

《Ubuntu Server 最佳方案》

篇	章	内容介绍
第 1 篇 拥抱 Ubuntu Server	第 1 章 敲开 Linux Server 的大门 第 2 章 拥抱 Ubuntu Server	介绍了 Linux 及其选型，并提供了 Ubuntu 快速入门指南
第 2 篇 LAMP 服务器	第 3 章 用 Apache 做 Web 服务器 第 4 章 LAMP 服务器之 PHP 篇 第 5 章 LAMP 服务器之 Perl 篇 第 6 章 LAMP 服务器之 Python 篇 第 7 章 Apache Tomcat 架设 第 8 章 最佳代理、反向代理服务器：Squid	介绍了最佳 LAMP 服务器，并对 PHP、Perl、Python、Tomcat 分别进行了详细的介绍。此外，由于代理服务器也属于 Web 范畴，因此本篇中还介绍了最佳代理服务器、反向代理服务器方案（Squid）

续

篇	章	内容介绍
第 3 篇 Mail 服务器	第 9 章 最佳邮件服务器方案 第 10 章 最佳邮件列表：Mailman	介绍了最佳邮件服务器方案（Postfix），该方案不仅支持虚拟用户、虚拟域，还支持 POP3 收信、IMAP 收信、SMTP 认证、TLS 加密、邮件别名和转发、磁盘限额、垃圾邮件过滤，支持病毒防护、Web 邮件界面，基本上涵盖了邮件服务器的方方面面。本篇还介绍了最佳邮件列表方案（Mailman），让您可以创建自己的邮件列表服务器
第 4 篇 文件服务器	第 11 章 最佳 FTP 服务器方案 第 12 章 最佳 NFS 服务器方案 第 13 章 与 Windows 共舞：Samba	介绍了最佳 FTP 服务器方案（PureFTPd）和最佳 NFS 方案，并介绍了与 Windows 环境沟通的方法（Samba）
第 5 篇 虚拟化	第 14 章 最佳虚拟化方案：OpenVZ	介绍了最佳虚拟化方案（OpenVZ），使您可以在一台物理服务器上虚拟几台、十几台甚至上百台 Linux 服务器
第 6 篇 DNS 和 DHCP 服务器	第 15 章 最佳 DNS 服务器：Bind9 第 16 章 DNS 轮询 第 17 章 最佳 DHCP 服务器方案	介绍了最佳 DNS 服务器（Bind9）和最佳 DHCP 方案。如果您管理一个内部网络，这两种服务器不可或缺

第 7 篇 负载均衡和 集群	第 18 章 负载均衡、高可用的 Web 集群 第 19 章 负载均衡、高可用的 MySQL 集群	介绍了负载均衡、高可用的最佳 Web 集群方案和最佳 MySQL 数据库集群方案。作为一个 Linux 管理员，您需要在架构设计之初，就能预见未来几年内的需求增长，否则在业务迅速增长、需要添加服务器时，您将手足无措
第 8 篇 远程控制与 监控	第 20 章 最佳远程控制方案： SSH 第 21 章 最佳服务器监控方案： Nagios	介绍了最佳远程控制方案（SSH）和最佳监控方案（Nagios）。远程控制最重要的是安全，否则黑客很有可能闯进来，使您失去控制权。监控可以让您及时了解服务器状况，免得总是“救火”
第 9 篇 数据备份与 系统安全	第 22 章 最佳 RAID 方案： RAID10 第 23 章 最佳数据安全方案： RAID10+LVM 第 24 章 Ubuntu Server 系统安全	介绍了最佳 RAID 方案（RAID10）、最佳数据安全方案（RAID10+LVM），并介绍了如何使 Ubuntu Server 变得更安全。安全是重中之重，您不仅要学会如何设置防火墙，还要熟知入侵检测和肉鸡检测的方法，以及如何处理被入侵的服务器

目 录

第 1 篇 拥抱 UbuntuServer

第 1 章 敲开 Linux Server 的大门 2

1.1 Linux 到底是什么 2

1.2 为何选 Linux，不选

1.2 Windows 3

1.2.1 Linux 可以定制 4

1.2.2 Linux 比 Windows 稳定 4

1.2.3 Linux 比 Windows 响应快 4

1.2.4 Linux 比 Windows 安全 5

1.2.5 Linux 不用花钱买 6

1.2.6 Linux 更适合远程管理 6

1.3 学习 Linux 的终南捷径 6

1.3.1 兴趣、试验 6

1.3.2 真正的捷径——LFS 7

1.4 选择哪个 Linux 发行版呢 7

1.4.1 先排除 Gentoo 8

1.4.2 再排除 Red Hat、CentOS 和 SUSE 8

1.4.3 Debian 不错 8

1.4.4 推荐使用 Ubuntu 8

1.5 应该买什么样的服务器呢 10

1.5.1 架构设计 10

1.5.2 服务器的选型 10

1.5.3 机房的选择 11

第 2 章 拥抱 Ubuntu Server	12
2.1 Ubuntu 的前世今生	12
2.2 安装 Ubuntu Server	13
2.2.1 安装前的准备	13
2.2.2 安装 Ubuntu Server	13
2.2.3 把语言环境变量改为英文	29
2.2.4 安全补丁、版本升级	30
2.3 Ubuntu 快速入门指南	31
2.3.1 nano 编辑器	31
2.3.2 强大的“资源管理器”：mc	32
2.3.3 快速查找文件	32
2.3.4 软件包管理	33
2.3.5 使用 apt 工具	35
2.3.6 给 Red Hat 用户	42
2.3.7 Ubuntu 网络配置	45
2.3.8 远程管理 Ubuntu Server	48
2.3.9 系统更新：apt-get update && apt-get upgrade	49
第 2 篇 LAMP 服务器	
第 3 章 用 Apache 做 Web 服务器	51
3.1 Apache 简介	51
3.2 Apache 的安装、配置	52
3.2.1 Apache 的安装	52
3.2.2 Apache 的配置	52
3.2.3 Apache 虚拟主机	54
3.3 Apache 性能优化	62
3.3.1 正确选择 MPM	63
3.3.2 优化 Apache 配置	63
3.3.3 使用反向代理	68
3.4 Apache 压力测试 (ab)	68
3.5 Apache 安全	70
3.5.1 安全更新	70
3.5.2 隐藏敏感信息	70
3.5.3 不要以 root 身份运行 Apache	72
3.5.4 密码认证	73
3.5.5 检查文件权限	75
3.5.6 关闭不用的模块	75
3.5.7 DDoS 攻击防范	76
3.6 Apache 日志分析	78
3.6.1 用 Webalizer 分析 Apache 日志	78
3.6.2 用 AWStats 分析 Apache 日志	80
3.6.3 Apache 日志合并	82
第 4 章 LAMP 服务器之 PHP 篇	84
4.1 MySQL 简介	84
4.2 PHP 简介	85

4.3	安装 LAMP 相关软件包	85
4.3.1	LAMP 软件包安装	85
4.3.2	LAMP 软件包删除	86
4.4	配置 Apache、MySQL、PHP	86
4.4.1	MySQL 配置	86
4.4.2	PHP 配置	88
4.5	Apache、MySQL、PHP 之间的关联	89
4.6	用 phpMyAdmin 管理 MySQL 数据库	90
4.6.1	phpMyAdmin 的安装	90
4.6.2	phpMyAdmin 排错	90
4.7	实例：用 Drupal 快速架设 Blog 网站	90
4.7.1	Drupal 是什么	90
4.7.2	获取 Drupal	91
4.7.3	为 Drupal 创建 MySQL 数据库和用户	91
4.7.4	为 Drupal 配置 PHP	92
4.7.5	为 Drupal 配置 Apache	92
4.7.6	安装 Drupal	92
4.7.7	为 Drupal 安装 Blog 模块	93
4.7.8	Drupal 的中文界面	94
第 5 章	LAMP 服务器之 Perl 篇	95
5.1	安装 Perl 模块	95
5.2	配置 cgi-bin 目录	95
5.3	Perl 程序测试	96
5.4	用 Perl 访问 MySQL 数据库	96
5.5	CGI 排错	98
5.6	实例：用 Twiki 假设 Wiki	99
5.6.1	安装 Twiki	99
5.6.2	配置 Twiki	99
第 6 章	LAMP 服务器之 Python 篇	101
6.1	安装 mod_python	101
6.2	配置 Apache	101
6.2.1	Publisher Handler	101
6.2.2	PSP Handler	102
6.3	让 Python 支持 MySQL	103
6.3.1	Python 连接 MySQL 数据库测试	103
6.3.2	Python 的 CGI 程序	104
6.3.3	CGI 排错	105
6.4	实例：用 Django 开发 Web 应用程序	105
6.4.1	安装 Django	105
6.4.2	创建自己的 Django 项目	105
6.4.3	运行 Django 开发服务器	105
6.4.4	连接 MySQL 数据库	106
6.4.5	后续开发步骤	107
6.5	实例：用 MoinMoin 实现 Wiki	107

6.5.1	安装 MoinMoin	108
6.5.2	创建 MoinMoin 实例	108
6.5.3	MoinMoin 权限控制	110
6.6	Python Web 应用的性能优化	111
6.6.1	mod-wsgi 介绍	111
6.6.2	mod-wsgi 支持的程序	112
6.6.3	mod-wsgi 的安装	112
6.6.4	测试	113
第 7 章	Apache Tomcat 架设	114
7.1	安装 Tomcat	114
7.2	配置 Tomcat	115
7.3	Tomcat 和 Apache 的整合: mod_jk	115
7.3.1	mod_jk 的安装	116
7.3.2	mod_jk 的配置	116
7.4	Tomcat 安全	117
7.4.1	保护 shutdown 端口	117
7.4.2	修改默认错误页面	118
7.4.3	删除样例文件	118
7.4.4	Manager WebApp 安全	118
第 8 章	最佳代理、反向代理服务器: Squid	119
8.1	Squid 安装	119
8.2	为 Squid 配置主机名	119
8.3	访问控制列表	120
8.4	正向代理	121
8.4.1	设置端口号	121
8.4.2	禁止某些 IP 地址上网	121
8.4.3	禁止在某时间段上网	122
8.4.4	个别网站的控制	122
8.4.5	用 NCSA 做密码认证	123
8.4.6	透明代理的设置	123
8.5	反向代理	126
8.5.1	Squid 反向代理单个后台 Web 服务器	127
8.5.2	Squid 反向代理多个后台 Web 服务器	127
8.6	Squid 排错	128
8.6.1	Squid 运行状态检查	128
8.6.2	Squid 日志文件	128
8.7	使用 SquidGuard	128
8.7.1	SquidGuard 能做什么	129
8.7.2	安装 SquidGuard	131
8.7.3	SquidGuard 基本配置	131
8.7.4	SquidGuard 高级配置	135
第 3 篇	Mail 服务器	
第 9 章	最佳邮件服务器方案	141
9.1	安装所有相关软件	142

9.1.1	安装服务器软件	142
9.1.2	安装内容过滤软件	143
9.1.3	安装其他软件	143
9.2	为 Postfix 准备数据库	144
9.2.1	创建数据库 maildb	144
9.2.2	为数据库 maildb 创建数据表	144
9.2.3	为数据库 maildb 创建视图	147
9.3	配置 Postfix	149
9.3.1	Postfix 与 MySQL 的 关联配置	149
9.3.2	让 Postfix 使用 Dovecot 分发邮件	155
9.4	配置 Dovecot	156
9.4.1	配置 dovecot.conf	156
9.4.2	配置 dovecot-sql.conf	158
9.4.3	修改配置文件权限	158
9.4.4	重新启动 Dovecot	158
9.5	用 Telnet 进行 SMTP/POP3/IMAP 测试	158
9.5.1	SMTP 测试	159
9.5.2	测试 POP3	161
9.5.3	测试 IMAP	162
9.6	用 Thunderbird 进行 SMTP/POP3/IMAP 测试	164
9.6.1	在 Thunderbird 中创建账号	164
9.6.2	修改 hosts 文件	165
9.6.3	在 Thunderbird 中用 POP 收取邮件	166
9.6.4	在 Thunderbird 中用 SMTP 发送邮件	167
9.6.5	在 Thunderbird 中用 IMAP 收取邮件	167
9.7	实现 SMTP 认证	168
9.7.1	配置 Postfix	169
9.7.2	用 Telnet 测试 SMTP 认证	169
9.7.3	用 Thunderbird 测试 SMTP 认证	170
9.8	强迫用户使用 TLS 加密连接 SMTP	171
9.9	使用自己创建的安全证书	172
9.10	利用 Dovecot 实现 Quota（磁盘限额）	173
9.10.1	启用 quota 插件	173
9.10.2	配置 quota	174
9.11	垃圾邮件、病毒过滤	176
9.11.1	配置 SpamAssassin	176
9.11.2	配置 AMaViSd	176
9.11.3	配置 Postfix，将邮件交给 AMaViSd 过滤	180
9.11.4	垃圾邮件测试	182
9.11.5	非法附件测试	183
9.11.6	将 Spam 自动转存到“垃圾”文件夹	183
9.12	Webmail 的实现	186

