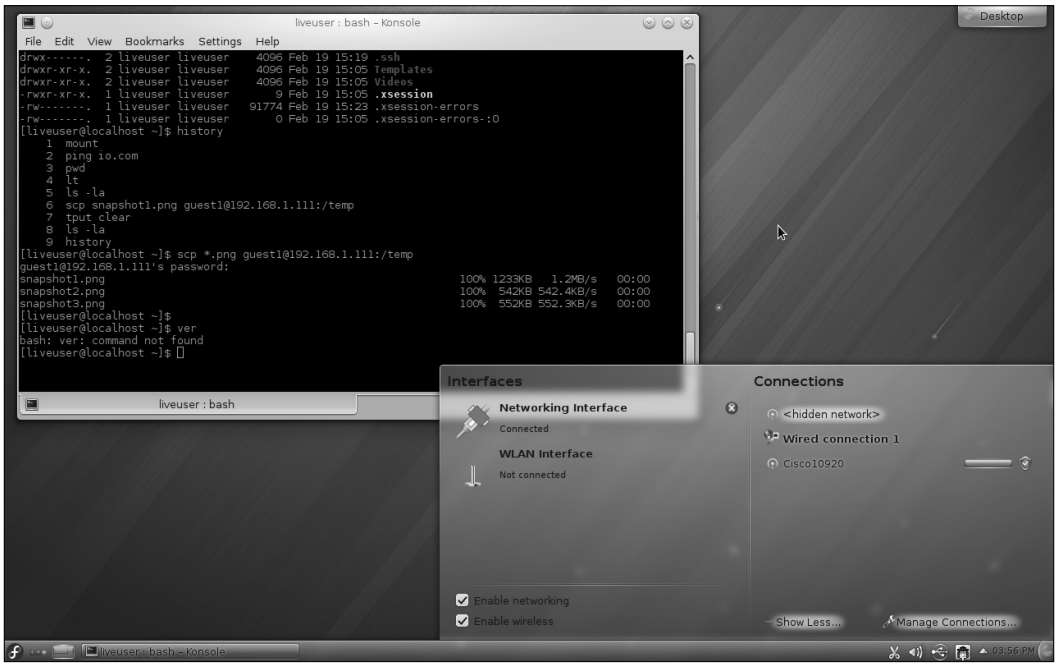
If you click on the next icon it will bring up the **Activity Manager**. Here you can create the activities and monitor them. The next icon allows you to select which desktop is currently in the foreground, and the next items are the windows that are currently open. Over to the far right is the **Clipboard**.

Here is a screenshot of the clipboard menu:



Next is the volume control, device notifier, and networking status.

Here is a screenshot of **Interfaces** and **Connections** dialog:



Lastly, there is a button to show the hidden icons and the time.

How to do it...

Let's add a few things to this desktop:

1. We should add a console. Right-click on an empty space on the desktop. A dialog will come up with several options; select **Konsole**. You should now have a terminal.
2. Close that dialog by clicking on some empty space.
3. Now let's add some more desktops. Right-click on the third icon on the bottom left of the screen. A dialog will appear, click on **Add Virtual Desktop**. I personally like four of these.
4. Now let's add something to the panel. Right-click on some empty space on the panel and hover the mouse over **Panel Options**; click on **AddWidgets**.
5. You will be presented with a few widgets. Note that the list can be scrolled to see a whole lot more. For example, scroll over to **Web Browser** and double-click on it.
6. The web browser icon will appear on the panel near the time.

There's more...

You can obviously do quite a bit of customization using the KDE desktop. I would suggest trying out all of the various options, to see which ones you like the best.

