网络应用系统安全手册

中联绿盟 编著

清华大学出版社

网络应用系统安全手册

中联绿盟　编著

清华大学出版社

北　京

内 容 简 介

网络安全是一个整体的概念，网络中的风险包括信息出入口的安全脆弱性、操作系统的安全脆弱性、应用程序的安全脆弱性、传输协议的安全脆弱性、数据库的安全脆弱性、系统遭受病毒感染的安全威胁和硬件设备的安全脆弱性。

对于一些系统管理员而言，似乎有了防火墙，网络安全就有了保障。可是实际上，一个看似固若金汤，配备了防火墙、IDS 等一系列安全设备的网络系统却由于 Web 服务上一个小小的 CGI 漏洞，可能会导致攻击者入侵服务器甚至渗透整个网络。所以防火墙只是安全的起步，对人员的管理和对应用系统的加固才是实现网络安全之道。

本书详细介绍了各种常用网络服务配置及其安全设置，并列举了该服务程序历年来出现过的漏洞和解决方法，相信能给网络管理员带来很大的帮助。

本书适合于初级网络管理员使用，也可供相关专业的广大工程技术人员参考。

图书在版编目（CIP）数据

网络应用系统安全手册/中联绿盟编著.—北京：清华大学出版社，2003

ISBN 7-302-06673-6

Ⅰ. 网… Ⅱ. 中… Ⅲ. 计算机网络-安全技术-技术手册 Ⅳ. TP393.08-62

中国版本图书馆 CIP 数据核字（2003）第 041503 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969

组稿编辑：吴宏伟
文稿编辑：钟志芳
封面设计：秦　铭
版式设计：张红英
印 刷 者：北京彩艺印刷有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：28.25　**字数**：650 千字
版　　次：2003 年 9 月第 1 版　2003 年 9 月第 1 次印刷
书　　号：ISBN 7-302-06673-6/TP・4995
印　　数：1～5000
定　　价：38.00 元

前　言

随着计算机技术和通信技术的飞速发展，网络正逐步改变着人们的工作方式和生活方式，成为当今社会发展的一个主题。网络的开放性、互连性、共享性程度的扩大，特别是Internet的出现，使网络的重要性和对社会的影响也越来越大。随着网络上电子商务、电子现金、数字货币、网络国税等新兴业务的兴起，网络安全问题变得越来越重要。

计算机网络犯罪所造成的经济损失令人吃惊。仅在美国每年因计算机犯罪所造成的直接经济损失就达150亿美元；在全球平均每20秒就发生一次网上入侵事件，有近80%的公司至少每周在网络上要被大规模地入侵一次，并且一旦黑客找到系统的薄弱环节，所有用户都会遭殃。

根据CNNIC在2002年初进行的第九次中国互联网络发展状况统计调查数据表明，在过去一年内用户计算机被入侵的情况如下：

- 被入侵过：63.3%
- 没有被入侵过：29.9%
- 不知道：6.8%

面对计算机网络的种种安全威胁，必须采取有力的措施来保证安全。无论是在局域网还是在广域网中，网络的安全措施应是能全方位地针对各种不同的威胁和脆弱性，这样才能确保网络信息的保密性、完整性和可用性。

本书就是立足于此，希望通过对本书的阅读，使网络管理员平时安装配置应用服务时能够多考虑安全方面的问题，这样就可以大大减少系统或者网络遭受攻击的可能性。

本书声明

由于本书是介绍应用系统安全，一般都针对应用服务有许多的配置例子或者解决方案。这些例子和解决方案不可避免地需要使用各种信息，如网络地址、域名、组织名称、邮件地址、口令等，所有的举例都尽可能使用虚拟信息，如果不慎有雷同，那仅仅是巧合，并希望谅解。

特别感谢

本书主要由许治坤、汪列军、于旸完成编写，其中得到CIW朋友的大力帮助，整理了很多素材。另外季昕华帮助编写了MS SQL SERVER和Oracle安全配置的这两节内容。要感谢的朋友很多，这里不一一列举。