9.12.1	配置 SquirrelMail	186
9.12.2	访问 Webmail	187
9.13	修改系统别名/etc/aliases	188
9.14	Web 管理工具	189
9.14.1	安装 Virtual Mail Manager	189
9.14.2	使用 Virtual Mail Manager	190
第 10 章	最佳邮件列表: Mailman	191
10.1	安装 Mailman	191
10.2	配置 Mailman	192
10.2.1	修改主机名	192
10.2.2	配置 Apache	192
10.2.3	配置 Postfix	193
10.2.4	创建默认邮件列表	194
10.3	管理 Mailman	195
10.3.1	通过 Web 管理 Mailman	196
10.3.2	通过命令行管理 Mailman	197
10.4	普通用户的 Web 界面	199
第 4 篇	文件服务器	
第 11 章	最佳 FTP 服务器方案	201
11.1	要实现的功能	201
11.2	FTP 服务器的选择	202
11.2.1	淘汰标准一: 安全	202
11.2.2	淘汰标准二: 易用性	203
11.3	Pure-FTPd 的安装、配置	203
11.3.1	安装 Pure-FTPd	203
11.3.2	配置 Pure-FTPd	203
11.4	实现 FTP 用户的 Web 管理	206
11.4.1	安装 User manager for PureFTPd	207
11.4.2	配置 User manager for PureFTPd	207
11.4.3	设置 User manager for PureFTPd 管理员	207
11.4.4	Web 管理界面	208
11.5	Pure-FTPd 配置选项介绍	209
11.5.1	逻辑型配置选项	209
11.5.2	数值型配置选项	210
11.5.3	字符串型配置选项	211
11.5.4	IP 地址型配置选项	212
11.5.5	文件型配置选项	212
11.6	实现 TLS 认证	212
11.6.1	证书设置	212
11.6.2	服务器的 TLS 设置	213
11.6.3	FTP 客户端的 TLS 设置	213
11.7	FXP 协议支持	214
11.8	允许匿名访问	214

11.8.1	Pure-FTPd 设置	214
11.8.2	添加系统用户	215
第 12 章	最佳 NFS 服务器方案	216
12.1	安装前须知	217
12.1.1	用户权限	217
12.1.2	组权限	217
12.2	NFS 服务器的安装及配置	217
12.2.1	/etc/hosts 配置	218
12.2.2	安装 NFS 服务器软件	218
12.2.3	Portmap 安全	218
12.2.4	NIS 服务器配置	218
12.2.5	用/etc/exports 配置共享目录	220
12.3	NFS 客户端的安装及配置	220
12.3.1	/etc/hosts 配置	220
12.3.2	安装 NFS 客户端	221
12.3.3	配置 NFS 客户端	221
第 13 章	与 Windows 共舞: Samba	223
13.1	Samba 的好处	223
13.1.1	高性能	223
13.1.2	省钱	224
13.2	安装 Samba 并测试	224
13.2.1	安装 Samba	224
13.2.2	在 Windows 客户端上测试	224
13.3	Samba 配置	225
13.3.1	最简单的 Samba 配置	225
13.3.2	Samba 的安全认证	227
13.3.3	共享权限控制	229
13.3.4	文件写入实验	229
13.4	基本的家目录共享方案	231
13.4.1	创建私人目录	232
13.4.2	创建新用户	232
13.4.3	配置 Samba	233
13.5	其他共享方案	236
13.5.1	共享光驱	236
13.5.2	小组共享	237
第 5 篇	虚拟化	
第 14 章	最佳虚拟化方案: OpenVZ	240
14.1	OpenVZ 简介	240
14.1.1	可扩展性	240
14.1.2	密度	240
14.1.3	管理方便	241
14.2	安装 OpenVZ	241
14.2.1	安装前的准备	241
14.2.2	安装 OpenVZ	242

14.2.3	配置 OpenVZ	242
14.3	虚拟机的基本操作	244
14.3.1	虚拟机的创建	244
14.3.2	虚拟机的启停	245
14.4	vzctl 用法详解	246
14.4.1	vzctl 基本用法	246
14.4.2	创建虚拟机	246
14.4.3	虚拟机的启停等操作	247
14.4.4	设置虚拟机参数	247
14.4.5	其他命令和参数	253
14.5	/etc/vz/vz.conf 详解	253
14.5.1	全局参数	253
14.5.2	磁盘限额参数	253
14.5.3	网卡参数	254
14.5.4	虚拟机默认值	254
14.6	VE 的备份与恢复	254
14.6.1	安装 vzdump	255
14.6.2	vzdump 的用法	255
14.6.3	备份 VE	256
14.6.4	恢复 VE	256
14.7	OpenVZ 排错	256
第 6 篇 DNS 和 DHCP 服务器		
第 15 章 最佳 DNS 服务器: Bind9 259		
15.1	安装 Bind9	259
15.2	Bind9 的几种角色	260
15.3	配置 Bind9	260
15.3.1	Bind9 配置文件介绍	260
15.3.2	DNS 记录类型	260
15.3.3	DNS 缓存服务器的配置	261
15.3.4	主 DNS 服务器的配置	262
15.3.5	从 DNS 服务器的配置	266
15.4	让 Bind9 运行在 Chroot 环境	268
15.4.1	创建 Chroot 环境	268
15.4.2	Bind9 配置	269
15.4.3	日志路径设置	269
15.4.4	测试	269
15.5	Bind9 排错	269
15.5.1	DNS 测试	269
15.5.2	日志文件	271
第 16 章 DNS 轮询 273		
16.1	为什么要用 DNS 轮询	273
16.2	DNS 轮询是怎么工作的	273
16.3	DNS 轮询的实现方法	273
16.3.1	多个 CNAMEs 的方法 (Bind4、Bind8)	273

16.3.2 多个 A 记录的方法 (Bind9)	274
16.4 DNS 轮询的测试	274
16.5 DNS 轮询的缺陷	275
第 17 章 最佳 DHCP 服务器方案	276
17.1 DHCP 的好处	276
17.2 DHCP 提供信息的方法	277
17.3 安装 DHCP 服务器软件	277
17.4 配置 DHCP 服务器	278
17.4.1 网络环境介绍	278
17.4.2 DHCP 配置	278
17.4.3 测试	279
17.5 DHCP 排错	280
第 7 篇 负载均衡和集群	
第 18 章 负载均衡、高可用的 Web 集群	282
18.1 介绍	282
18.1.1 HAProxy 介绍	282
18.1.2 Keepalived 介绍	282
18.1.3 HAProxy+Keepalived 的好处	283
18.2 架构	283
18.2.1 架构详情	283
18.2.2 架构图	284
18.3 架构的实现	284
18.3.1 Web 服务器的安装及配置	284
18.3.2 HAProxy 的安装及配置	285
18.3.3 Keepalived 的安装及配置	287
18.4 测试	289
18.4.1 Web 节点故障模拟	289
18.4.2 负载均衡节点故障模拟	289
18.5 HAProxy 的 Web 统计页面	290
第 19 章 负载均衡、高可用的 MySQL 集群	291
19.1 MySQL 集群架构介绍	291
19.1.1 架构图	291
19.1.2 本例中的服务器	292
19.2 管理节点 (MGM) 的安装及配置	292
19.2.1 安装 MySQL	293
19.2.2 配置 ndb_mgmd.cnf	293
19.3 存储节点 (NDB) 的安装及配置	294
19.3.1 安装 MySQL	294
19.3.2 配置 my.cnf	294
19.4 阶段测试	295
19.4.1 集群连接状态测试	295
19.4.2 测试	296
19.5 实现负载均衡	300
19.5.1 ldirectord+heartbeat 介绍	300

19.5.2	让内核支持 IPVS	301
19.5.3	安装 heartbeat、ldirectord 等软件	302
19.5.4	配置 heartbeat	302
19.5.5	配置 ldirectord	303
19.5.6	NDB 节点配置	304
19.5.7	测试	305
19.6	注意事项	307
19.6.1	数据库引擎问题	307
19.6.2	内存问题	308
19.6.3	安全问题	308
第 8 篇 远程控制与监控		
第 20 章 最佳远程控制方案: SSH 310		
20.1	关于公钥认证	310
20.1.1	为什么要用公钥认证	310
20.1.2	公钥认证是怎么工作的	311
20.2	SSH 的安装	311
20.2.1	安装 SSH 服务器和客户端	311
20.2.2	测试	311
20.3	SSH 配置	312
20.3.1	生成密钥对	312
20.3.2	将公钥复制到服务器	312
20.3.3	SSH 登录测试	312
20.3.4	SSH 服务器配置	314
20.4	SSH 小技巧	315
20.4.1	用 scp 远程复制文件	315
20.4.2	在客户端上指定命令	316
20.4.3	在服务器上限制所执行的命令	316
20.4.4	修改密钥口令	317
20.4.5	将密钥放入内存	317
第 21 章 最佳服务器监控方案:		
第 21 章 Nagios 318		
21.1	Nagios 介绍	318
21.2	安装 Nagios	319
21.3	配置 Nagios	319
21.3.1	Nagios 初始化设置	319
21.3.2	Nagios 监控设置	320
21.4	手机短信提醒	327
21.5	Nagios 排错	328
第 9 篇 数据备份与系统安全		
第 22 章 最佳 RAID 方案: RAID10 330		
22.1	RAID 方案的选择	330
22.2	RAID10 的实现	332
22.2.1	手动分区	333
22.2.2	第一块硬盘分区	334

22.2.3	分区复制	338
22.2.4	创建 RAID 阵列	339
22.2.5	在 RAID 上创建分区	340
22.2.6	保存分区	342
22.3	RAID10 的日常维护	343
22.3.1	mdadm 的主要工作模式	343
22.3.2	mdadm 的选项	343
22.3.3	创建 RAID 阵列	345
22.3.4	查询 RAID 阵列	345
22.3.5	RAID 的监控	346
22.3.6	RAID 的启动/停止	346
22.4	故障处理	347
22.4.1	从 RAID 中移除设备	347
22.4.2	添加已有 RAID 物理卷	348
22.4.3	更换全新硬盘	348
22.5	添加备用硬盘	350
22.5.1	插入新硬盘	351
22.5.2	新硬盘分区	351
22.5.3	将新分区加入 RAID	351
22.5.4	设置 grub	352
22.5.5	故障模拟	352
22.6	RAID10 的空间扩展	352
第 23 章 最佳数据安全方案:		
第 23 章	RAID10+LVM	354
23.1	创建 RAID 物理卷	354
23.1.1	将第一块硬盘分区	354
23.1.2	剩余硬盘的分区处理	355
23.2	创建 RAID 阵列	355
23.2.1	创建 RAID1 阵列	355
23.2.2	创建 RAID10 阵列	356
23.3	LVM 的创建和配置	356
23.3.1	创建 LVM 物理卷	356
23.3.2	LVM 配置	356
23.4	创建/boot 分区	358
23.5	LVM 的相关命令	359
23.5.1	LVM 物理卷相关命令	359
23.5.2	LVM 卷组相关命令	360
23.5.3	LVM 逻辑卷相关命令	363
23.6	添加新硬盘	365
23.6.1	插入新硬盘	365
23.6.2	配置 RAID	365
23.6.3	在 RAID 上配置 LVM	367
23.6.4	扩容文件系统	368
23.7	更换硬盘	368

23.8	LVM 分区备份	368
23.8.1	创建快照	369
23.8.2	备份快照内容	369
23.8.3	删除快照	370
第 24 章	Ubuntu Server 系统安全	371
24.1	系统安全更新	371
24.1.1	订阅安全列表	371
24.1.2	自动更新	371
24.2	控制台安全	372
24.3	用户、密码管理	372
24.3.1	关于 root 用户	372
24.3.2	关于 sudo	373
24.3.3	关于/etc/sudoers	373
24.3.4	密码策略	375
24.4	ufw 防火墙	376
24.4.1	启用、禁用 ufw	376
24.4.2	基本规则设置	377
24.4.3	常用规则设置	378
24.4.4	高级规则设置	380
24.4.5	IP 伪装	382
24.5	入侵检测	384
24.5.1	安装 LAMP	384
24.5.2	安装、配置 Snort	384
24.5.3	安装、配置 BASE	387
24.6	肉鸡检测	392
24.6.1	chkrootkit 的使用	393
24.6.2	rkhunter 的使用	394
24.6.3	unhide 的使用	396
24.7	数据完整性检测	397
24.7.1	安装 Tripwire	398
24.7.2	配置 Tripwire	400
24.7.3	初始化 Tripwire 数据库	403
24.7.4	执行完整性检测	403
24.7.5	检测报告分析	403
24.7.6	查看 Tripwire 数据库内容	405
24.7.7	使用 Tripwire 的注意事项	406
24.8	被入侵后的系统恢复	406
24.8.1	保持冷静	407
24.8.2	断开网络	407
24.8.3	找到黑客入侵的方法	407
24.8.4	黑客文件清理	412
24.8.5	恢复未受影响的服务	412
24.8.6	修复问题	412
24.8.7	恢复受影响的服务	412

第 2 章



拥抱 Ubuntu Server





2.2 安装 Ubuntu Server

2.2.1 安装前的准备

```
/dev/sdb2 /home ext3 defaults 0 2
```

```
$ sudo blkid
```

```
/dev/sda1: UUID="ac369a10-e335-42b1-a3a5-ce9524c8130b" TYPE="ext3"
```

```
/dev/sda5: TYPE="swap" UUID="8f7943ed-6589-41db-90d0-f57a8ad7cdbd"
```