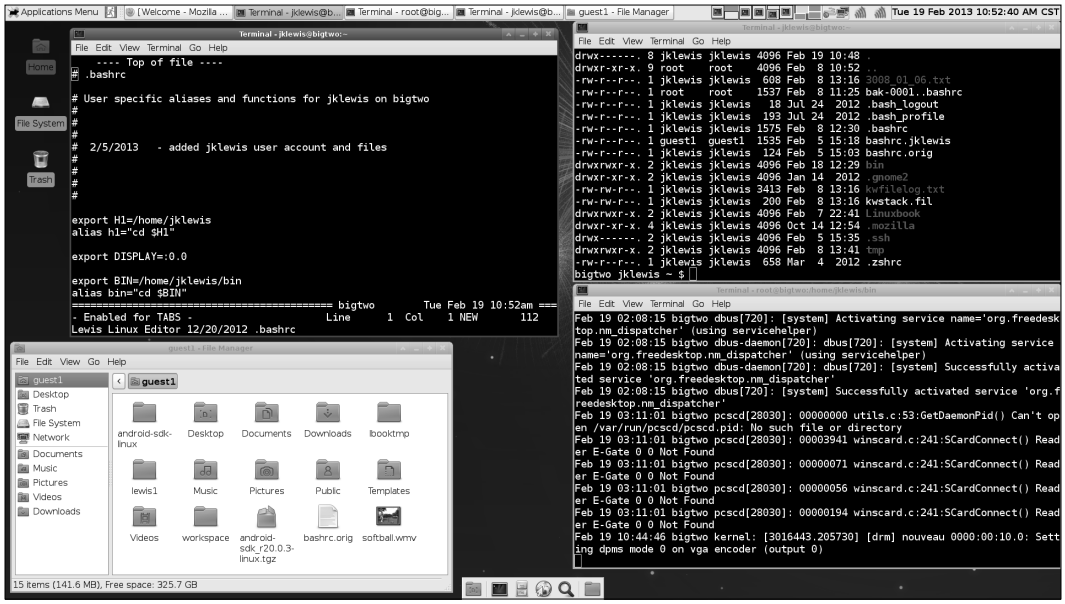
KDE is actually a large community of open source developers, of which KDE Plasma desktop is a part. This desktop is probably the heaviest of the ones reviewed, but also one of the most powerful. I believe this is a good choice for people who need a very elaborate desktop environment.

xfce

xfce is another desktop environment for Linux and UNIX systems. It tends to run very crisply and not use as many system resources. It is very intuitive and user-friendly.

Getting ready

The following is a screenshot of xfce running on the Linux machine I am using to write this book:



If you have a machine running the xfce desktop, you can perform these actions. I recommend a Live Media image from the Fedora web page.

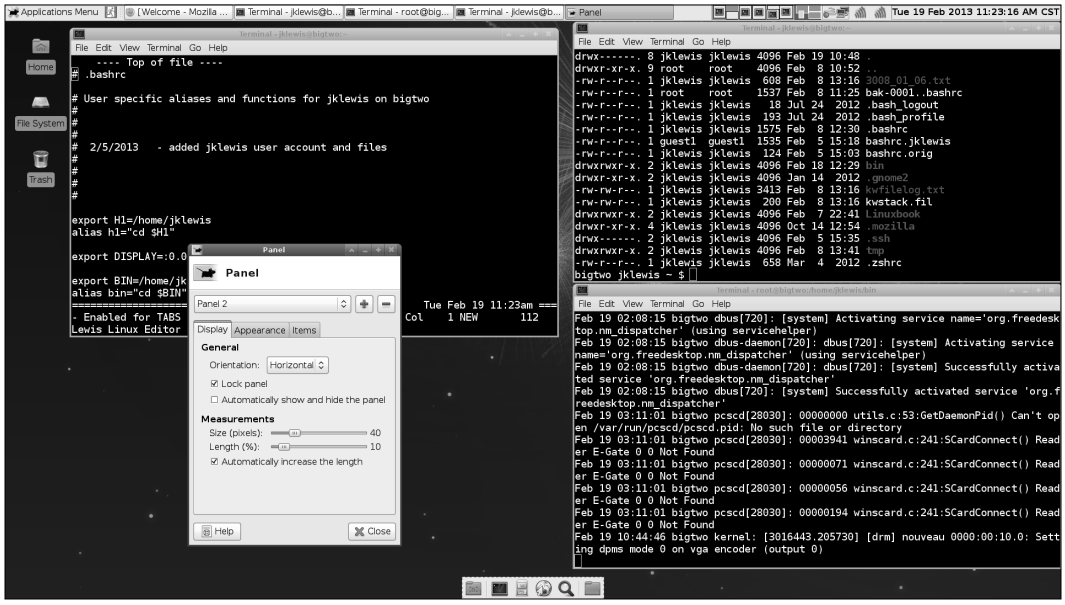
While somewhat similar to GNOME 2, the layout is somewhat different. Starting with the panel on the top (**panel 1**) is the **Applications Menu**, following by a **LogOut** dialog. The currently open windows are next. Clicking on one of these will either bring it up or minimize it depending on its current state. The next item is the **Workspaces** of which I have four, then the Internet status. To complete the list is the volume and mixer apps and the date and time. The screen contents are mostly self-explanatory; I have three terminal windows open and the **File Manager** folder.