中联绿盟

2003年4月

目　录

第 1 章　免费应用系统的安装

1.1　开放源码应用系统介绍

世界上有数不胜数的程序员在孜孜不倦地开发开放源码、完全免费的应用软件和服务软件，为 Internet 的发展做出了不可磨灭的贡献。尤其在 Unix/Linux 平台下，涌现出无数优秀的软件，而基于 Windows 平台的相对就要少的多。虽然这些应用系统是免费的，也许界面要稍微简陋一些，但它们的功能却丝毫不逊色于商业软件，而且在网络上被广泛使用，如 Apache、Wu-FTPD、MySQL 和 BIND 等。

大部分开放源码的软件都是基于 GNU 的 GPL（General Public License）发布的。GPL 协议是自由软件基金会发起的，力图保证所有用户的共享和修改自由软件的自由，对所有用户都是自由的。但它有一些限制，要求发布的软件必须开放源码，任何人都可以对自由软件进行修改，但是修改过的软件必须开放源码发布。

大部分 BSD 下的软件都是基于 BSD 许可证发布的。BSD 许可更加自由，允许任何人修改发布基于 BSD 许可证的软件，而且再发布时可以不开放源码。商业软件公司自然非常喜欢 BSD 许可证，可以自由地复制代码而不用公开修改后自己产品的代码。

这些许可证对软件开发者来说非常重要，但对于软件的使用者来说都是很自由的，没有任何限制。免费的应用系统由于没有任何经济收入，所以不像商业软件一样提供上门服务或其他方式的技术支持。但是开发者和使用者都写了许多安装和使用的技术文档，而且在网上的 News、Mailing List 或 BBS 里还会有无数的热心者会替你解答问题。

1.2　开放源码软件安装

1.2.1　简介

Unix/Linux 平台下的应用系统软件大都提供源码包，以便编译和安装。GNU Autoconf、Automake 和 Libtool 是用来自动生成配置编译参数的软件包，大部分软件都使用它们来开发生成编译配置文件。对于这类软件的安装只需执行./configure、make、make install 三步就可以了。这种安装方式是开发者和使用者都喜欢的，开发者只需提供一个最新的源码软件包，使用者经过自动配置可得到最佳的编译效果，使软件运行的效率提高，而且使用者如有特殊需求和能力，可以自己修改源码。

1.2.2 解压源码包

一般的软件的发布都提供 xxx.tar.gz 形式的压缩包，以方便使用者下载。用户用 tar 和 gzip 命令来解开压缩包，然后进行配置、编译和安装。

一般大多数的 Unix/Linux 可以用 tar 直接解开这种压缩包。tar.gz 的解包命令如下：

```
$ tar vxzf xxx.tar.gz
```

tar.bz2 的解包命令如下：

```
$ tar vxjf xxx.tar.bz2
```

1.2.3 查看 Readme 或 Install 文件

基本上所有的源码软件包都带有 Readme、Install 等说明文件，它们一般都详细说明了该软件的安装方法，以及一些需要特殊注意的配置选项。所以，安装前一定要仔细看一下说明文件。

1.2.4 编译和安装

标准的源码软件发布包都是用 GNU Autoconf、Automake 和 Libtool 工具来处理各种移植性的问题，用这些工具完成系统配置信息的收集，制作 makefile 文件。一般只需执行：

```
# ./configure
# make
# make install
```

当然，一般都可以使用./configure 来配置各种参数，如安装的路径、模块的加载等；可以通过./configure --help 来得到所有的配置信息。

1.2.5 反安装

卸载源码安装的软件比较麻烦，它没有像 Windows 下那些程序提供很直观的反安装程序，只能到安装目录中手工删除。有些源码软件发布包也提供了 make uninstall 命令来执行反安装，具体还要查看该软件的说明文件。

http://asic-linux.com.mx/~izto/checkinstall 提供一个小工具 Checkinstall，在执行完./configure、make 以后运行 Checkinstall，可以把要安装的软件打成一个包（目前支持

slackware 的 package 包、RedHat 的 rpm 包和 Debian 的 deb 包），然后，利用 package、rpm 或 deb 软件包管理机制就可以很方便地反安装。