2.2.3 把语言环境变量改为英文

```
$ locale
```

```
LANG=zh_CN.UTF-8
```

```
LANGUAGE=zh_CN:zh
```

```
LC_CTYPE="zh_CN.UTF-8"
```

```
LC_NUMERIC="zh_CN.UTF-8"
```

```
LC_TIME="zh_CN.UTF-8"
```

```
LC_COLLATE="zh_CN.UTF-8"
```

```
LC_MONETARY="zh_CN.UTF-8"
```

```
LC_MESSAGES="zh_CN.UTF-8"
```

```
LC_PAPER="zh_CN.UTF-8"
```

```
LC="zh_CN.UTF-8"
```

```
LC_ADDRESS="zh_CN.UTF-8"
```

```
LC_TELEPHONE="zh_CN.UTF-8"
```

```
LC_MEASUREMENT="zh_CN.UTF-8"
```

```
LC_IDENTIFICATION="zh_CN.UTF-8"
```

```
LC_ALL=
```

```
LANG="zh_CN.UTF-8"
```

```
LANGUAGE="zh_CN:zh"
```

```
$ sudo nano /etc/default/locale
```

```
LANG="en_US.UTF-8"
```

```
LANGUAGE="en_US:en"
```

```
$ locale
```

```
LANG=en_US.UTF-8
```

```
LANGUAGE=en_US:en
```

```
LC_CTYPE="en_US.UTF-8"
```

```
LC_NUMERIC="en_US.UTF-8"
```



```
LC_TIME="en_US.UTF-8"  
LC_COLLATE="en_US.UTF-8"  
LC_MONETARY="en_US.UTF-8"  
LC_MESSAGES="en_US.UTF-8"  
LC_PAPER="en_US.UTF-8"  
LC="en_US.UTF-8"  
LC_ADDRESS="en_US.UTF-8"  
LC_TELEPHONE="en_US.UTF-8"  
LC_MEASUREMENT="en_US.UTF-8"  
LC_IDENTIFICATION="en_US.UTF-8"  
LC_ALL=
```

2.2.4 安全补丁、版本升级

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

```
$ sudo apt-get dist-upgrade
```

```
$ sudo do-release-upgrade
```

2.3 Ubuntu 快速入门指南

2.3.3 快速查找文件

1. find 命令

```
$ find /usr/share/doc -name *.txt
```

```
$ find /tmp -name core | xargs /bin/rm -f
```

```
$ find $HOME -mtime 0
```

```
$ find . -perm 664
```

2. locate 命令

```
$ sudo updatedb
```



```
$ locate apt-get
```

```
/usr/bin/apt-get
/usr/share/man/es/man8/apt-get.8.gz
/usr/share/man/fr/man8/apt-get.8.gz
/usr/share/man/ja/man8/apt-get.8.gz
/usr/share/man/man8/apt-get.8.gz
```

```
$ locate apt-get -c
```

```
5
```

2.3.4 软件包管理

```
$ apt-cache show php5-mysql
```

```
Package: php5-mysql
Priority: optional
Section: web
Installed-Size: 236
Maintainer: Ubuntu Core Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian PHP Maintainers <pkg-php-maint@lists.alioth.
debian.org>
Architecture: i386
Source: php5
Version: 5.2.4-2ubuntu5.3
Replaces: php5-mysqli
Depends: libc6 (>= 2.4), libmysqlclient15off (>= 5.0.27-1), php5-common (=
5.2.4-2 ubuntu5.3), phpapi-20060613+lfs
Conflicts: php5-mysqli
Filename: pool/main/p/php5/php5-mysql_5.2.4-2ubuntu5.3_i386.deb
Size: 65242
MD5sum: 003114b8d97dd35d435763338b3113f7
SHA1: 8d977486b1098c54b54036815398223a191e590d
SHA256: 8d8b301f1a1e85891d5da5c5fdebd9d7c16d7828b4be84640d824e881fe9f1df
Description: MySQL module for php5
This package provides modules for MySQL database connections directly from
PHP scripts. It includes the generic "mysql" module which can be used
to connect to all versions of MySQL, an improved "mysqli" module for
MySQL version 4.1 or later, and the pdo_mysql module for use with
the PHP Data Object extension.
.
PHP5 is an HTML-embedded scripting language. Much of its syntax is borrowed
from C, Java and Perl with a couple of unique PHP-specific features thrown
in. The goal of the language is to allow web developers to write
dynamically generated pages quickly.
Bugs: mailto:ubuntu-users@lists.ubuntu.com
Origin: Ubuntu
Task: lamp-server
```

2.3.5 使用 apt 工具

2. /etc/apt/sources.list 文件

deb(或 deb-src)	网络地址	主版本代号	软件仓库 1	软件仓库 2	软件仓库 3	...
----------------	------	-------	--------	--------	--------	-----

```
deb http://archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://archive.ubuntu.com/ubuntu/ hardy main restricted

deb http://archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://archive.ubuntu.com/ubuntu/ hardy-updates main restricted

deb http://archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://archive.ubuntu.com/ubuntu/ hardy universe
deb http://archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://archive.ubuntu.com/ubuntu/ hardy-updates universe

deb http://archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://archive.ubuntu.com/ubuntu/ hardy-updates multiverse

deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

```
$ sudo cp /etc/apt/sources.list /etc/apt/sources.list-backup
```

3. apt-get 命令

```
$ sudo apt-get install php5-mysql apache2-mpm-prefork libapache2-mod-php5
```

```
$ sudo apt-get update && sudo apt-get upgrade
```

4. apt-cache 命令

```
$ apt-cache search mysql
```

```
$ apt-cache search mysql | grep server
```

```
$ apt-cache show ssh
```



5. aptitude 命令

```
$ sudo aptitude update
```

```
$ sudo aptitude upgrade
```

```
$ aptitude search mysql | grep server
```

```
$ sudo aptitude clean --purge-unused
```

6. tasksel 命令

```
$ tasksel --task-packages lamp-server
```

```
apache2  
mysql-client-5.0  
libapache2-mod-php5  
apache2.2-common  
apache2-utils  
php5-common  
libaprutil1  
php5-mysql  
libmysqlclient15off  
libdbi-perl  
mysql-server  
libplrpc-perl  
mysql-server-5.0  
libdbd-mysql-perl  
libnet-daemon-perl  
libapr1  
libxml2  
libpcre3  
libpq5  
apache2-mpm-prefork  
mysql-common
```

```
$ tasksel --list-tasks
```

```
u dns-server      DNS server  
u edubuntu-server Edubuntu server  
u lamp-server     LAMP server  
u mail-server     Mail server  
i openssh-server  OpenSSH server  
u postgresql-server PostgreSQL database  
u print-server    Print server  
u samba-server    Samba File server
```

```
$ tasksel install lamp-server
$ tasksel remove lamp-server
```

7. dpkg 命令

```
$ dpkg -l apt
```

```
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-f/Unpacked/Failed-cfg/Half-inst/t-aWait/T-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name                Version              Description
+++-----
ii apt                  0.7.9ubuntu17       Advanced front-end for dpkg
```

```
$ dpkg -l apache
```

```
No packages found matching apache.
```

```
$ dpkg -L whiptail
```

```
/.
/usr
/usr/bin
/usr/bin/whiptail
/usr/share
/usr/share/doc
/usr/share/doc/whiptail
/usr/share/doc/whiptail/README.whiptail
/usr/share/doc/whiptail/copyright
/usr/share/doc/whiptail/newt.spec.gz
/usr/share/doc/whiptail/changelog.Debian.gz
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/whiptail.1.gz
```

```
$ dpkg -S /bin/ls
```

```
coreutils: /bin/ls
```

```
$ dpkg -C
```

```
$ man dpkg
```

8. dpkg-reconfigure 命令

```
$ sudo dpkg-reconfigure postfix
```



9. 给 apt 设置代理服务器

```
$ export http_proxy=http://yourproxyaddress:proxyport
```

```
$ sudo nano /etc/apt/apt.conf
```

```
Acquire::http::Proxy "http://yourproxyaddress:proxyport";
```

2.3.6 给 Red Hat 用户

1. 关于 root 用户

```
$ sudo su
[sudo] password for hiweed:
#
```

2.3.7 Ubuntu 网络配置

1. 网络配置文件/etc/network/interfaces

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

```
auto eth0
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    gateway 192.168.1.1
```

```
$ sudo /etc/init.d/networking restart
```

```
$ sudo ifdown eth0
$ sudo ifup eth0
```

2. 域名服务器配置文件/etc/resolv.conf

```
search localdomain
nameserver 192.168.1.1
nameserver 202.102.14.68
```

3. /etc/hosts 文件

```
127.0.0.1        localhost
127.0.1.1        ubuntu.localdomain    ubuntu

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ff02::3         ip6-allhosts
```



4. TCP/IP 协议简介

```
$ ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0c:29:f1:fb:b9
          inet addr:192.168.1.140  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feff:fb9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:47 errors:0 dropped:0 overruns:0 frame:0
          TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5967 (5.8 KB)  TX bytes:7288 (7.1 KB)
          Interrupt:17 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

6. NTP 时间同步

```
$ sudo nano /etc/cron.daily/timeupdate
```

```
ntpdate ntp.ubuntu.com
```

```
$ sudo chmod 755 /etc/cron.daily/timeupdate
```

```
ntpdate ntp.ubuntu.com pool.ntp.org
```

2.3.8 远程管理 Ubuntu Server

```
$ sudo apt-get install openssh-server
```

```
$ ssh 192.168.1.10 -p 3322
```

```
$ ssh hiweed@192.168.1.10 -p 3322
```


2.3.9 系统更新: apt-get update && apt-get upgrade

```
$ sudo apt-get update && apt-get upgrade
```

第 3 章



用 Apache 做 Web 服务器



3.2 Apache 的安装、配置

3.2.1 Apache 的安装

```
$ sudo apt-get install apache2
```

3.2.2 Apache 的配置

2. Apache 模块

```
$ sudo a2enmod
```

Which module would you like to enable?

Your choices are: actions alias asis auth_basic auth_digest authn_alias
authn_anon authn_dbd authn_dbm authn_default authn_file authnz_ldap authz_dbm
authz_default authz_groupfile authz_host authz_owner authz_user autoindex cache
cern_meta cgid c gi charset_lite dav_fs dav dav_lock dbd deflate dir disk_cache
dump_io env expires ext_filter file_cache filter headers ident imagemap include
info ldap log_forensic mem_cache mime mime_magic negotiation php5 proxy_ajp
proxy_balancer proxy_connect proxy_ftp proxy_http proxy_rewrite setenvif spelling
ssl status substitute suexec u nique_id userdir usertrack version vhost_alias

Module name?

```
$ sudo a2dismod
```

Which module would you like to disable?

Your choices are: alias auth_basic authn_file authz_default authz_groupfile
authz_host authz_user autoindex cgi dir env mime negotiation php5 rewrite
setenvif status

Module name?



3.2.3 Apache 虚拟主机

1. 创建一个新的虚拟主机

```
$ sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/  
blog.mytest.com
```

```
$ sudo nano /etc/apache2/sites-available/blog.mytest.com
```

```
$ sudo mkdir /var/www/blog.mytest.com  
$ echo "<h1>Oh yeah~</h1>" | sudo tee /var/www/blog.mytest.com/index.html
```

```
$ sudo a2dissite default && sudo a2ensite blog.mytest.com  
$ sudo /etc/init.d/apache2 restart
```

2. 虚拟主机配置详解

```
NameVirtualHost *  
<VirtualHost *>  
    ServerAdmin webmaster@localhost  
  
    DocumentRoot /var/www/  
    <Directory />  
        Options FollowSymLinks  
        AllowOverride None  
    </Directory>  
    <Directory /var/www/>  
        Options Indexes FollowSymLinks MultiViews  
        AllowOverride None  
        Order allow,deny  
        allow from all  
    </Directory>  
  
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
    <Directory "/usr/lib/cgi-bin">  
        AllowOverride None  
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch  
        Order allow,deny  
        Allow from all  
    </Directory>  
  
    ErrorLog /var/log/apache2/error.log  
  
    # Possible values include: debug, info, notice, warn, error, crit,  
    # alert, emerg.  
    LogLevel warn  
  
    CustomLog /var/log/apache2/access.log combined  
    ServerSignature On  
  
    Alias /doc/ "/usr/share/doc/"
```

```
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

```
</VirtualHost>
```

(1) NameVirtualHost 指令

```
NameVirtualHost 192.168.1.10:8080
```

```
NameVirtualHost *
```

(2) <VirtualHost></VirtualHost>指令

```
<VirtualHost IP地址[:端口号] [IP地址[:端口号]] ...>
...
</VirtualHost>
```

```
<VirtualHost 192.168.1.10>
    ServerAdmin webmaster@mytest.com
    DocumentRoot /www/docs/www.mytest.com
    ServerName www.mytest.com
    ErrorLog logs/www.mytest.com-error_log
    TransferLog logs/www.mytest.com-access_log
</VirtualHost>
```

(3) ServerAdmin 指令

```
ServerAdmin E-mail地址
```

```
ServerAdmin webmaster@hiweed.com
```

(4) DocumentRoot 指令

```
DocumentRoot /var/www/blog.mytest.com
```

(5) <Directory></Directory>指令

```
<Directory /var/www/blog.mytest.com>
...
</Directory>
```

```
<Directory /var/www/*.mytest.com>
... # 将匹配/var/www/目录下所有以.mytest.com 结尾的目录
</Directory>
```

```
<Directory ~ "^/var/www/.*/[0-9]{3}">
... # 将匹配/var/www/目录下所有由 3 位数字构成的目录
</Directory>
```

(6) Options 指令

```
<Directory /var/www>
Options Indexes FollowSymLinks
</Directory>
```



```
<Directory /var/www/spec>
Options Includes
</Directory>
```

```
<Directory /var/www>
Options Indexes FollowSymLinks
</Directory>
```

```
<Directory /var/www/spec>
Options +Includes -Indexes
</Directory>
```

(7) AllowOverride 指令

```
AllowOverride All | None | directive-type [directive-type] ...
```

(9) Allow 指令