The smaller panel on the bottom of the screen is called **panel 2**.

How to do it...

Let's work with the panels a bit:

1. In order to change panel 2 we must unlock it first. Right-click on the top panel, and go to **Panel** | **PanelPreferences**.
2. Use the arrows to change to panel 2. See the screenshot below:



3. You can see it is locked. Click on **Lock panel** to unlock it and then close this dialog.
4. Now go to panel 2 (on the bottom) and right-click on one of the sides. Click on **AddNewItems...**.
5. Add the applications you desire.

There's more...

This is by no means an exhaustive list of what xfce can do. The features are modular and can be added as needed. See `http://www.xfce.org` for more information.

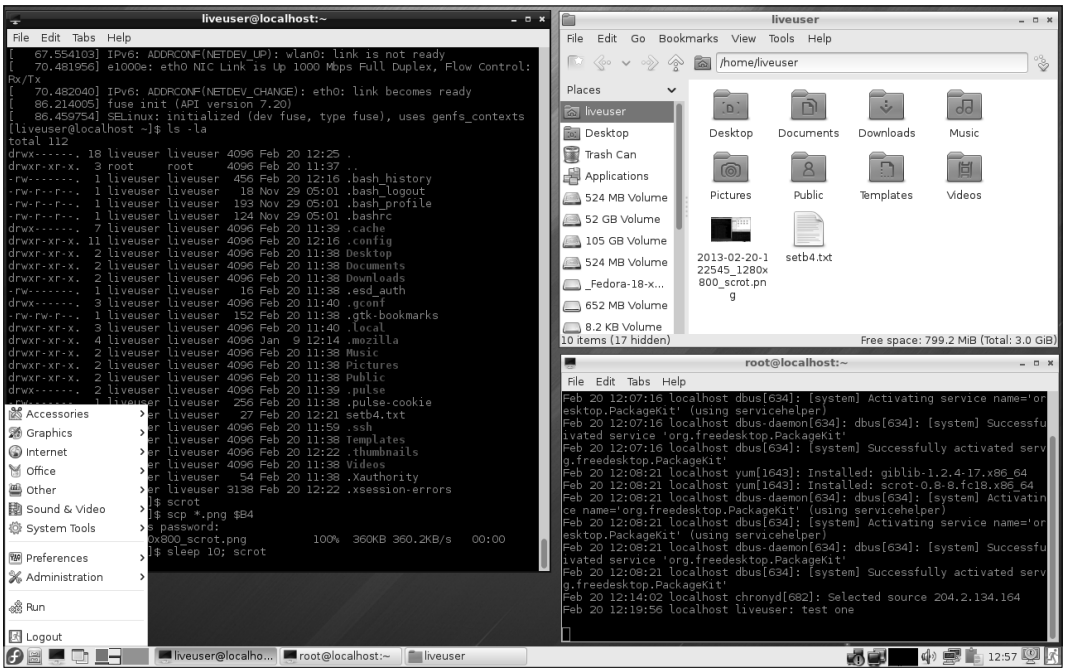
LXDE

LXDE (**Lightweight X11 Desktop Environment**) was designed to work well in low resource conditions and is a relatively new environment. Unlike most of the other desktops, the components of LXDE do not have many dependencies and can run independently.

Getting ready

If you have a machine using this desktop you can follow along with this section.

This is a screenshot of LXDE running on a Live Media image of Fedora 18:



As you can see, there are two terminals open and the file manager. Starting on the left of the panel is the familiar-looking Fedora icon, which has just been clicked on. It brings up the pulldown as shown. The next icon is the file manager and then an LXTerminal.