以下是一个简单的示例。在编译完 rsync 以后，在源码目录执行：

```
# /usr/local/sbin/checkinstall

checkinstall 1.5.3, Copyright 2001 Felipe Eduardo Sanchez Diaz Duran
        This software is released under the GNU GPL.


The package documentation directory ./doc-pak does not exist.
Should I create a default set of package docs?  [y]:

Preparing package documentation...OK

Installing with "make install"...

================== Installation results ======================

Copying documentation directory...
mkdir -p /usr/local/rsync/bin
/usr/bin/install -c  -m 755 rsync /usr/local/rsync/bin
mkdir -p /usr/local/rsync/man/man1
mkdir -p /usr/local/rsync/man/man5
/usr/bin/install -c -m 644 ./rsync.1 /usr/local/rsync/man/man1
/usr/bin/install -c -m 644 ./rsyncd.conf.5 /usr/local/rsync/man
/man5

================== Installation succesful ======================

Copying files to the temporary directory...OK

Striping ELF binaries and libraries...OK

Compressing man pages...OK

Building file list...OK

Please choose the packaging method you want to use.
Slackware [S], RPM [R] or Debian [D]?
```

Checkinstall 会根据安装信息提示选择生成包的类型，比如选择 RPM，按 R，则执行：

```
Please write a description for the package.
End your description with an empty line or EOF.
>> checkinstall packaging
>>

This package will be built according to these values:

1 - Summary: [ checkinstall packaging ]
2 - Name:    [ rsync ]
3 - Version: [ 2.5.5 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group:   [ Applications/System ]
7 - Architecture: [ i386 ]
8 - Source location: [ rsync-2.5.5 ]
9 - Alternate source location: [  ]
10 - Provides: [ rsync ]

Enter a number to change any of them or press ENTER to continue:

**************************************
**** RPM package creation selected ***
**************************************

Building RPM package...OK

Installing RPM package...OK

Erasing temporary files...OK

Deleting doc-pak directory...OK

Writing backup package...OK

Deleting temp dir...OK


******************************************************************

Done. The new package has been installed and saved to
```

```
/usr/src/redhat/RPMS/i386/rsync-2.5.5-1.i386.rpm

You can remove it from your system anytime using:

    rpm -e rsync-2.5.5-1

**************************************************************
```

在设定完一些包的信息后，在/usr/src/redhat/RPMS/i386/目录下就生成了编译的rsync-2.5.5-1.i386.rpm。这样就可以直接安装这个 RPM 包，然后卸载时就可以利用 RPM 管理机制进行卸载。

1.3　Linux 的 RPM 机制

1.3.1　RPM 简介

RPM 是 RedHat Package Manager 的缩写，是由 RedHat 公司开发的软件包安装和管理程序。RPM 是一个强大的工具，用来安装、升级、卸载、查询和校验系统上的软件包。

RPM 可直接以 binary 方式安装软件包，并且可帮助用户查询是否已经安装了有关的库文件。在用 RPM 删除程序时，它会询问是否删除有关的程序。如果使用 RPM 来升级软件，RPM 会保留原先的配置文件，这样就不用重新配置新的软件。RPM 保留一个数据库，这个数据库中包含了所有软件包的资料。通过这个数据库，用户可以进行软件包的查询。RPM 虽然是为 Linux 而设计的，但是它已经移植到 SunOS、Solaris、AIX 和 Irix 等其他 Unix 系统上了。RPM 遵循 GPL 版权协议，用户可以在符合 GPL 协议的条件下自由使用及传播 RPM。