```
Allow from all | host | env=env-variable [host | env=env-variable] ...
```

```
Allow from apache.org
Allow from .net example.edu
```

```
Allow from 10.1.2.3
Allow from 192.168.1.104 192.168.1.205
```

```
Allow from 10.1
Allow from 10 172.20 192.168.2
```

```
Allow from 10.1.0.0/255.255.0.0
```

```
Allow from 10.1.0.0/16
```

(11) ErrorLog 指令

```
ErrorLog /var/log/apache/error_log
```

```
ErrorLog "|/usr/local/bin/httpd_errors"
```

(13) CustomLog 指令

```
CustomLog file|pipe format|nickname [env=[!]environment-variable]
```

```
# 使用 nickname
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

```
# 使用格式字符串
CustomLog logs/access_log "%h %l %u %t \"%r\" %>s %b"
```

(14) ServerSignature 指令

Apache/2.2.6 (Ubuntu) Server at blog.mytest.com Port 80

(15) Alias 指令

Alias URL-path file-path|directory-path

Alias /doc/ "/usr/share/doc/"

3. HTTPS 的实现

```
$ sudo a2enmod ssl
```

```
$ sudo apt-get install openssl
```

```
$ openssl genrsa -des3 -out server.key 1024
```

```
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key: (在这里输入密码, 越复杂就越安全)
Verifying - Enter pass phrase for server.key: (再输入一次密码)
```

```
$ openssl genrsa -out server.key 1024
```

```
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
```

```
$ openssl req -new -key server.key -out server.csr
```

```
$ openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

```
$ sudo cp server.crt /etc/ssl/certs
$ sudo cp server.key /etc/ssl/private
```

```
SSL Engine on
```

```
SSLOptions +StrictRequire
```

```
SSLCertificateFile /etc/ssl/certs/server.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/server.key
```



```
$ sudo /etc/init.d/apache2 restart
```

4. Apache 排错

```
[warn] NameVirtualHost *:0 has no VirtualHosts
```

```
apache2: Could not determine the server's fully qualified domain name, using  
127.0.0.1 for ServerName
```

```
$ echo "ServerName localhost" | sudo tee /etc/apache2/conf.d/fqdn
```

3.3 Apache 性能优化

3.3.2 优化 Apache 配置

2. 优化 MaxClients

```
<IfModule mpm_worker_module>  
    StartServers      2  
    MaxClients        150  
    MinSpareThreads   25  
    MaxSpareThreads   75  
    ThreadsPerChild   25  
    MaxRequestsPerChild 0  
</IfModule>
```

```
<IfModule mpm_worker_module>  
    StartServers      10  
    MaxClients        256  
    MinSpareThreads   25  
    MaxSpareThreads   75  
    ThreadsPerChild   25  
    MaxRequestsPerChild 0  
</IfModule>
```

```
<IfModule mpm_worker_module>  
StartServers      10  
ServerLimit       512  
    MaxClients        512  
    MinSpareThreads   25  
    MaxSpareThreads   75  
    ThreadsPerChild   25  
    MaxRequestsPerChild 0  
</IfModule>
```



```
[error] server reached MaxClients setting, consider raising the MaxClients setting
```

4. 启用压缩

```
$ sudo a2enmod deflate
$ sudo /etc/init.d/apache2 force-reload
```

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain text/xml
</IfModule>
```

```
<IfModule mod_deflate.c>
    SetOutputFilter DEFLATE
    SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
    SetEnvIfNoCase Request_URI \.(?:exe|t?gz|zip|bz2|sit|rar)$ \
        no-gzip dont-vary
    SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary
</IfModule>
```

```
DeflateFilterNote Input input_info
DeflateFilterNote Output output_info
DeflateFilterNote Ratio ratio_info
LogFormat "%r" %{output_info}n/%{input_info}n (%{ratio_info}n%)' deflate
CustomLog /var/log/apache2/deflate_log deflate
```

```
"GET /a.html HTTP/1.1" 6508/181296 (3%)
```

6. 使用缓存 (mod_cache)

(1) mod_disk_cache 示例

```
$ sudo a2enmod disk_cache
```

```
<IfModule mod_disk_cache.c>
    CacheEnable disk /
    CacheRoot /var/www/blog.mytest.com/cache
    CacheDefaultExpire 7200
    CacheMaxExpire 604800
</IfModule>
```

```
CacheEnable disk /
```

```
CacheRoot /var/www/blog.mytest.com/cache
```



```
CacheDefaultExpire 7200
```

```
CacheMaxExpire 604800
```

```
$ sudo mkdir /var/www/blog.mytest.com/cache
$ sudo chown www-data:www-data /var/www/blog.mytest.com/cache
```

```
$ sudo /etc/init.d/apache2 restart
```

(2) mod_mem_cache 示例

```
<IfModule mod_mem_cache.c>
  CacheEnable mem /
  CacheDefaultExpire 7200
  CacheMaxExpire 604800
</IfModule>
```

(3) 不被 cache 的内容

```
CacheDisable /secure
```

3.4 Apache 压力测试 (ab)

```
ab [options] [http[s]://]hostname[:port]/path
```

```
$ ab -n 20000 -c 200 http://localhost/
```

```
This is ApacheBench, Version 2.0.40-dev <$Revision: 1.146 $> apache-2.0
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Copyright 2006 The Apache Software Foundation, http://www.apache.org/
```

```
Benchmarking localhost (be patient)
Completed 2000 requests
Completed 4000 requests
Completed 6000 requests
Completed 8000 requests
Completed 10000 requests
Completed 12000 requests
Completed 14000 requests
Completed 16000 requests
Completed 18000 requests
Finished 20000 requests
//以上为进度指示
```

```
Server Software:      Apache/2.2.8
Server Hostname:      localhost
Server Port:          80
```

```
//URL 路径
Document Path:      /
//文档长度
Document Length:    45 bytes

//并发数
Concurrency Level:   200
//测试所花的时间
Time taken for tests: 34.12302 seconds
//完成的请求总数
Complete requests:   20000
//失败的请求总数
Failed requests:     0
Write errors:        0
//总共传输的字节数
Total transferred:   7795696 bytes

//总共传输的 HTML 的字节数
HTML transferred:    904140 bytes
//平均每秒钟处理的请求数（mean 是平均的意思）
Requests per second: 588.02 [#/sec] (mean)

//平均每个请求所花的时间，单位是毫秒。这个时间，是每个请求从开始的那个时刻到结束的那个时刻的时间差的平均值
Time per request:    340.123 [ms] (mean)

//每个请求实际运行的平均时间，单位是毫秒。这个时间，是每个请求实际在 CPU 运行时间的平均值。对于并发请求，CPU 并不是同时处理的，而是按照每个请求获得的时间片逐个轮转处理的；所以，上面的时间约等于该时间乘以并发请求数（即 ab 的 -c 参数所指定的数值）
Time per request:    1.701 [ms] (mean, across all concurrent requests)

//传输速率，每秒钟收到的千字节（KB）数（如果流量过大，可能导致响应变慢）
Transfer rate:       223.80 [Kbytes/sec] received

//以下的时间详情，Hiweed 不是很了解
Connection Times (ms)
      min mean[+/-sd] median  max
Connect:    0 125 212.0   139   9284
Processing: 56 207 336.5   161  11349
Waiting:    45 179 331.5   146  11323
Total:     151 333 400.8   301  11353

//下面的数值，是相应时间内完成的请求的百分比。可以看出，在 301 毫秒内完成了 50% 的请求，换句话说，50% 的请求的响应时间都小于等于 301 毫秒；99% 的请求在 842 毫秒内被响应，最长的响应时间为 11353 毫秒
Percentage of the requests served within a certain time (ms)
 50%    301
 66%    318
 75%    330
 80%    341
 90%    387
 95%    527
 98%    759
 99%    842
100%  11353 (longest request)
```



3.5 Apache 安全

3.5.2 隐藏敏感信息

```
$ telnet localhost 80
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 03 Nov 2008 01:37:59 GMT
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.3 with Suhosin-Patch
Last-Modified: Mon, 03 Nov 2008 00:46:59 GMT
ETag: "34943-2d-45abe48d446c0"
Accept-Ranges: bytes
Content-Length: 45
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

```
ServerTokens Prod
```

```
$ sudo /etc/init.d/apache2 reload
```

```
$ telnet localhost 80
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 03 Nov 2008 02:08:43 GMT
Server: Apache
Last-Modified: Mon, 03 Nov 2008 00:46:59 GMT
ETag: "34943-2d-45abe48d446c0"
Accept-Ranges: bytes
Content-Length: 45
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

3.5.3 不要以 root 身份运行 Apache

```
$ ps auxf | grep apache
```

```
hiweed    5536  0.0  0.2   3004   756 pts/0    S+   21:29   0:00 \_ grep
```

```
apache
root      5420  0.0  2.4 18524  6236 ?        Ss   20:36   0:00 /usr/sbin/apache2
-k start
www-data  5505  0.0  1.4 18524  3652 ?        S    21:08   0:00 \_ /usr/sbin/
apache2 -k start
www-data  5506  0.0  1.2 18524  3188 ?        S    21:08   0:00 \_ /usr/sbin/
apache2 -k start
www-data  5507  0.0  1.2 18524  3188 ?        S    21:08   0:00 \_ /usr/sbin/
apache2 -k start
www-data  5508  0.0  1.2 18524  3188 ?        S    21:08   0:00 \_ /usr/sbin/
apache2 -k start
www-data  5509  0.0  1.2 18524  3188 ?        S    21:08   0:00 \_ /usr/sbin/
apache2 -k start
```

```
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

```
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
```

```
$ cat /etc/group|grep www
$ cat /etc/passwd|grep www
```

```
$ sudo groupadd www-data
$ sudo useradd -g www-data www-data
```

```
$ sudo /etc/init.d/apache2 restart
```

3.5.4 密码认证

1. 基本认证

```
$ sudo nano /etc/apache2/sites-available/blog.mytest.com
```

```
<Directory /var/www/blog.mytest.com/private>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride AuthConfig
    Order allow,deny
    allow from all
</Directory>
```

```
$ sudo /etc/init.d/apache2 reload
```



```
$ sudo mkdir /var/www/blog.mytest.com/auth
$ sudo chmod a+rx /var/www/blog.mytest.com/auth
$ cd /var/www/blog.mytest.com/auth
```

```
$ sudo htpasswd -bc private.passwords username password
```

Adding password for user username

```
$ sudo mkdir /var/www/blog.mytest.com/private
$ cd /var/www/blog.mytest.com/private
$ sudo nano .htaccess
```

```
AuthName "Password Needed"
AuthType Basic
AuthUserFile /var/www/blog.mytest.com/auth/private.passwords
Require valid-user
```

2. 摘要式认证

```
$ sudo a2enmod auth_digest
$ sudo /etc/init.d/apache2 restart
```

```
AuthType Digest
AuthName "Please Give Your Password"
AuthDigestDomain /var/www/blog.mytest.com/private
AuthUserFile /var/www/blog.mytest.com/auth/digest.passwords
require valid-user
```

```
$ cd /var/www/blog.mytest.com/private
$ sudo htdigest -c digest.passwords "Please Give Your Password" username
Adding password for username in realm Please Give Your Password.
New password:
Re-type new password:
```

3.5.5 检查文件权限

```
$ sudo chmod 777 index.cgi
```

```
$ sudo chmod 755 index.cgi
```

```
Options FollowSymLinks
AllowOverride None
```

3.5.6 关闭不用的模块

```
$ sudo a2dismod
```

```
Which module would you like to disable?
Your choices are: alias auth_basic authn_file authz_default authz_groupfile
authz_host authz_user autoindex cgi dir env mime negotiation php5 setenvif
status
Module name? auth_digest <-- 此处我们输入“auth_digest”
Module auth_digest disabled; run /etc/init.d/apache2 force-reload to fully
disable.
```

```
$ sudo /etc/init.d/apache2 force-reload
```

```
* Reloading web server config apache2 [OK]
```

3.5.7 DDoS 攻击防范

1. mod-evasive 的工作原理

```
Nov 3 22:48:50 ubuntu mod_evasive[4255]: Blacklisting address xxx.xxx.xxx.xxx:
possible DoS attack.
```

2. mod-evasive 的安装

```
$ sudo apt-get install libapache2-mod-evasive
```

3. mod-evasive 的配置

```
$ sudo nano /etc/apache2/conf.d/evasive
```

```
<IfModule mod_evasive20.c>
    DOSHashTableSize    3097
    DOSPageCount        2
    DOSSiteCount        50
    DOSPageInterval     1
    DOSSiteInterval     1
    DOSBlockingPeriod   10

#还可以加入以下备选配置
    DOSEmailNotify      you@yourdomain.com
    DOSSystemCommand    "su - someuser -c '/sbin/... %s ...'"
    DOSLogDir            "/var/lock/mod_evasive"

#还可以加入以下的“白名单”配置
    DOSWhitelist        127.0.0.1
    DOSWhitelist        127.0.0.*
</IfModule>
```

4. DDoS 攻击测试

```
$ cd /usr/share/doc/libapache2-mod-evasive/examples
$ perl test.pl
```



```
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
HTTP/1.1 403 Forbidden
```

3.6 Apache 日志分析

3.6.1 用 Webalizer 分析 Apache 日志

1. 安装 Webalizer

```
$ sudo apt-get install webalizer
```

```
$ sudo webalizer
```

2. 配置 Webalizer

```
LogFile /var/log/apache2/access.log.1
```

```
#LogType      clf
```

```
OutputDir /var/www/webalizer
```

```
HostName      xxxxxx
```

```
PageType      htm*
```

3.6.2 用 AWStats 分析 Apache 日志

1. 安装 AWStats

```
$ sudo apt-get install awstats
```

```
0,10,20,30,40,50 * * * * www-data [ -x /usr/lib/cgi-bin/awstats.pl -a -f
/etc/awstats/awstats.conf -a -r /var/log/apache/access.log ] && /usr/lib/cgi-
bin/awstats.pl -config=awstats -update >/dev/null
```

2. 配置 Apache

