# 配套光盘内容介绍

本光盘为书中编程训练的配套光盘，包含第3章到第12章十次编程训练的相关代码。光盘中所有代码均使用C语言编写，在ubuntu 9.04平台上采用g++与gcc编译器进行调试。本书部分章节的代码需要在特定内核版本下运行。本书中对Linux内核的分析及扩展全部基于Linux内核2.6.24版本，尚未对其它版本进行测试。光盘中的所有程序仅适用于32位操作系统，尚未在64位系统上进行测试。

#### 第3章

DES加密聊天工具，共包含3个文件：chat.cpp、des.h、makefile。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。

运行：命令行模式下输入 ./chat，然后选择程序执行角色（客户端或服务器）。如选择服务器，则程序自动打开端口开始监听；如选择客户端，则需要输入服务器地址。最后输入命令quit退出程序。

#### 第4章

基于DES和RSA的加密聊天工具，共包含4个文件: chat.cpp、des.h、rsa.h、makefile。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。

运行：命令行模式下输入 ./chat，然后选择程序执行角色（客户端或服务器），如选择服务器，则程序自动打开端口开始监听；如选择客户端，则需要输入服务器地址。最后输入命令quit退出程序。

此外，代码中实现了异步，select以及普通三种IO模型，需要通过定义宏控制。

#### 第5章

基于MD5的文件完整性检测程序，共包含4个文件：md5.h、md5.cpp、test.cpp、nankai.txt。其中md5.h与md5.cpp文件实现了md5类。test.cpp文件包含了整个程序的main函数，根据不同的命令完成不同的功能，如显示摘要信息，生成MD5摘要等。nankai.txt是本书提供的用于验证文件完整性测试文件。读者可以自己编写程序来生成nankai.txt的md5摘要，然后与书中的结果进行比较，验证所编程序的正确性。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。

运行：在命令行下输入 ./md5 [-参数] 运行程序。根据不同的功能，参数分为五种类型，功能如下：

* h是帮助信息
* t是测试MD5应用程序
* c是计算指定文件的MD5值
* mv是手动验证指定文件的完整性
* fv是通过.md5文件验证指定文件的完整性

#### 第6章

基于RawSocket的数据包过滤程序，共包含7个文件：inetheader.h、rawsocsniffer.h、rawsocket.h、rawsocket.c、rawsocsniffer.c、main.c、makefile。

编译：命令行模式下，切换到代码所在目录，输入make即可编译。编译后将生成一个可执行文件main，以及一些obj文件。

运行：命令行下输入 ./main [-s 源IP] [-d 目的IP] [-atui] 即可运行程序，可以通过设置不同的参数可以实现简单的过滤功能。参数功能如下：

* s是根据源IP地址过滤数据包
* d是根据目的IP地址过滤数据包
* a是过滤ARP包
* t是过滤TCP包
* u是过滤UDP包
* i是过滤ICMP包

#### 第7章

基于OpenSSL的web服务器，共包含13个文件。

编译：编译本章程序需要使用OpenSSL库。OpenSSL是一款由加拿大的Eric Yang等人编写的开源软件包，实现了SSL及相关加密技术。OpenSSL的官方网站http://www.openssl.org/。读者可以从ftp://ftp.openssl.org/source/下载OpenSSL的源代码。本章编程使用Open SSL 0.9.8版本。

正确安装OpenSSL库后，在命令行模式下，切换到代码所在目录，输入make后回车即可完成编译。

运行：命令行下输入 ./MyWebServer 运行程序。然后打开浏览器，在地址栏输入 https://[服务器IP地址]:8000，即可登录服务器页面。

#### 第8章

综合端口扫描器，共包含12个文件。其中Scaner.h实现了ping函数以及一些共用函数的编写。Scaner.cpp文件包含了main函数，根据不同的命令调用不同的端口扫描模块。DataStructure.h定义了各扫描线程的参数以及IP数据报头的格式。TCPConnectScan.h与TCPConnectScan.cpp文件实现了TCP Connect扫描。TCPFINScan.h与TCPFINScan.cpp文件实现了TCP FIN扫描。TCPSYNScan.h与TCPSYNScan.cpp文件实现了TCP SYN扫描。UDPScan.h与UDPScan.cpp文件实现了UDP扫描。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。

运行：在命令行下输入 ./Scaner [-参数] 运行程序。根据不同的功能，参数分为五种类型，功能如下：

* h是帮助信息
* c是TCP Connect扫描
* s是TCP SYN扫描
* f是TCP FIN扫描
* u是UDP扫描

#### 第9章

简单的网络诱骗系统分为2个部分：basic文件夹中为基本功能部分，extend文件夹中为扩展功能部分。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。编译后将生成内核模块文件HoneyPot.ko。

运行：命令行模式下，切换到代码所在目录，输入insmod HoneyPot.ko 将内核模块文件加载到内核中。此外HoneyPot.ko模块将日志存储于/tmp/logfile（UID）.txt文件中，其中（UID）对应产生该log进程的用户ID。测试结束后，在命令行模式下输入rmmod移出本模块。

#### 第10章

简单的入侵检测系统分为2个部分：数据预处理与Kmeans算法实现。在DataPretreat文件夹中包含了实现数据预处理任务的文件DataPretreat.h与DataPretreat.cpp。另外文件corrected和Test是数据预处理的输入文件，而corrected\_datatreat和Test\_datatreat则是处理后的输出文件。在文件夹Kmeans中包含了文件Kmeans.h、Kmeans.cpp、ClusterTree.h、ClusterTree.cpp、Commmom.h。其中Kmeans.h与Kmeans.cpp文件实现了K-Means算法，而ClusterTree.h与ClusterTree.cpp则实现了聚类树结构。Commmom.h文件包含了程序共用的数据结构。此外，Result.txt文件用于记录聚类分析的结果，而Log.txt文件则记录聚类过程中的相关信息。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。

运行：命令行模式下，切换到目录DataPretreat，输入 ./DataPretreat [处理文件名] 对指定文件进行数据预处理。切换到目录Kmeans，输入./Kmeans 完成入侵检测数据的聚类与测试工作。

#### 第11章

基于Netfilter的防火墙包含4个文件， filter\_ip.c、filter\_port.c、 filter\_prot.c、 makefile。其中filter\_ip.c中代码实现的功能就是基于源IP地址过滤，filter\_port.c实现了基于目的端口过滤，filter\_prot.c实现了基于协议过滤的功能。

编译：命令行模式下，切换到代码所在目录，输入make后回车即可。编译后将生成三个文件，filter\_ip.ko，filter\_port.ko以及 filter\_prot.ko。

运行：通过本章所介绍的加载模块的方式将这些模块加载到内核即可实现不同的过滤功能。比如在命令行下运行insmod filter\_ip.ko即可将filter\_ip.ko模块加载到内核中，从而实现基于IP地址过滤的功能。

#### 第12章

Linux内核协议栈加固程序只包括内核模块代码，不包括修改后的Linux内核代码，。

编译：命令行模式下，切换到代码所在目录，输入./execute后回车即可。

运行：采用与第9章和第11章相同的内核模块加载方法，使用insmod命令将AntiDos.ko模块加载到内核中。加载成功后，在另一台主机上尝试利用ping命令连接当前主机。如果加载的内核模块生效，将会得到失败的结果。

注意：此代码只有在修改后的Linux内核中才可以正常工作，具体方法详见本书第12章内容。