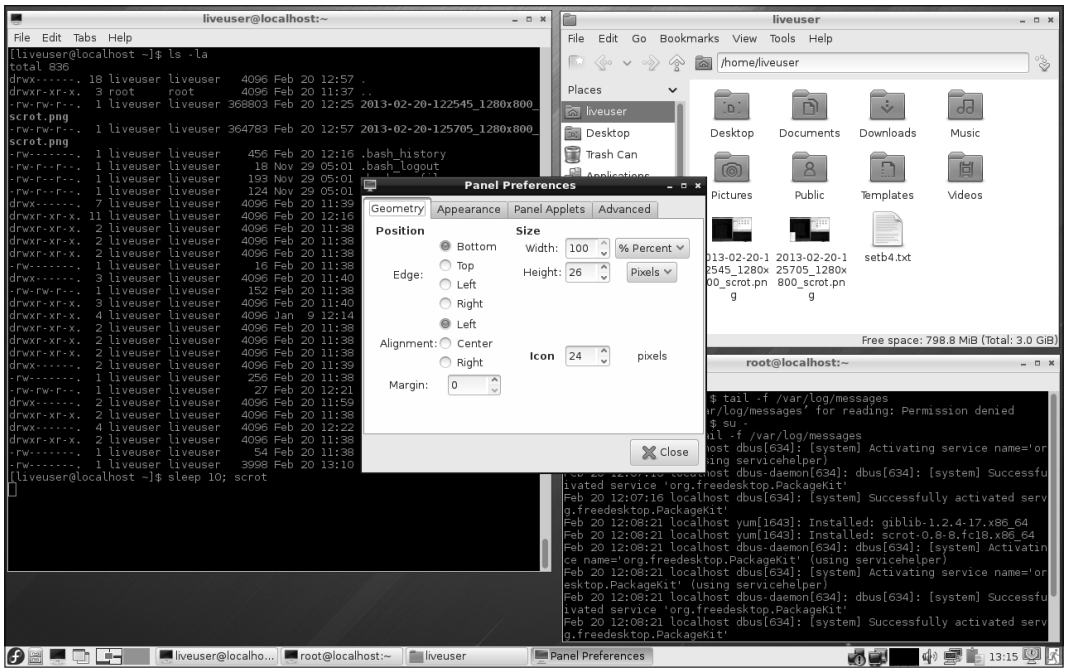
The next icon says "Left click to iconify all windows. Middle click to shade them". I chose to leave this icon as is.

The next are two desktop icons, and then the event list. Farther to the right is a Wi-Fi icon (Wi-Fi not activated), a wired Ethernet status, a system monitor, volume control, and the Network Manager Applet. After that is the clipboard manager, time, a lock-screen icon, and a logout box.

How to do it...

Let's work with this desktop a bit:

1. Right-click on an empty spot of the panel, a pulldown will be displayed.
2. Click on **Panel Settings**. The following screen will pop up:



3. Let's change the font size. Click on **Appearance**, and then **Size** under **Font**.
4. Using the scroll keys change the value to something else. The change will appear instantly. When it looks good, select **Close**.
5. Let's add an app. Bring up the panel settings again and click on **Add / Remove Panel Items**.
6. Click on **Add**, scroll down and click on **Desktop Number / Workspace Name**. The name of the workspace you are currently in shows up on the far right of the panel. I personally like this feature a lot.

There's more...

I found LXDE to be very intuitive and fast. I believe it would work well, particularly on laptops and mobile devices, where power is at a premium.

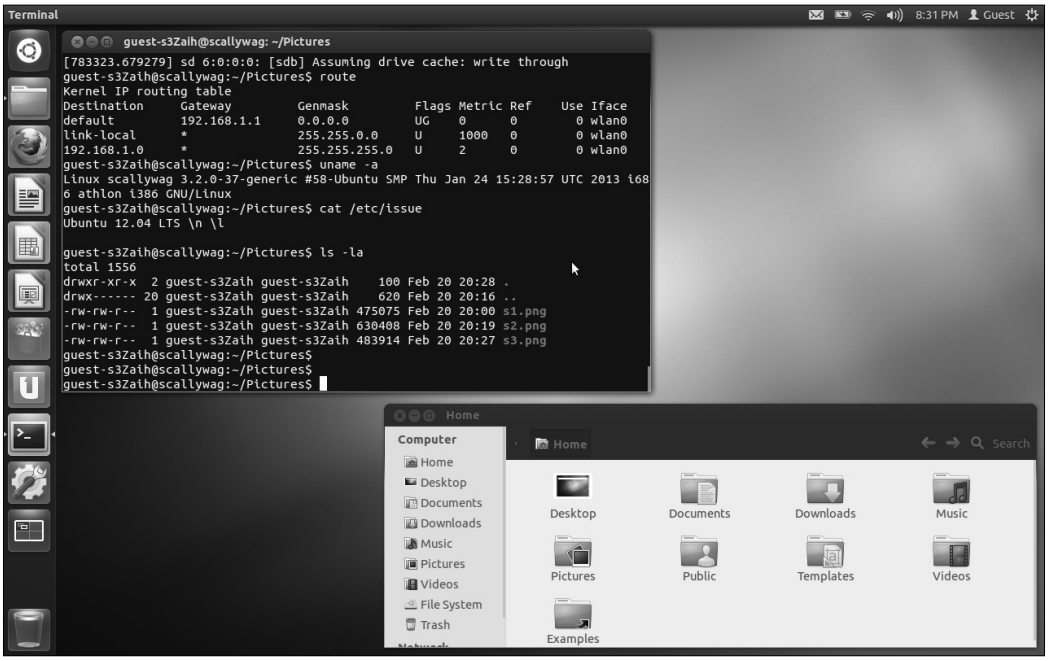
Unity

Unity is a shell interface for the GNOME environment used primarily on Ubuntu systems. It was designed to work well on small screens, for example, it employs a vertical application switcher. Unlike the other desktops/managers, it is not itself a collection of executables but uses existing applications.