```
$ sudo nano /etc/apache2/awstats.conf
```



```
Alias /awstatsclasses "/usr/share/awstats/lib/"
Alias /awstats-icon/ "/usr/share/awstats/icon/"
Alias /awstatscss "/usr/share/doc/awstats/examples/css"
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
ScriptAlias /awstats/ /usr/lib/cgi-bin/
Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
```

```
$ sudo nano /etc/apache2/apache2.conf
```

```
Include /etc/apache2/awstats.conf
```

```
$ sudo /etc/init.d/apache2 reload
```

3. 配置 AWStats

```
$ sudo cp /etc/awstats/awstats.conf /etc/awstats/awstats.192.168.1.10.conf
```

```
$ sudo nano /etc/awstats/awstats.192.168.1.10.conf
```

```
LogFile="/var/log/apache2/access.log"
SiteDomain="mytest.com"
```

```
$ sudo /usr/bin/perl /usr/lib/cgi-bin/awstats.pl -update -config=192.168.1.10
```

```
0 4 * * * www-data [ -x /usr/lib/cgi-bin/awstats.pl -a -f /etc/awstats/
awstats.conf -a -r /var/log/apache/access.log ] && /usr/lib/cgi-bin/awstats.pl -
config=awstats -update >/dev/null
```

3.6.3 Apache 日志合并

```
$ sudo cp /usr/share/doc/awstats/examples/logresolvemerge.pl /usr/local/bin
```

```
127.0.0.1 - - [02/Nov/2008:20:17:25 -0500] "GET /" 200 45 "-" "-"
127.0.0.1 - - [02/Nov/2008:20:17:26 -0500] "GET /" 200 45 "-" "-"
192.168.1.119 - - [02/Nov/2008:20:18:04 -0500] "GET / HTTP/1.1" 200 45 "-"
192.168.1.119 - - [02/Nov/2008:20:18:04 -0500] "GET /favicon.ico HTTP/1.1" 404
```



```
127.0.0.1 - - [02/Nov/2008:20:17:26 -0500] "GET /" 200 45 "-" "-"
127.0.0.1 - - [02/Nov/2008:20:17:27 -0500] "GET /" 200 45 "-" "-"
192.168.1.119 - - [02/Nov/2008:20:18:05 -0500] "GET / HTTP/1.1" 200 45 "-"
192.168.1.119 - - [02/Nov/2008:20:18:05 -0500] "GET /favicon.ico HTTP/1.1" 404
127.0.0.1 - - [02/Nov/2008:20:18:19 -0500] "KKK" 501 289 "-" "-"
```

```
127.0.0.1 - - [02/Nov/2008:20:17:25 -0500] "GET /" 200 45 "-" "-"
127.0.0.1 - - [02/Nov/2008:20:17:26 -0500] "GET /" 200 45 "-" "-"
127.0.0.1 - - [02/Nov/2008:20:17:27 -0500] "GET /" 200 45 "-" "-"
192.168.1.119 - - [02/Nov/2008:20:18:04 -0500] "GET / HTTP/1.1" 200 45 "-"
192.168.1.119 - - [02/Nov/2008:20:18:04 -0500] "GET /favicon.ico HTTP/1.1" 404
192.168.1.119 - - [02/Nov/2008:20:18:05 -0500] "GET / HTTP/1.1" 200 45 "-"
192.168.1.119 - - [02/Nov/2008:20:18:05 -0500] "GET /favicon.ico HTTP/1.1" 404
127.0.0.1 - - [02/Nov/2008:20:18:19 -0500] "KKK" 501 289 "-" "-"
```

```
$ sudo su -c "logresolvemerge.pl /var/log/webcluster/access_log_server* >
/var/log/webcluster/access_log_overall"
```

第 4 章

LAMP 服务器之 PHP 篇

4.3 安装 LAMP 相关软件包

4.3.1 LAMP 软件包安装

```
$ sudo apt-get install apache2 libapache2-mod-php5 php5-mysql mysql-server
```

New password for the MySQL "root" user: <-- 输入密码

Repeat password for the MySQL "root" user: <-- 再输入一次

```
$ mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('yourpassword');
mysql> SET PASSWORD FOR 'root'@'ubox.mytest.com' = PASSWORD('yourpassword');
```

```
$ sudo apt-get install php5-memcache
```



4.3.2 LAMP 软件包删除

```
$ sudo apt-get remove apache2 apache2-mpm-prefork apache2-utils apache2.2- common
l ibapache2-mod-php5 libapr1 libaprutil1 libdbd-mysql-perl libdbi- perl
libmysqlclie nt15off libnet-daemon-perl libplrpc-perl libpq5 mysql-client-5.0
mysql-common mys ql-server mysql-server-5.0 php5-common php5-mysql
```

4.4 配置 Apache、MySQL、PHP

4.4.1 MySQL 配置

2. 创建 MySQL 数据库

```
DROP TABLE IF EXISTS `users`;
CREATE TABLE `users` (
  `uid` int(10) unsigned NOT NULL default '0',
  `name` varchar(60) NOT NULL default '',
  `pass` varchar(32) NOT NULL default '',
  `mail` varchar(64) default '',
  PRIMARY KEY (`uid`),
  UNIQUE KEY `name` (`name`)
);

INSERT INTO `users` (`uid`, `name`, `pass`, `mail`) VALUES
(1, 'Hiweed', MD5('passwdHiweed'), 'hiweed@test.com'),
(2, 'Ning', MD5('passwdNing'), 'ning@test.com'),
(3, 'Guoce', MD5('passwdGuoce'), 'guoce@test.com')
```

```
$ mysqladmin -uroot -p create mydb
$ mysql mydb -uroot -p < mydb.sql
```

```
$ mysql mydb -uroot -p
```

```
mysql> select * from users;
```

uid	name	pass	mail
1	Hiweed	e2d9dfef9b36eece2ab0117abd40445	hiweed@test.com
2	Ning	b942fc6f6b0cc6d1a0e665306c6fcc2b	ning@test.com
3	Guoce	aeaa8187b8df25e4f9a291ce2eb00291	guoce@test.com

3 rows in set (0.00 sec)

```
mysql> quit
```

```
Bye
```

3. 创建数据库用户并分配权限

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, INDEX, ALTER ON mydb.* TO  
'username'@'localhost' IDENTIFIED BY 'password';  
mysql> flush privileges;
```

```
mysql> exit
```

4.4.2 PHP 配置

1. PHP 配置文件

```
;;;;;;;;;;;;;;;;;;;;;;;;;  
; Resource Limits ;  
;;;;;;;;;;;;;;;;;;;;;;;;;  
  
max_execution_time = 30    ; 单个 PHP 文件的最大运行时间，单位是秒  
max_input_time = 60       ; 单个 PHP 文件接收数据花费的最大时间，单位是秒  
memory_limit = 16M        ; 单个 PHP 文件可以占用的最大内存
```

2. PHP 测试

```
$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/blog.mytest.com/phpinfo.php
```

3. PHP 排错

```
$ sudo a2enmod php5  
$ sudo apache2ctl restart
```



4.6 用 phpMyAdmin 管理 MySQL 数据库

4.6.1 phpMyAdmin 的安装

```
$ sudo apt-get install phpmyadmin
```

```
$ sudo ln -s /usr/share/phpmyadmin /var/www/blog.mytest.com/phpmyadmin
```

4.6.2 phpMyAdmin 排错

```
$ sudo cat /var/lib/phpmyadmin/blowfish_secret.inc.php | grep blowfish_secret >>  
/ etc/phpmyadmin/config.inc.php
```

4.7 实例：用 Drupal 快速架设 Blog 网站

4.7.2 获取 Drupal

```
$ sudo rm /var/www/blog.mytest.com/index.html
```

```
$ wget http://ftp.osuosl.org/pub/drupal/files/projects/drupal-6.6.tar.gz  
$ tar xfvz drupal-6.6.tar.gz  
$ sudo mv drupal-6.6/{*,.htaccess} /var/www/blog.mytest.com
```

4.7.3 为 Drupal 创建 MySQL 数据库和用户

```
$ mysqladmin -uroot -p create drupal6
```

```
$ mysql -uroot -p
```

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE  
TEMPORARY TABLES, LOCK TABLES ON drupal6.* TO 'drupaluser'@'localhost'  
IDENTIFIED BY 'yourpassword';
```

```
mysql> FLUSH PRIVILEGES; <-- 刷新权限，使之生效
```

```
mysql> \q
```

4.7.4 为 Drupal 配置 PHP

```
memory_limit = 32M ; 单个 PHP 文件可以占用的最大内存
```

4.7.5 为 Drupal 配置 Apache

```
$ sudo a2enmod rewrite
```

```
<Directory /var/www/blog.mytest.com/>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride all  
    Order allow,deny  
    allow from all  
</Directory>
```

```
$ sudo /etc/init.d/apache2 force-reload <-- 重新装载 Apache2
```

4.7.6 安装 Drupal

```
$ sudo chmod o+w /var/www/blog.mytest.com/sites/default
```

```
$ sudo chmod o-w /var/www/blog.mytest.com/sites/default
```

第 5 章

LAMP 服务器之 Perl 篇

5.1 安装 Perl 模块

```
$ sudo apt-get install libapache2-mod-perl2
```

5.2 配置 cgi-bin 目录

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
<Directory "/usr/lib/cgi-bin">  
    AllowOverride None  
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch  
    Order allow,deny  
    Allow from all  
</Directory>
```

5.3 Perl 程序测试

```
$ sudo mkdir /usr/lib/cgi-bin/test  
$ sudo nano /usr/lib/cgi-bin/test/test.pl
```

```
#!/usr/bin/perl -w  
  
print "Content-type: text/html\n\n";  
print "Hello, World.";
```

```
$ sudo chmod a+x /usr/lib/cgi-bin/test/test.pl
```

5.4 用 Perl 访问 MySQL 数据库

```
#!/usr/bin/perl  
  
use DBI;
```



```
# 连接数据库
my $dbh = DBI->connect("DBI:mysql:database=mydb;host=localhost","username","password", { 'RaiseError' => 1 });

# 查询
my $sqr = $dbh->prepare("SELECT name, mail FROM users");
$sqr->execute();

# 打印结果
while(my $ref = $sqr->fetchrow_hashref()) {
    print "$ref->{'name'}, $ref->{'mail'}\n";
}

# 关闭数据库连接
$dbh->disconnect();
```

```
$ perl /var/www/dbtest.pl
```

```
Hiweed, hiweed@test.com
Ning, ning@test.com
Guoce, guoce@test.com
```

```
$ sudo cp /var/www/dbtest.pl /usr/lib/cgi-bin/test/
```

```
print "Content-type: text/html\n\n";
```

```
#!/usr/bin/perl

use DBI;

my $dbh = DBI->connect("DBI:mysql:database=mydb;host=localhost","username","password", { 'RaiseError' => 1 });

my $sqr = $dbh->prepare("SELECT name, mail FROM users");
$sqr->execute();

print "Content-type:text/html\n\n";

while(my $ref = $sqr->fetchrow_hashref()) {
    print "$ref->{'name'}, $ref->{'mail'}\n\n";
}

$dbh->disconnect();
```

```
$ sudo chmod 755 /usr/lib/cgi-bin/test/dbtest.pl
```



5.5 CGI 排错

```
#!/usr/bin/perl
```

```
print "Content-type: text/html\n\n";
```

```
$ chmod a+x test.pl
```

```
$ chmod 755 test.pl
```

5.6 实例：用 Twiki 假设 Wiki

5.6.1 安装 Twiki

```
$ sudo apt-get install twiki
```

5.6.2 配置 Twiki

```
RedirectMatch /twiki/?$ http://localhost/cgi-bin/twiki/view$1  
RedirectMatch /twiki(/([A-Z].*)?)?$ http://localhost/cgi-bin/twiki/view$1
```

3. Twiki 安全

```
$ sudo -u www-data htpasswd /var/lib/twiki/data/.htpasswd TWikiGuest
```

第 6 章

LAMP 服务器之 Python 篇

6.1 安装 mod_python

```
$ sudo apt-get install libapache2-mod-python
```

6.2 配置 Apache

6.2.1 Publisher Handler

```
AddHandler mod_python .py
PythonHandler mod_python.publisher
PythonDebug On
```

```
[...]
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    AddHandler mod_python .py
    PythonHandler mod_python.publisher
    PythonDebug On
</Directory>
[...]
```

```
$ sudo /etc/init.d/apache2 restart
```

```
$ sudo nano /var/www/test.py
```

```
def index(req):
    return "Hello World";
```

6.2.2 PSP Handler

```
AddHandler mod_python .psp
PythonHandler mod_python.psp
PythonDebug On
```

```
[...]
<Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    AddHandler mod_python .psp
    PythonHandler mod_python.psp
    PythonDebug On
</Directory>
[...]
```

```
$ sudo /etc/init.d/apache2 restart
```

```
$ sudo nano /var/www/test.psp
```

```
<html>
<body>
<h1><% req.write("Hello, PSP World") %></h1>
</body>
</html>
```