1.3.2　RPM 基本使用参数介绍

1. 安装

RPM 的安装模式如下：

```
rpm  -i  [安装选项]    <软件包>
```

可以用于所有模式的选项如下：

```
-v              显示信息
-h              用“#”显示完成的进度
```

```
--keep-temps 保留临时文件,临时文件通常位于/tmp/rpm-*,这个选项要用于debug
--quiet      安静工作，只有当出现错误时才给出提示信息
--help       显示帮助
--version    显示当前使用的RPM版本
```

一个典型的 RPM 包格式如“foo-1.0-1.i386.rpm”，其中指明了包名（foo）、版本号（1.0）、发行号（1）和硬件平台（i386）。以下是安装一个软件包的典型命令：

```
# rpm -ivh foo-1.0-1.i386.rpm
foo                  ########################################
```

RPM 支持远程安装，这个 RPM 包可以是远程的文件，如从 FTP 安装一个软件包，则执行：

```
# rpm -ivh ftp://ftp.redhat.com/foo-1.0-1.i386.rpm
Retrieving ftp://ftp.redhat.com/foo-1.0-1.i386.rpm
foo                  ########################################
```

如果软件包已被安装，会显示以下信息：

```
# rpm -ivh foo-1.0-1.i386.rpm
foo                  package foo-1.0-1 is already installed
error: foo-1.0-1.i386.rpm cannot be installed
```

如果仍要安装该包，可以使用升级模式或在命令行中使用--replacepkgs 选项，这样 RPM 将忽略该错误信息：

```
# rpm -ivh --replacepkgs foo-1.0-1.i386.rpm
foo                  ########################################
```

如果要安装的软件包中有一个文件已在安装其他包时被安装，会显示以下信息：

```
# rpm -ivh foo-1.0-1.i386.rpm
foo          /usr/bin/foo conflicts with file from bar-1.0-1
error: foo-1.0-1.i386.rpm cannot be installed
```

要想让 RPM 忽略该错误信息，请使用--replacefiles 命令行选项：

```
# rpm -ivh --replacefiles foo-1.0-1.i386.rpm
```

```
foo                    ########################################
```

--replacepkgs 选项和--replacefiles 选项也可以用--force 选项来代替。

未解决依赖关系的 RPM 包可能会“依赖”其他软件包，即要求在安装了特定的软件包之后才能安装该软件包。如果在安装这个软件包时未解决这种存在的依赖关系，会看到：

```
# rpm -ivh bar-1.0-1.i386.rpm
failed dependencies:
        foo is needed by bar-1.0-1
```

只有先安装完所依赖的软件包，才能解决这个问题。 如果想强制安装（这不是个好办法，因为安装后的软件包未必能正常运行），可以使用--nodeps 命令行选项。也可以用--test 选项进行测试，不实际安装，只是为了检查并显示可能存在的冲突。

2. 升级

升级软件包和安装软件包十分类似，以下是一个典型的 RPM 包升级命令：

```
# rpm -Uvh foo-2.0-1.i386.rpm
foo                    ########################################
```

RPM 将自动卸载已安装的老版本的 foo 软件包。用户可以总是使用“-U”来安装软件包，因为即便以往未安装过该软件包，也能正常运行。

RPM 执行智能化的软件包升级，自动处理配置文件，会显示如下信息:

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

在使用旧版本的 RPM 包来升级新版本的软件时，会产生以下信息:

```
# rpm -Uvh foo-1.0-1.i386.rpm
foo  package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
```

要使用 RPM 强行“升级”，请使用--oldpackage 选项命令：

```
# rpm -Uvh --oldpackage foo-1.0-1.i386.rpm
foo                    ########################################
```

--force 选项也能达到同样的效果。

RPM 更新模式是检查命令行中指明的包版本与安装在系统中的包版本是否一致。 当 RPM 更新选项处理完已安装包的新版本时，该包会升级到新版本。 但是，更新模式无法