Getting ready

If you have a machine running the Unity desktop, you can follow along with this section.

The following is a screenshot of Unity running on Ubuntu 12.04:



On the desktop is a GNOME terminal session and the **Home** folder. Starting with the vertical panel on the left is the Dash Home icon. It allows the user to find things quickly. Under that is the **Home** folder (already opened) and then the Firefox web browser. The next three are Libre Office Writer, Calculator, and Impress. Next is the Ubuntu Software Center, which is used to search for and purchase applications. The next icons are for Ubuntu One, a Terminal, System Settings, the Workspace Switcher, and the Trash folder.

To complete the discussion of the top panel, on the far right is the icon for Evolution. The next is the Battery status icon, network status (both wired and wireless), and the volume control. The remaining icons are the time, a switch user accounts icon, and the log out button.

Interesting enough, the terminal was not available by default on this guest desktop.

How to do it...

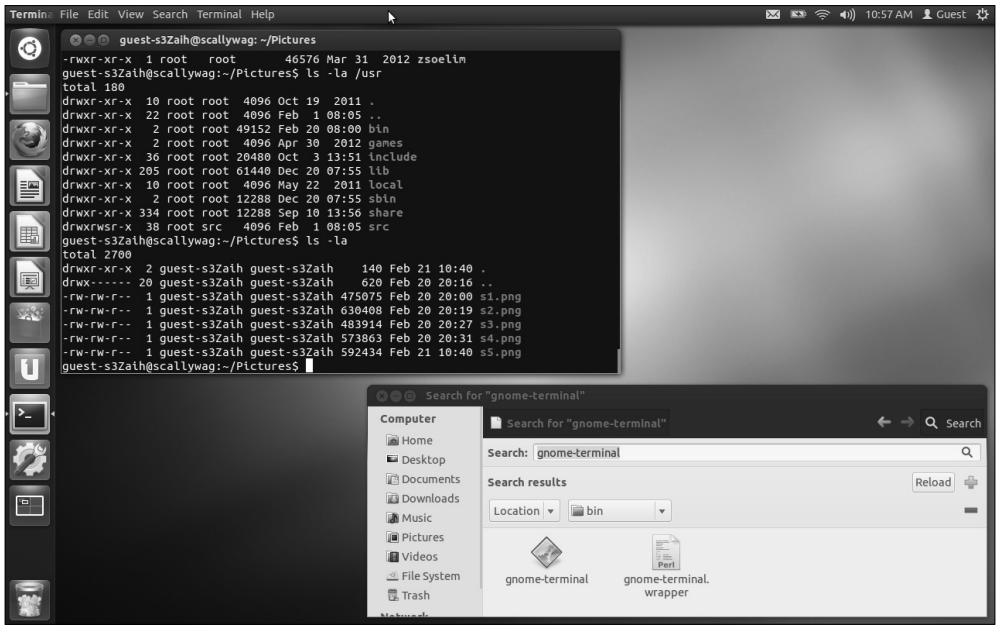
Let's add a terminal to this desktop:

1. Open the **Home** folder and then click on **File System**.
2. Double-click on the **usr** folder and then the **bin** folder.
3. Click on **Search** to open that dialog box.
4. Type in `gnome-terminal` and press *Enter*.
5. Double-click on the **gnome-terminal** icon.
6. It will open up on the screen, and you also see it as an icon along the left side panel.
7. Right-click on this icon and select **Lock to Launcher**. You now have a terminal icon.

The top panel on Unity works a little differently from the other desktops. Try the given steps:

1. Open the **Home** folder.
2. Open a terminal if you haven't already done so.
3. Now, click somewhere on the **Home** folder. The text **Home Folder** will be shown on the panel.

4. Now click on the **Terminal**. The text **Terminal** now appears. The menu items listed on the panel always correspond to the window or app that has the focus.