6.3 让 Python 支持 MySQL

```
$ sudo apt-get install python-mysqldb
```

6.3.1 Python 连接 MySQL 数据库测试

```
$ sudo nano /var/www/dbtest.py
```

```
#!/usr/bin/python

# 导入数据库模块
import MySQLdb

# 连接数据库
db
```



```
MySQLdb.connect(host="localhost",user="username",passwd="password",db="mydb"
)

# 创建一个游标
cursor = db.cursor()

# 执行 SQL 语句
cursor.execute("SELECT name, mail FROM users")

# 获取查询结果（数组）
result = cursor.fetchall()

# 打印查询结果
for record in result:
    print record[0] , "-->", record[1]
```

```
$ python /var/www/dbtest.py
```

```
Hiweed --> hiweed@test.com
Ning --> ning@test.com
Guoce --> guoce@test.com
```

6.3.2 Python 的 CGI 程序

```
print "Content-type:text/html\n"
```

```
#!/usr/bin/python

import MySQLdb
db
MySQLdb.connect(host="localhost",user="username",passwd="password",db="mydb"
)
cursor = db.cursor()
cursor.execute("SELECT name, mail FROM users")
result = cursor.fetchall()

print "Content-type:text/html\n"

for record in result:
    print record[0] , "-->", record[1]
```

```
$ sudo cp /var/www/dbtest.py /usr/lib/cgi-bin/test/
```

```
$ sudo chmod 755 /usr/lib/cgi-bin/test/dbtest.py
```

6.4 实例：用 Django 开发 Web 应用程序

6.4.1 安装 Django

```
$ sudo apt-get install python-django
```

6.4.2 创建自己的 Django 项目

```
$ cd ~
$ django-admin startproject mysite
```

```
mysite/
  init__.py      <-- 该文件告诉 Python，此目录是一个 Python Package
  manage.py      <-- 本项目的命令行管理工具
  settings.py    <-- 本项目的配置文件
  urls.py        <-- 用以设置 URL 的对应关系和样式
```

6.4.3 运行 Django 开发服务器

```
$ cd ~/mysite
$ manage.py runserver 192.168.1.10:8000
```

```
Validating models...
0 errors found.
```

```
Django version 0.96.1, using settings 'mysite.settings'
Development server is running at http://192.168.1.10:8000/
Quit the server with CONTROL-C.
```

6.4.4 连接 MySQL 数据库

```
DATABASE_ENGINE = 'mysql'
DATABASE_NAME = 'mydb'
DATABASE_USER = 'username'
DATABASE_PASSWORD = 'password'
DATABASE_HOST = 'localhost'
```

```
$ python manage.py syncdb
```

```
$ mysql -uusername -ppassword mydb
mysql> show tables;
```

```
+-----+
| Tables_in_mydb |
+-----+
| auth_group      |
```



```
| auth_group_permissions |
| auth_message          |
| auth_permission       |
| auth_user             |
| auth_user_groups      |
| auth_user_user_permissions |
| django_content_type   |
| django_session        |
| django_site           |
| users                 |
+-----+
11 rows in set (0.00 sec)
```

6.5 实例：用 MoinMoin 实现 Wiki

6.5.1 安装 MoinMoin

```
$ sudo apt-get install python-moinmoin
```

6.5.2 创建 MoinMoin 实例

1. 创建 Wiki 实例

```
$ cd /usr/share/moin/
```

```
$ sudo cp -R data mywiki
```

```
$ sudo cp -R underlay mywiki
```

```
$ sudo cp server/moin.cgi mywiki
```

```
$ sudo chown -R www-data.www-data mywiki
```

```
$ sudo chmod -R ug+rwX mywiki
$ sudo chmod -R o-rwx mywiki
```

2. 将实例添加到 MoinMoin 中

```
$ sudo nano /etc/moin/mywiki.py
```

```
data_dir = '/org/mywiki/data/'
```

```
data_dir = '/usr/share/moin/mywiki/data'
```

3. 配置 Apache

```
### moin
ScriptAlias /mywiki "/usr/share/moin/mywiki/moin.cgi"
alias /wiki "/usr/share/moin/htdocs"
<Directory /usr/share/moin/htdocs>
Order allow,deny
allow from all
</Directory>
### end moin
```

```
$ sudo /etc/init.d/apache2 restart
```

6.5.3 MoinMoin 权限控制

```
#acl hiweed:read,write All:read
```

1. 语法

```
#acl  [+~]User[,SomeGroup,...]:[right[,right,...]]  [[+~]OtherUser:...]  [[+~]
Truste d:...]  [[+~]Known:...]  [[+~]All:...]  [Default]
```

2. 定义组

```
#acl hiweed:read,write,admin,delete,revert All:read
* hiweed
* shanghao
* xiaoning
```

3. 组权限

```
#acl AdminGroup:read,write,revert All:read
```

6.6 Python Web 应用的性能优化

6.6.3 mod-wsgi 的安装

```
$ sudo apt-get remove --purge libapache2-mod-python
```

```
$ sudo nano /etc/apache2/sites-available/default
```

```
# AddHandler mod_python .py
```




```
# PythonHandler mod_python.publisher  
# PythonDebug On
```

```
$ sudo apt-get install libapache2-mod-wsgi
```

第 7 章

Apache Tomcat 架设

7.1 安装 Tomcat

```
$ sudo apt-get install sun-java5-jdk tomcat5.5 tomcat5.5-admin
```

```
JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun
```

```
$ sudo /etc/init.d/tomcat5.5 start
```

```
$ sudo /etc/init.d/tomcat5.5 stop  
$ sudo /etc/init.d/tomcat5.5 restart
```

```
$ java -version
```

7.2 配置 Tomcat

```
$ sudo nano /var/lib/tomcat5.5/conf/tomcat-users.xml
```

```
<?xml version='1.0' encoding='UTF-8'>  
<tomcat-users>  
  <role rolename="manager"/>  
  <role rolename="admin"/>  
  <user username="hiweed" password="HiPass" roles="admin,manager"/>  
</tomcat-users>
```

7.3 Tomcat 和 Apache 的整合: mod_jk



7.3.1 mod_jk 的安装

```
$ sudo apt-get install libapache2-mod-jk
```

7.3.2 mod_jk 的配置

```
$ sudo nano /etc/libapache2-mod-jk/workers.properties
```

```
workers.tomcat_home=/usr/share/tomcat5.5
workers.java_home=/usr/lib/jvm/java-1.5.0-sun
ps=/
worker.list=worker1
worker.ajp13_worker.port=8180
worker.ajp13_worker.host=localhost
worker.ajp13_worker.type=ajp13
worker.ajp13_worker.lbfactor=1
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=ajp13_worker
```

```
$ sudo nano /etc/apache2/apache.conf
```

```
# 告诉 JK 到哪里去找 workers.properties
JkWorkersFile /etc/libapache2-mod-jk/workers.properties

# 定义 JK 日志的位置
JkLogFile /var/log/apache2/mod_jk.log

# 设置 JK 日志的级别 [debug/error/info]
JkLogLevel info

# 设置 Log 的格式
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# JK 选项
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# 设置请求格式
JkRequestLogFormat "%w %V %T"
```

```
$ sudo nano /etc/apache2/sites-available/default
```

```
jkMount /* worker1
```

```
$ sudo /etc/init.d/apache2 restart
```

7.4 Tomcat 安全

7.4.1 保护 shutdown 端口

```
<Server port="8005" shutdown="SHUTDOWN">
```



7.4.2 修改默认错误页面

```
<error-page>
  <exception-type>java.lang.Throwable</exception-type>
  <location>/error.jsp</location>
</error-page>
```

7.4.4 Manager WebApp 安全

```
$ sudo nano /etc/tomcat5.5/Catalina/localhost/manager.xml
```

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
  allow="192.168.1.*" />
```

```
<Valve className="org.apache.catalina.valves.RemoteHostValve"
  allow="*.localdomain.com" />
```

第 8 章

最佳代理、反向代理服务器：Squid

8.1 Squid 安装

```
$ sudo apt-get install squid
```

```
FATAL: Could not determine fully qualified hostname. Please set 'visible_hostname'
```

8.2 为 Squid 配置主机名

```
$ sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.backup  
$ sudo chmod a-w /etc/squid/squid.conf.backup
```

```
$ cd /etc/squid/  
$ sudo cat squid.conf.backup | grep -v ^$ | grep -v ^# | sudo tee squid.conf
```

```
$ sudo nano /etc/squid/squid.conf
```

```
visible_hostname ubproxy
```

```
$ sudo /etc/init.d/squid restart
```

8.3 访问控制列表

```
acl name type value1 value2 ...
```

例如：

```
acl NormalUsers src 192.168.1.0/24, 192.168.2.0/24
```

```
acl NormalUsers src 192.168.1.0/24  
acl NormalUsers src 192.168.2.0/24
```

```
http_access deny NormalUser
```



8.4 正向代理

8.4.1 设置端口号

```
http_port 8888
```

8.4.2 禁止某些 IP 地址上网

```
acl WorkShop src 192.168.1.0-192.168.2.0/24
```

```
acl WorkShop src 192.168.1.0/24  
acl WorkShop src 192.168.2.0/24
```

```
acl WorkShop src 192.168.1.0/24, 192.168.2.0/24
```

```
http_access deny WorkShop
```

8.4.3 禁止在某时间段上网

```
acl NormalUsers src 192.168.1.0/24  
acl WorkingHours time D 09:00-10:00  
http_access deny !WorkingHours NormalUsers
```

```
http_access allow NormalUsers WorkingHours
```

8.4.4 个别网站的控制

```
$ sudo nano /etc/squid/allowedSites.list
```

```
hiweed.com  
aixingzou.cn
```

```
$ sudo nano /etc/squid/deniedSites.list
```

```
www.illegalsite.com  
abcdef.com
```

```
acl office_network src 192.168.1.0/24  
acl GoodSites dstdomain "/etc/squid/allowedSites.list"  
acl BadSites dstdomain "/etc/squid/denySites.list"
```



```
http_access deny BadSites
http_access allow office_network GoodSites
```

8.4.5 用 NCSA 做密码认证

```
$ sudo touch /etc/squid/auth-password
$ sudo chmod o+r /etc/squid/auth-password
```

```
$ sudo htpasswd /etc/squid/auth-password username
```

```
New password:
Re-type new password:
Adding password for user username
```

```
$ dpkg -L squid | grep ncsa_auth
/usr/lib/squid/ncsa_auth
```

```
# 定义认证程序和密码文件的位置
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/auth-password
# 定义派生认证进程的数量
auth_param basic children 5
# 要求输入用户名和密码时显示的信息
auth_param basic realm Please Login First
# 每隔 2 小时就重新认证一次
auth_param basic credentialsttl 2 hours
# 大小写敏感: 关闭 (对用户名不区分大小写)
auth_param basic casesensitive off

acl ncsa_users proxy_auth REQUIRED
http_access allow ncsa_users
```

8.4.6 透明代理的设置

1. 服务器网卡配置

```
$ cat /etc/network/interfaces

auto eth1
iface eth1 inet static
address 192.168.1.10
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

3. Squid 的透明代理配置

```
http_port 192.168.1.10:3128 transparent
```



4. iptables 防火墙的配置

```
$ iptables --list
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

```
$ sudo iptables -t nat -A PREROUTING -i eth1 -p tcp \
--dport 80 -j REDIRECT --to-port 3128
$ sudo iptables -A INPUT -j ACCEPT -m state \
--state NEW,ESTABLISHED,RELATED -i eth1 -p tcp \
--dport 3128
$ sudo iptables -A OUTPUT -j ACCEPT -m state \
--state NEW,ESTABLISHED,RELATED -o eth0 -p tcp \
--dport 80
$ sudo iptables -A INPUT -j ACCEPT -m state \
--state ESTABLISHED,RELATED -i eth0 -p tcp \
--sport 80
$ sudo iptables -A OUTPUT -j ACCEPT -m state \
--state ESTABLISHED,RELATED -o eth1 -p tcp \
--sport 80
```

```
$ iptables -L
```

```
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     tcp  --  anywhere              anywhere        state    NEW,RELATED,ESTABLIS
HED tcp dpt:3128
ACCEPT     tcp  --  anywhere              anywhere        state    RELATED,ESTABLISHED
tcp spt:www

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     tcp  --  anywhere              anywhere        state    NEW,RELATED,ESTABLIS
HED tcp dpt:www
ACCEPT     tcp  --  anywhere              anywhere        state    RELATED,ESTABLISHED
tcp spt:www
```

5. 保存 iptables 规则

```
$ sudo sh -c "iptables-save > /etc/iptables.rules"
```

```
pre-up iptables-restore < /etc/iptables.rules
```

```
post-down iptables-save -c > /etc/iptables.rules
```

```
auto eth1
iface eth1 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    pre-up iptables-restore < /etc/iptables.rules
    post-down iptables-save -c > /etc/iptables.rules
```

8.5 反向代理

8.5.1 Squid 反向代理单个后台 Web 服务器

1. Web 和 Squid 在同一台机器上

```
http_port 80 vhost vport
cache_peer 127.0.0.1 parent 81 0 no-query originserver
```

2. Web 和 Squid 在不同的机器上

```
http_port 80 vhost vport
cache_peer 221.214.14.185 parent 80 0 no-query originserver
```

8.5.2 Squid 反向代理多个后台 Web 服务器

```
192.168.1.10 news.163.com
192.168.1.10 news.baidu.com
192.168.1.10 news.google.com
```

```
202.108.9.79 news.163.com
61.135.163.87 news.baidu.com
209.85.175.99 news.google.com
```

```
acl ServerIPs dst 202.108.9.79 61.135.163.87 209.85.175.99
acl ServerDomains dstdomain news.163.com news.baidu.com news.google.com
```

