From Technologies to Solutions

Designing and Implementing

Linux Firewalls and QoS

using netfilter, iproute2, NAT, and L7-filter

Learn how to secure your system and implement QoS using real-world scenarios for networks of all sizes

Lucian Gheorghe

PACKT PUBLISHING

Designing and Implementing Linux Firewalls and QoS using netfilter, iproute2, NAT, and L7-filter

Learn how to secure your system and implement QoS using real-world scenarios for networks of all sizes

Lucian Gheorghe

BIRMINGHAM - MUMBAI

Designing and Implementing Linux Firewalls and QoS using netfilter, iproute2, NAT, and L7-filter

First published: October 2006

Production Reference: 2181006

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 1-904811-65-5

www.packtpub.com

Cover Image by www.visionwt.com

Credits

Author
Lucian Gheorghe

Reviewer
Barrie Dempster

Development Editor
Louay Fatoohi

Assistant Development Editor
Nikhil Bangera

Technical Editor
Niranjan Jahagirdar

Code Testing
Ankur Shah

Editorial Manager
Dipali Chittar

Indexer
Mithil Kulkarni

Proofreader
Chris Smith

Layouts and Illustrations
Shantanu Zagade

Cover Designer
Shantanu Zagade

About the Author

Lucian Gheorghe has just joined the Global NOC of Interoute, Europe's largest voice and data network provider. Before Interoute, he was working as a senior network engineer for Globtel Internet, a significant Internet and Telephony Services Provider to the Romanian market. He has been working with Linux for more than 8 years putting a strong accent on security for protecting vital data from hackers and ensuring good quality services for internet customers. Moving to VoIP services he had to focus even more on security as sensitive billing data is most often stored on servers with public IP addresses. He has been studying QoS implementations on Linux to build different types of services for IP customers and also to deliver good quality for them and for VoIP over the public Internet. Lucian has also been programming with Perl, PHP, and Smarty for over 5 years mostly developing in-house management interfaces for IP and VoIP services.

I would like to thank everyone who is reading this book and the people that run netfilter, iproute2, and L7-filter projects. Your feedback is very important to me, so drop me a line at `lucian.firewallbook@gmail.com`. The book is far from being perfect so please send me errata information on the same email address (I would love to receive erratas from readers because it will convince me that people who read this book actually learned something :-))

I want to dedicate this book to my father, my mother, and my sister—I love you very very much. Many thanks go to the team at Globtel who were like second family to me, to my girlfriend for understanding me and standing by me, to Louay and the rest of the team at Packt Publishing for doing a great job, to Nigel Coulson, Petr Klobasa and the rest of the people at Interoute for supporting me, to Claudiu Filip who is one of the most intelligent people I know, and last, but not least, to the greatest technical author alive—Cristian Darie.

About the Reviewer

Barrie Dempster is currently employed as a Senior Security Consultant for NGS Software Ltd, a world-renowned security consultancy well known for its focus in enterprise-level application vulnerability research and database security. He has a background in Infrastructure and Information Security in a number of specialized environments such as financial services institutions, telecommunications companies, call centers, and other organizations across multiple continents. Barrie has experience in the integration of network infrastructure and telecommunications systems requiring high-caliber secure design, testing, and management. He has been involved in a variety of projects from the design and implementation of Internet banking systems to large-scale conferencing and telephony infrastructure, as well as penetration testing and other security assessments of business-critical infrastructure.

Table of Contents

Preface

A networking firewall is a logical barrier designed to prevent unauthorized or unwanted communications between sections of a computer network. Linux-based firewalls besides being highly customizable and versatile are also robust, inexpensive, and reliable.

The two things needed to build firewalls and QoS with Linux are two packages named netfilter and iproute. While netfilter is a packet-filtering framework included in the Linux kernels 2.4 and 2.6, iproute is a package containing a few utilities that allow Linux users to do advanced routing and traffic shaping.

L7-filter is a packet classifier for the Linux kernel that doesn't look up port numbers or Layer 4 protocols, but instead looks at the data in an IP packet and does a regular expression match on it to determine what kind of data it is, mainly what application protocol is being used. IP2P is an alternative to L7-filter, but has been designed for filtering only P2P applications while L7-filter takes into consideration a wider range of applications.

What This Book Covers

Chapter 1 is a brief introduction to networking concepts. It covers the OSI and TCP/IP networking models with explanations of their layers, TCP and UDP as Layer 4 protocols, and then rounds off the chapter with a discussion on IP addresses, Subnetting, and Supernetting.