There's more...

I found Unity to be very different from the other desktops. At first it was a bit difficult, but like everything else it gets better with time. I believe this desktop would be popular with users who do not have much experience with Linux/UNIX systems.

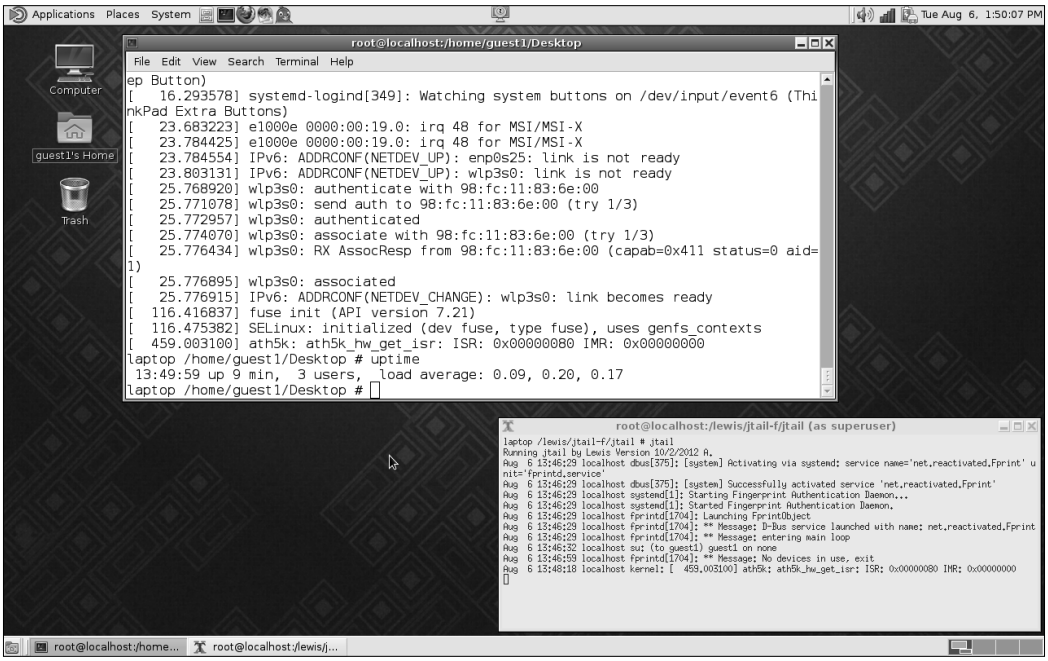
Mate

The **Mate** desktop was created to give users a more productive environment similar to GNOME 2. I am currently running Fedora 19 on my laptop using Mate and it is running fine. Note that I downloaded the F19 installation DVD and chose Mate during the install process.

Getting started

You can use a Live Image or a full install DVD from the Fedora site to follow along with these steps, whichever you prefer.

The following is a screenshot of Mate on Fedora 19:



You can see I already have two terminals open. On the top left is the **Applications** pulldown, which allows you to browse and run installed applications. The next one is **Places**, which allows you to access documents, folders, and network places. Next is **System**, where you can change the desktop appearance and behavior, get help, or log out. The icons are Caja, a file manager, and then a terminal. Yes, the Mate people were smart enough to include one by default. The next icons are Firefox, a mail app, and a messenger app. I added the lock-screen **icon**, which is in the middle. On the right is the volume, then the Wi-Fi bars, the battery status, and the date (which I customized a bit).

On the bottom left is an icon that allows you to hide all windows and show the desktop. And finally, on the far right are four workspaces.

How to do it...

Let's change a few things on this desktop:

1. First let's add the **Lock Screen** app. Right-click in the middle of the top panel.
2. Click on **Add to Panel...**.
3. Click on **Lock Screen** and follow the instructions. Close the dialog.
4. Now let's work with the time and date. Left-click on it and you will notice a calendar is displayed.
5. Left-click on the time and date again to close the calendar and then right-click on it. Click on the **Preferences** tab.
6. The **Clock Preferences** window should be displayed. Here you can change how the time and date are shown. I clicked on **Show seconds** because I like seeing the full time.
7. Close the dialog.

There's more...

As you can see, **Mate** works very much like **GNOME 2**. It is very intuitive and easy to use. The designers did a fine job creating this desktop.

Where to buy this book

You can buy Linux Utilities Cookbook from the Packt Publishing website:
`http://www.packtpub.com/linux-utilities-cookbook/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/linux-utilities-cookbook/book