```
always_direct allow ServerDomains
never_direct allow !ServerDomains
http_access allow ServerIPs
http_access allow ServerDomains
```



8.6 Squid 排错

8.6.1 Squid 运行状态检查

```
$ sudo squid -NCd1
```

```
2009/06/22 09:56:26| Squid is already running! Process ID 4832
```

8.7 使用 SquidGuard

8.7.2 安装 SquidGuard

```
$ sudo apt-get install squidguard
```

8.7.3 SquidGuard 基本配置

1. 创建简单的 SquidGuard 配置文件

```
$ sudo mv /etc/squid/squidGuard.conf /etc/squid/squidGuard.conf-orig  
$ sudo nano /etc/squid/squidGuard.conf
```

```
#  
# CONFIG FILE FOR SQUIDGUARD  
#  
  
dbhome /var/lib/squidguard/db/blacklists  
logdir /var/log/squid  
  
dest spyware {  
    domainlist spyware/domains  
    urllist spyware/urls  
}  
  
acl {  
    default {  
        pass !spyware all  
        redirect http://192.168.1.10/block.html  
    }  
}
```

2. 准备黑名单

```
$ sudo su
```

```
# cd /var/lib/squidguard/db/
```

```
# wget http://squidguard.mesd.k12.or.us/blacklists.tgz
```

```
# tar xfvz blacklists.tgz
```

```
# chown proxy:proxy -R /var/lib/squidguard/db/*
```

```
# find /var/lib/squidguard/db -type f | xargs chmod 644
# find /var/lib/squidguard/db -type d | xargs chmod 755
```

```
# sudo -u proxy squidGuard -C all
```

3. 测试黑名单数据库

```
$ sudo su
# echo "http://hiweed.com / - - GET" | squidGuard -d
```

```
2009-03-20 04:24:31 [5371] init domainlist /var/lib/squidguard/db/blacklists/
spywa re/domains
2009-03-20 04:24:31 [5371] loading dbfile /var/lib/squidguard/db/blacklists/
spywar e/domains.db
2009-03-20 04:24:31 [5371] init urllist /var/lib/squidguard/db/blacklists/
spyware/ urls
2009-03-20 04:24:31 [5371] loading dbfile /var/lib/squidguard/db/blacklists/
spywar e/urls.db
2009-03-20 04:24:31 [5371] squidGuard 1.2.0 started (1237537471.949)
2009-03-20 04:24:31 [5371] squidGuard ready for requests (1237537471.953)

2009-03-20 04:24:31 [5371] squidGuard stopped (1237537471.955)
```

```
# exit
```

4. 准备 block.html

```
$ sudo nano /var/www/block.html
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="Expires" content="0">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>请勿访问非法网站
</title>
```

5. 让 Squid 使用 SquidGuard

```
redirect program /usr/bin/squidGuard
```

7. 黑名单或配置更新

```
$ echo "http://hiweed.com / - - GET" | sudo squidGuard -d
```

127

```
conf line 14
```

```
$ sudo /etc/init.d/squid reload
```

8.7.4 SquidGuard 高级配置

1. 禁止使用 IP 地址访问 Web

```
$ sudo nano /etc/squid/squidGuard.conf
```

```
acl {  
    default {  
        pass !in-addr !spyware all  
        redirect http://192.168.1.10/block.html  
    }  
}
```

:

```
$ echo "http://hiweed.com / - - GET" | sudo squidGuard -d
```

```
$ sudo /etc/init.d/squid reload
```

2. 设置时间段

```
$ sudo nano /etc/squid/squidGuard.conf
```

```
time afterwork {  
    weekly sat sun          # 周六、周日  
    weekly mtwhf 18:00-24:00 # 周一至周五的下班时间  
    date *.01.01           # 每年的元旦  
}
```

```
src admin {  
    ip 192.168.1.0/24  
}
```

```
acl {  
    admin within afterwork {  
        pass all  
    }  
    else {
```



```

        pass !in-addr !spyware all
    }
    default {
        pass none
        redirect http://192.168.1.10/block.html
    }
}

```

```
$ echo "http://hiweed.com / - - GET" | sudo squidGuard -d
```

```
$ sudo /etc/init.d/squid reload
```

3. 启用所有黑名单

```

#
# CONFIG FILE FOR SQUIDGUARD
#

dbhome /var/lib/squidguard/db/blacklists
logdir /var/log/squid

time afterwork {
    weekly sat sun           # 周六、周日
    weekly mtwhf 18:00-24:00 # 周一至周五的下班时间
    date *.01.01             # 每年的元旦
}

dest ads {
    domainlist ads/domains
    urllist ads/urls
}

dest aggressive {
    domainlist aggressive/domains
    urllist aggressive/urls
}

dest audio-video {
    domainlist audio-video/domains
    urllist audio-video/urls
}

dest drugs {
    domainlist drugs/domains
    urllist drugs/urls
}

dest gambling {
    domainlist gambling/domains
    urllist gambling/urls
}

dest hacking {
    domainlist hacking/domains

```




```
        urllist      hacking/urls
    }

    dest mail {
        domainlist    mail/domains
    }

    dest porn {
        domainlist     porn/domains
        urllist        porn/urls
    }

    dest proxy {
        domainlist     proxy/domains
        urllist        proxy/urls
    }

    dest redirector {
        domainlist     redirector/domains
        urllist        redirector/urls
    }

    dest spyware {
        domainlist     spyware/domains
        urllist        spyware/urls
    }

    dest suspect {
        domainlist     suspect/domains
        urllist        suspect/urls
    }

    dest violence {
        domainlist     violence/domains
        urllist        violence/urls
    }

    dest warez{
        domainlist     warez/domains
        urllist        warez/urls
    }

    src admin {
        ip 192.168.1.0/24
    }

    acl {
        admin within afterwork {
            pass all
        }
        else {
            pass !in-addr !ads !aggressive !audio-video !drugs !gambling !
hacking
!mail !porn !proxy !redirector !spyware !suspect !violence !warez all
        }
        default {
            pass none
            redirect http://192.168.1.10/block.html
        }
    }
}
```



```
}  
}
```

第 9 章

最佳邮件服务器方案

9.1 安装所有相关软件

9.1.1 安装服务器软件

```
$ sudo apt-get install postfix-mysql mysql-server dovecot-pop3d dovecot-imapd  
amav isd-new libclass-dbi-mysql-perl
```

```
New password for the MySQL "root" user: <-- 输入密码  
Repeat password for the MySQL "root" user: <-- 再次输入密码  
Create directories for web-based administration? <-- 选择 No  
General type of mail configuration: <-- 选择 Internet Site  
System mail name: <-- 输入 DNS 全名 (ubox.mytest.com)  
SSL certificate required <-- 选择 ok  
Web server to reconfigure automatically: <-- 选择 apache2
```

9.1.2 安装内容过滤软件

```
$ sudo apt-get install SpamAssassin clamav-daemon razor pyzor cpio arj zoo  
nomarch lzip cabextract pax lha unrar
```

9.1.3 安装其他软件

```
$ sudo apt-get install squirrelmail squirrelmail-locales php5-imap
```

```
$ sudo apt-get install openssl
```

```
$ sudo apt-get install phpmyadmin telnet mutt mailx
```

9.2 为 Postfix 准备数据库

9.2.1 创建数据库 maildb

```
$ mysql -uroot -p
```

```
mysql> create database maildb;
```

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON maildb.* TO  
'mailadmin'@'localhost' IDENTIFIED BY 'mailadminPassword';  
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON maildb.* TO  
'mailadmin'@'localhost.localdomain' IDENTIFIED BY 'mailadminPassword';  
mysql> FLUSH PRIVILEGES;
```

9.2.2 为数据库 maildb 创建数据表

```
mysql> use maildb;
```

1. 创建虚拟域表 virtual_domains

```
mysql> CREATE TABLE `virtual_domains` (  
id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(50) NOT NULL  
) TYPE=MyISAM;
```

```
mysql> INSERT INTO virtual_domains (name)  
VALUES ('mytest.com'),  
('dongyouji.cn');
```

2. 创建虚拟用户表 virtual_users

```
mysql> CREATE TABLE `virtual_users` (  
id int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
domain_id INT(11) NOT NULL,  
user VARCHAR(40) NOT NULL,  
password VARCHAR(32) NOT NULL,  
quota INT(10) DEFAULT '102400',  
CONSTRAINT UNIQUE_EMAIL UNIQUE (domain_id,user),  
FOREIGN KEY (domain_id) REFERENCES virtual_domains(id) ON DELETE CASCADE  
) TYPE=MyISAM;
```

```
mysql> INSERT INTO virtual_users (domain_id, user, password, quota)  
VALUES (1, 'bajie', MD5('bajiePassword'), 10240),
```



```
(1, 'wukong', MD5('wukongPassword'), 102400),  
(2, 'tangseng', MD5('tangsengPassword'), 1048576),  
(1, 'spams', MD5('spamsPassword'), 1024);
```

3. 创建别名表 virtual_aliases

```
mysql> CREATE TABLE `virtual_aliases` (  
id int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
domain_id INT(11) NOT NULL,  
source VARCHAR(40) NOT NULL,  
destination VARCHAR(80) NOT NULL,  
FOREIGN KEY (domain_id) REFERENCES virtual_domains(id) ON DELETE CASCADE  
) TYPE=MyISAM;
```

```
mysql> INSERT INTO virtual_aliases (domain_id, source, destination)
VALUES (2, 'tang seng', 'tang.sanzang@gmail.com'),
(1, 'bajie', 'bajie@mytest.com'),
(1, 'bajie', 'zhu_bajie@yahoo.com'),
(1, '', 'spams@mytest.com');
```

9.2.3 为数据库 maildb 创建视图

```
mysql> SELECT CONCAT(virtual_users.user, '@', virtual_domains.name) AS email,
virtual_users.password
FROM virtual_users
LEFT JOIN virtual_domains ON virtual_users.domain_id=virtual_domains.id;
```

email	password
bajie@mytest.com	00e28675230f4c9b16666098941e5d6d
wukong@mytest.com	407dbf9a4ca5a9906332ff0ea9c330fb
tang seng@dongyouji.cn	3dbc34bca13af30ed7aa2769ee468b40
spams@mytest.com	946bfb68686ed81aa5b0c58bb2633175

1. 创建用户视图 view_users

```
mysql> CREATE VIEW view_users AS
SELECT CONCAT(u.user, '@', virtual_domains.name) AS email, u.password
FROM virtual_users u
LEFT JOIN virtual_domains ON u.domain_id=virtual_domains.id;
```

```
mysql> SELECT * FROM view_users WHERE email LIKE 'bajie%';
```

email	password
bajie@mytest.com	00e28675230f4c9b16666098941e5d6d

2. 创建别名视图 view_virtual_aliases

```
mysql> CREATE VIEW view_aliases AS
SELECT CONCAT(virtual_aliases.source, '@', virtual_domains.name) AS email,
destination
FROM virtual_aliases
LEFT JOIN virtual_domains ON virtual_aliases.domain_id=virtual_domains.id;
```

```
mysql> SELECT * FROM view_aliases;
```

email	password
-------	----------



email	destination	
tangseng@mytest.com	tang.sanzang@gmail.com	
bajie@mytest.com	bajie@mytest.com	
bajie@mytest.com	zhu_bajie@yahoo.com	
@mytest.com	spams@mytest.com	

9.3 配置 Postfix

9.3.1 Postfix 与 MySQL 的关联配置

```
$ sudo mkdir /etc/postfix/mysql/
```

1. 虚拟域 virtual_mailbox_domains 配置

```
$ sudo nano /etc/postfix/mysql/domains.cf
```

```
user = mailadmin
password = mailadminPassword
hosts = 127.0.0.1
dbname = maildb
query = SELECT 1 FROM virtual_domains WHERE name='%s'
```

```
$ sudo postconf -e virtual_mailbox_domains=mysql:/etc/postfix/mysql/domains.cf
```

```
$ postmap -q mytest.com mysql:/etc/postfix/mysql/domains.cf
```

```
1
```

```
postmap: warning: connect to mysql server 127.0.0.1: Access denied for user
'mailadmin'@'localhost' (using password: YES)
```

```
postmap: warning: connect to mysql server 127.0.0.1: Can't connect to MySQL
server on '127.0.0.1'
```

```
bind-address = 127.0.0.1
```

2. 信箱映射 virtual_mailbox_maps 配置

```
$ sudo nano /etc/postfix/mysql/mailbox-maps.cf
```

```
user = mailadmin
```

```
password = mailadminPassword
hosts = 127.0.0.1
dbname = maildb
query = SELECT 1 FROM view_users WHERE email='%s'
```

```
$ postmap -q bajie@mytest.com mysql:/etc/postfix/mysql/mailbox-maps.cf
1
```

```
$ sudo postconf -e virtual_mailbox_maps=mysql:/etc/postfix/mysql/mailbox-
maps.cf
```

```
$ sudo groupadd -g 5000 vmail
$ sudo useradd -g vmail -u 5000 vmail -d /var/mail/virtual -m
```

```
$ sudo postconf -e virtual_uid_maps=static:5000
$ sudo postconf -e virtual_gid_maps=static:5000
```

3. 别名映射 virtual_alias_maps 配置（一）

```
$ sudo nano /etc/postfix/mysql/alias-maps.cf
```

```
user = mailadmin
password = mailadminPassword
hosts = 127.0.0.1
dbname = maildb
query = SELECT destination FROM view_aliases WHERE email='%s'
```

```
$ postmap -q bajie@mytest.com mysql:/etc/postfix/mysql/alias-maps.cf
bajie@mytest.com, zhu_bajie@yahoo.com
```

```
$ sudo postconf -e virtual_alias_maps=mysql:/etc/postfix/mysql/alias-maps.cf
```