Chapter 2 discusses possible security threats and vulnerabilities found at each of the OSI layers. The goal here is to understand where and how these threats can affect us and to stay protected from attackers. It then rounds off the discussion by sketching out the basic steps required to protect the services that run on our system.

Chapter 3 introduces two tools needed to build Linux firewalls and QoS. We first learn the workings of netfilter, which is a packet-filtering framework, and implement what we have learned to build a basic firewall for a Linux workstation. We then see how to perform advanced routing and traffic shaping using the IP and TC tools provided by the iproute2 package. The chapter ends with another example scenario where we implement the concepts learned in the chapter.

Chapter 4 discusses NAT, the types of NAT, how they work, and how they can be implemented with Linux by giving practical examples. It also describe packet mangling, when to use it, and why to use it.

Chapter 5 covers Layer 7 filtering in detail. We see how to install the L7-filter package, apply the necessary Linux kernel and iptables patches, and test our installation. We then learn the different applications of L7-filter and see how to put them to practical use. We also see how to install and use IPP2P, which is an alternative to the L7-filter package, but only for P2P traffic, and finally we set up a test between the two packages.

Chapter 6 raises two very popular scenarios, for which we design, implement, and test firewalls and a small QoS configuration. In the first scenario, we configure Linux as a SOHO router. Being a relatively smaller network with few devices, we learn how to adapt to what we have learned in the earlier chapters to suit this environment and build a secure network. We implement transparent proxies using squid and iptables so that children/minors cannot access malicious or pornographic web content. Our firewall setup implements NAT to redirect traffic from certain ports to other hosts using Linux. This configuration is tested by checking the NAT table and seeing how the kernel analyzes our rules.
As part of QoS, we split the bandwidth between the devices in a SOHO environment using HTB. Assuming a 1Mbps connection, we design a policy to split it between the 4 devices creating 4 HTB child classes for the 4 devices. In the end, we test our QoS configuration using the `tc class show` command.
In the second scenario, we configure Linux as router for a typical small to medium company.

Chapter 7 covers the design of a firewall system for a hypermarket having its headquarters in one location, one store in the same city, and several stores in other cities. The hypermarket has an application that uses MSSQL databases in each location, which are replicated at the headquarters. All locations have IP Analog Telephone Adapters with subscriptions at the main provider (the HQ provider). In this example we use, just like in the real H.323 as the VoIP protocol. We set up all remote locations to have an encrypted VPN connection using `ip tunnel` to connect to the headquarters. Users are shown how to create a QOS script with HTB that controls bandwidth usage based on priorities.

The next firewall taken up is that for a small ISP setup that has one internet connection, an access network, a server farm, and the internal departments. The setup of firewall scripts for each of them and methods to handle the tricky wireless server are covered. The QoS is handled by the intranet server, the wireless server, and the Core router.

Chapter 8 covers the design of a three-layered network deployed at a large provider of Internet and IP telephony services, the three layers being Core, Distribution, and Access. It explains network configuration first on the core and distribution levels and then moves on to building firewalls. The huge size of the network also means that there is a need to tackle newer security threats. We have four Cores running BGP under Zebra and each one is peculiar in its own way. There are three data services that this ISP can provide to its customers: Internet access, national network access, and metropolitan network access. This chapter will show you how to handle QoS so as to limit this traffic as needed.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

There are three styles for code. Code words in text are shown as follows: "To limit upload, we will mark packets in the `PREROUTING` chain of the `mangle` table".

A block of code will be set as follows:

```
#Drop SSH packets except from admins
$IPT -A INPUT -s ! 1.2.3.16/28 -p tcp --dport 22 -j DROP
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items will be made bold:

```
tc filter add dev eth0 protocol ip parent 1:0 prio 5 u32 match ip src
1.2.3.34 flowid 1:100
```

New terms and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "In the **IP: Netfilter Configuration** section you will find the options needed for NAT".

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to `feedback@packtpub.com`, making sure to mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on `www.packtpub.com` or email `suggest@packtpub.com`.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the Example Code for the Book

Visit `http://www.packtpub.com/support`, and select this book from the list of titles to download any example code or extra resources for this book. The files available for download will then be displayed.

The downloadable files contain instructions on how to use them.

Errata

Although we have taken every care to ensure the accuracy of our contents, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in text or code—we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent versions of this book. If you find any errata, report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **Submit Errata** link, and entering the details of your errata. Once your errata have been verified, your submission will be accepted and the errata added to the list of existing errata. The existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with some aspect of the book, and we will do our best to address it.

1
Networking Fundamentals

When it comes to theory, some of you out there might find it boring to read; so the first thing that may go through your mind is to skip this chapter. Don't do it. Even if you think that you know all the theoretical concepts, a recapitulation is good anytime.

Network professionals talk about protocols, devices, and software in terms of which OSI Layer they function at. When people talk about high-performance Layer 3 switches these days, they talk about switches that can perform OSI Layer 3 tasks and they expect you to know which tasks are at that layer. A simple deduction makes you realize that classic switches perform OSI Layer 2 functions.

Layer 3 switches are beyond the scope of this book, but that was a simple example of why you should know the OSI layered model, which is purely theoretical. Further in this book, you will learn about "Layer 7 filtering" which refers to how to filter what is on OSI Layer 7, which I'm sure you will find very attractive to read and implement.

By definition, a network is a group of two or more computer systems linked together, with the ability to communicate with each other.

The types of networks commonly used are:

- **LAN** (**Local Area Network**): A network in which the computers are close together (the same building).
- **WAN** (**Wide Area Network**): A network in which the computers are at very long distances.
- **MAN** (**Metropolitan Area Network**): A city-wide network.
- **CAN** (**Campus Area Network**): A network in a campus or a military base.
- **SAN** (**Storage Area Network**): A high-performance network used to move data between servers and dedicated storage devices.
- **VPN** (**Virtual Private Network**): A private network built over the public network infrastructure (over the Internet).

- **HAN** (**Home Area Network**): A network in a personal home. This term is rarely used; most people use the term LAN in this matter.

Computers in a user home network (a HAN) are usually connected to the building switch and form a LAN with the other users' computers. This switch is connected to a MAN or a CAN that is connected to the largest WAN, which is the Internet.

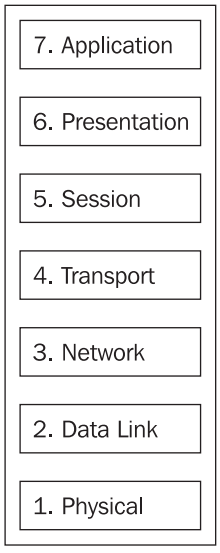
The OSI Model

In order for computers to communicate, they must speak the same language or protocol. In the early days of networking, networks were disorganized in many ways. Companies developed proprietary network technologies that had great difficulties in exchanging information with other or existing technologies; so network interconnections were very hard to build. To solve this problem, the International Organization for Standardization (ISO) created a network model that helps vendors to create networks compatible with each other.

In 1984, ISO released the Open Systems Interconnection (OSI) reference model, which is a well-defined set of specifications that ensures greater compatibility among various technologies.

In fact, OSI is a description of network communication that everyone refers to. It is not the only network model, but it has become the primary model for network communication. You will see further in this chapter, that the TCP/IP model is only a reduced version of the OSI model.

The OSI model consists of seven layers, each illustrating a particular network function.



Information contained in one layer usually has headers and trailers and data encapsulated from an upper layer. **Encapsulation** is the process of placing the data from an upper layer between headers and trailers so that when data is received by a layer, after it is analyzed, the protocol at that layer removes the headers and trailers and gives the data to the upper layer in the format that the upper layer understands.

At Layer 7 (application) of the OSI model we have the user interface (a web browser for example). Layer 6 (presentation) handles how data is presented (e.g. HTML). While accessing a web page, a computer may be sending/receiving emails. Keeping data from different applications separate is the job for Layer 5 (session) of the OSI model. At Layer 4 (transport) we find protocols that transfer the data (TCP for example), while at Layer 3 (network) we find logical addressing, which is used for path determination (e.g. IP). At Layer 2 (data link), we find network protocols such as Ethernet, and at the lowest layer, Layer 1 (physical), we find the cabling specifications (e.g. RJ-45).

This was a quick overview on the OSI layers. Now, let's have a closer look at these layers in order for us to understand the communication process.

OSI Layer 7: Application

The OSI application layer refers to communication services to applications. When programmers design an image editor for example, they don't have to think about adding OSI Layer 7 capabilities to that software, because it has no need for communication with other computers. On the other hand, when creating an FTP client, they must add communication capabilities to that software.

At Layer 7 we usually find Telnet, FTP, HTTP, SMTP, SNMP, or SSH.

When we say, for example, Layer 7 filtering, we refer to filtering application data, regardless of what port or computer it may come from.

OSI Layer 6: Presentation

The purpose of the presentation layer is defining the data formats in which data is represented. Data formats are usually standard formats like ASCII, JPEG, GIF, TIFF, MPEG, etc. OSI Layer 6 also defines encryption as a presentation layer service.

The importance of defining data formats is obvious. For example, when sending email, you usually send it plain text (ASCII) or HTML. If the receiving application doesn't know these data formats, your email will not be displayed correctly.

OSI Layer 6 provides a service to the upper OSI layer (application). It formats the data to be sent across the network in a manner that the receiving application is able to understand and/or manipulate.

OSI Layer 5: Session

The session layer defines how to start, control, and end conversations. These conversations are called sessions. OSI Layer 5 ensures inter-host communication, meaning that it establishes ways to manage sessions between applications.

An application may communicate with several other applications (on other PCs) at the same time. For each communication channel, Layer 5 starts a separate session that provides a service to the upper layer (presentation). The session layer ensures that a series of messages is completed. For example, if only half the data is received on a particular session, Layer 5 will not pass the data to the upper layer if the application is built this way. For example, suppose you go to an ATM machine, log in, print your account status, and insert an amount you want to extract from your account, but a communication error happens right then. The ATM will not give you the cash before it debits your account; instead, it will wait for the confirmation from the central system that the account was debited with that amount and then gives you the cash.

At the session layer, we find SQL, NFS, RPC, etc. Usually, the operating system is responsible for OSI Layer 5.

OSI Layer 4: Transport

The transport layer ensures the management of virtual circuits between hosts that can provide error correction. It contains a series of protocols concerned with transportation issues between hosts. These protocols may reorder the data stream if the packets arrive out of order. Layer 4 protocols are also responsible for multiplexing incoming data for different flows to applications running on the same host.

OSI Layer 4 provides a service to the session layer, meaning that after the data is received, multiplexed, and reordered, it is given to the upper layer (session) for handling.

The most common Layer 4 protocols are TCP, UDP, and SPX. The most important features of Layer 4 protocols are error correction and flow control. Because a router can discard packets for many reasons (communication errors, network congestion, etc.) Layer 4 protocols can provide retransmission of packets that the other host didn't receive. This is called **error correction**. Also, because of bandwidth limitations, if data is sent from one device using its full physical bandwidth, network congestion will occur. Layer 4 protocols are responsible for limiting transmission speed so that the network doesn't get flooded. This is called **flow control**.

We will see later in this chapter how error connection and flow control are accomplished and what protocols provide reliable or unreliable transport.

OSI Layer 3: Network

The network layer defines end-to-end delivery of data. In order for computers to be identified, the network layer defines logical addressing (e.g. IP addresses). OSI Layer 3 also defines how routing works and how routes are learned by routers for packet delivery. Also, the network layer defines fragmentation of packets, which is the process that breaks packets into smaller units in order to accommodate media with smaller maximum transmission unit (MTU) sizes.

Usually at OSI Layer 3 we find IP and IPX. When we think about OSI Layer 3, we must think of "routing". For example, routers are Layer 3 devices that run routing protocols for path determination.

Routers make their routing decisions based on the routing tables they have. Routing tables are collections of rules that define where data should go for a specific address or network.

At the beginning of this chapter, I was talking about one very common issue these days—"Layer 3 switches". Layer 3 switches switch packets according to a Layer 3 routing table. Usually, routers have a small number of interfaces that connect to switches for connectivity with other endpoints. In IP, Layer 3 switches are transparent routers with a very high density of ports.

OSI Layer 2: Data Link

The data link layer specifications are concerned with transferring data over a particular medium. For example, IEEE 802.3, which is the protocol for Ethernet, is found at OSI Layer 2. Hubs and switches are Layer 2 devices because they forward Ethernet packets over copper wires. At the data link layer we find protocols like ATM, Frame Relay, HDLC, PPP, FDDI, etc.

What we need to understand from this is that OSI Layer 2 specifies how packets are sent to the communication link. When we think about OSI Layer 2, we can think "switching", for example.

OSI Layer 1: Physical

The physical layer contains specifications for the physical medium of transmission that the data link layer protocols use. Layer 1 specifications are about connectors, pins, electrical currents, light modulation, etc. At Layer 1, we find the 802.3 standard, which has definitions about the Ethernet pinout, cable lengths, voltages, etc. More than that, we find cabling specification standards for RJ45, RJ48, V.35, V.24, EIA/TIA-232, and so on.

When we think about Layer 1, we can think "cables and connectors".