4. 别名映射 virtual_alias_maps 配置（二：实验）

```
$ mysql maildb -u root -p
```

```
mysql> DELETE FROM `virtual_aliases`;
```




```
mysql> INSERT INTO virtual_aliases (domain_id, source, destination)
VALUES (1, '', 'hiweedtest@gmail.com');
mysql> exit
```

```
$ mail bajie@mytest.com
```

```
Subject: hi bajie, this is wukong      <-- 输入邮件主题
ni ge bi ma wen!                      <-- 输入正文
.                                      <-- 输入“.”结束正文
Cc:                                    <-- “抄送”地址，可以留空，直接回车
```

```
$ sudo tail /var/log/mail.log
```

```
Dec  7 23:16:36 mail postfix/smtp[8990]: 5304F42BA5: to=<hiweedtest@gmail.com>,
orig_to=<bajie@mytest.com>, relay=gmail-smtp-in.l.google.com[209.85.143.114]:25,
delay=32, delays=0.28/0.09/0.47/31, dsn=2.0.0, status=sent (250 2.0.0 OK
1228709789 i6si27692tid.5)
```

```
Dec  7 23:22:45 mail postfix/smtp[9005]: 9656542BA7: to=<hiweed@126.com>,
orig_to= <bajie@mytest.com>, relay=126.mxmail.netease.com[220.181.15.135]:25,
delay=0.42, delays=0.06/0.01/0.27/0.08, dsn=5.0.0, status=bounced (host
126.mxmail.netease.co m[220.181.15.135] said: 550 MI:SPF
mx5,I8mowLC7KBAUoTxJxeflWQ--.47417S2 1228710164
http://mail.163.com/help/help_spam_16.htm?ip=1020372198&hostid=mx5&time=12287101
64 (in reply to MAIL FROM command))
```

```
$ ls /var/mail/virtual/
```

```
$ sudo postconf -e virtual_alias_maps=
```

```
$ mail bajie@mytest.com
```

```
Subject: hi bajie, test 2              <-- 输入邮件主题
Bajie, ni ge bi ma wen!               <-- 输入正文
Test 2, Hehe.                         <-- 输入“.”结束正文
.                                      <-- 输入“.”结束正文
Cc:                                    <-- “抄送”地址，可以留空，直接回车
```

```
Dec  8 00:19:31 mail postfix/qmgr[8873]: 0AE9542BA7: from=<root@ubox.mytest.
```

```
com>, size=302, nrcpt=1 (queue active)
Dec  8 00:19:32 mail postfix/pipe[9067]: 0AE9542BA7: to=<bajie@mytest.com>,
relay= dovecot, delay=1.6, delays=0.67/0.65/0/0.31, dsn=2.0.0, status=sent
(delivered via dovecot service)
```

```
$ sudo mutt -f /var/mail/virtual/mytest.com/bajie/Maildir
```

5. 别名映射 virtual_alias_maps 配置（三：“自己给自己”）

```
$ sudo nano /etc/postfix/mysql/email2email.cf
```

```
user = mailadmin
password = mailadminPassword
hosts = 127.0.0.1
dbname = maildb
query = SELECT email FROM view_users WHERE email='%s'
```

```
$ postmap -q bajie@mytest.com mysql:/etc/postfix/mysql/email2email.cf
bajie@mytest.com
```

```
$ sudo postconf -e virtual_alias_maps=mysql:/etc/postfix/mysql/alias-maps.cf,
mysql:/etc/postfix/mysql/email2email.cf
```

6. 修改配置文件权限

```
$ sudo chgrp postfix /etc/postfix/mysql/*.
$ sudo chmod u=rw,g=r,o= /etc/postfix/mysql/*.
```

9.3.2 让 Postfix 使用 Dovecot 分发邮件

```
$ sudo nano /etc/postfix/master.cf
```

```
dovecot unix - n n - - pipe
flags=DRhu user=vmail:vmail argv=/usr/lib/dovecot/deliver -d ${recipient}
```

```
$ sudo postfix reload
```

```
$ sudo postconf -e virtual_transport=dovecot
```



```
$ sudo postconf -e dovecot_destination_recipient_limit=1
```

9.4 配置 Dovecot

9.4.1 配置 dovecot.conf

```
$ sudo cp /etc/dovecot/dovecot.conf /etc/dovecot/dovecot.conf-orig
$ sudo nano /etc/dovecot/dovecot.conf
```

1. 全局部分配置

```
protocols = imap imaps pop3 pop3s
```

```
mail_location = maildir:/var/mail/virtual/%d/%n/Maildir
```

```
ssl_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
ssl_disable = no
disable_plaintext_auth = no
```

2. auth default 部分配置

```
mechanisms = plain login
```

```
passdb sql {
    [...]
    args = /etc/dovecot/dovecot-sql.conf
    [...]
}
```

```
userdb static {
    [...]
    args = uid=5000 gid=5000 home=/var/mail/virtual/%d/%n allow_all_users=
yes
    [...]
}
```

```
socket listen {
    master {
        path = /var/run/dovecot/auth-master
        mode = 0600
        user = vmail
    }

    client {
        path = /var/spool/postfix/private/auth
        mode = 0660
        user = postfix
        group = postfix
    }
}
```

```
}
```

3. protocol lda 部分配置

```
protocol lda {  
    [...]  
    log_path = /var/mail/virtual/dovecot-deliver.log  
    auth_socket_path = /var/run/dovecot/auth-master  
    postmaster_address = postmaster@mytest.com  
    [...]  
}
```

至此，`/etc/dovecot/dovecot.conf` 就修改完了。

9.4.2 配置 dovecot-sql.conf

```
$ sudo nano /etc/dovecot/dovecot-sql.conf
```

```
driver = mysql  
connect      =      host=127.0.0.1      dbname=maildb      user=mailadmin      password=  
mailadminPassword  
default_pass_scheme = PLAIN-MD5  
password_query = SELECT email as user, password FROM view_users WHERE email='%u';
```

9.4.3 修改配置文件权限

```
$ sudo chgrp vmail /etc/dovecot/dovecot.conf  
$ sudo chmod g+r /etc/dovecot/dovecot.conf
```

9.4.4 重新启动 Dovecot

```
$ sudo /etc/init.d/dovecot restart
```

```
dovecot: Dovecot v1.0.rc15 starting up  
dovecot: auth-worker(default): mysql: Connected to 127.0.0.1 (maildb)
```

9.5 用 Telnet 进行 SMTP/POP3/IMAP 测试

9.5.1 SMTP 测试

1. 用 Telnet 通过 SMTP 发送一封邮件

```
$ telnet localhost smtp
```

```
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^['.
```



```
220 ubox.mytest.com ESMTP Postfix (Ubuntu)
```

```
ehlo mytest.com
```

```
250-ubox.mytest.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

```
mail from:<tangseng@mytest.com>
```

```
250 2.1.0 Ok
```

```
rcpt to:<bajie@mytest.com>
```

```
250 2.1.5 Ok
```

```
data
```

```
354 End data with <CR><LF>.<CR><LF>
```

```
Subject: Hi Bajie, ni hai huo zhe ma?
```

```
Hi BiMaWen,
Wo shi ni HouGe ya!
Wo chao gu pei le, ni zen me yang a?
Bu luo suo le, hui tou jian!
HouGe
.
```

```
250 2.0.0 Ok: queued as 9F7F642BA9
```

```
quit
```

2. 检查 Postfix 日志

```
$ sudo tail /var/log/mail.log
```

```
Dec 8 09:15:41 mail postfix/qmgr[9260]: 9F7F642BA9: from=<tangseng@mytest.com>,
size=486, nrcpt=1 (queue active)
Dec 8 09:15:42 mail postfix/pipe[18711]: 9F7F642BA9: to=<bajie@mytest.com>,
relay =dovecot, delay=20, delays=19/0.04/0/0.26, dsn=2.0.0, status=sent
(delivered via dovecot service)
```

3. 检查用户的 Maildir

```
$ sudo find /var/mail/virtual/mytest.com/bajie/Maildir
```

```
/var/mail/virtual/mytest.com/bajie/Maildir
/var/mail/virtual/mytest.com/bajie/Maildir/new
/var/mail/virtual/mytest.com/bajie/Maildir/new/1228743306.P18619Q0M520619.ubox
/var/mail/virtual/mytest.com/bajie/Maildir/dovecot.index
/var/mail/virtual/mytest.com/bajie/Maildir/dovecot-uidlist
/var/mail/virtual/mytest.com/bajie/Maildir/tmp
/var/mail/virtual/mytest.com/bajie/Maildir/cur
/var/mail/virtual/mytest.com/bajie/Maildir/dovecot.index.log
```

```
$ sudo mutt -f /var/mail/virtual/mytest.com/bajie/Maildir
```

9.5.2 测试 POP3

```
$ telnet localhost pop3
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK Dovecot ready.
```

```
user bajie@mytest.com
```

```
+OK
```



```
pass bajiePassword
```

```
+OK Logged in.
```

```
List
```

```
+OK 2 messages:
```

```
1 371  
2 503  
.
```

```
retr 2
```

```
+OK 503 octets  
Return-Path: <tangseng@mytest.com>  
Delivered-To: bajie@mytest.com  
Received: from mytest.com (localhost [127.0.0.1])  
    by ubox.mytest.com (Postfix) with ESMTP id 9F7F642BA9  
    for <bajie@mytest.com>; Mon, 8 Dec 2008 09:15:22 -0500 (EST)  
Subject: Hi Bajie, ni hai huo zhe ma?  
Message-Id: <20081208141526.9F7F642BA9@ubox.mytest.com>  
Date: Mon, 8 Dec 2008 09:15:22 -0500 (EST)  
From: tangseng@mytest.com  
To: undisclosed-recipients;;
```

```
Hi BiMaWen,
```

```
Wo shi ni HouGe ya!
```

```
Wo chao gu pei le, ni zen me yang a?
```

```
Bu luo suo le, hui tou jian!
```

```
HouGe  
.
```

```
quit
```

```
+OK Logging out.
```

```
Connection closed by foreign host.
```

9.5.3 测试 IMAP

```
$ telnet localhost imap2
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK Dovecot ready.
```

```
1 login bajie@mytest.com bajiePassword
```

```
1 OK Logged in.
```

```
2 list "" ""
```

```
* LIST (\HasChildren) "." "INBOX"
2 OK List completed.
```

```
3 select "INBOX"
```

```
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags
permitted.
* 2 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1228713572] UIDs valid
* OK [UIDNEXT 4] Predicted next UID
3 OK [READ-WRITE] Select completed.
```

```
4 fetch 2 all
```

于是, Dovecot 一边收信, 一边将大概信息告诉你:

```
* 3 FETCH (FLAGS (\Seen) INTERNALDATE "08-Dec-2008 09:15:42 -0500" RFC822.SIZE
554 ENVELOPE ("Mon, 8 Dec 2008 09:15:22 -0500 (EST)" "Hi Bajie, ni hai huo zhe
ma?" ((N IL NIL "tang seng" "mytest.com")) ((NIL NIL "tang seng" "mytest.com"))
((NIL NIL "ta ngseng" "mytest.com")) ((NIL NIL "undisclosed-recipients" NIL) (NIL
NIL "" "MISSING_ DOMAIN") (NIL NIL NIL NIL) NIL NIL NIL
"<20081208141526.9F7F642BA9@ubox.mytest.co m>"))
4 OK Fetch completed.
```




```
5 fetch 2 body[]
```

```
* 2 FETCH (BODY[] {554})
Return-Path: <tangseng@mytest.com>
Delivered-To: bajie@mytest.com
Received: from mytest.com (localhost [127.0.0.1])
    by ubox.mytest.com (Postfix) with ESMTP id 9F7F642BA9
    for <bajie@mytest.com>; Mon, 8 Dec 2008 09:15:22 -0500 (EST)
Subject: Hi Bajie, ni hai huo zhe ma?
Message-Id: <20081208141526.9F7F642BA9@ubox.mytest.com>
Date: Mon, 8 Dec 2008 09:15:22 -0500 (EST)
From: tangseng@mytest.com
To: undisclosed-recipients:;

Hi BiMaWen,

Wo shi ni HouGe ya!
Wo chao gu pei le, ni zen me yang a?

Bu luo suo le, hui tou jian!

HouGe
)
5 OK Fetch completed.
```

```
6 logout
```

```
* BYE Logging out
6 OK Logout completed.
Connection closed by foreign host.
```

9.6 用 Thunderbird 进行 SMTP/POP3/IMAP 测试

9.6.2 修改 hosts 文件

```
192.168.1.10    ubox.mytest.com
```

9.7 实现 SMTP 认证

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

9.7.1 配置 Postfix

```
$ sudo postconf -e smtpd_sasl_type=dovecot
$ sudo postconf -e smtpd_sasl_path=private/auth
$ sudo postconf -e smtpd_sasl_auth_enable=yes
$ sudo postconf -e smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_
aut henticated,reject_unauth_destination
```

9.7.2 用 Telnet 测试 SMTP 认证

```
$ perl -MMIME::Base64 -e \
    'print encode_base64("bajie@mytest.com\0bajie@mytest.com\0password")';
```

```
$ sudo telnet localhost smtp
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 ubox.mytest.com ESMTP Postfix (Ubuntu)
```

```
ehlo mytest.com
```

```
250-ubox.mytest.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

```
auth plain YmFqaWVAbX10ZXN0LmNvbQBjYWppZUBteXRlc3QuY29tAGJhamllUGFz3dvcmQ=
```

```
235 2.7.0 Authentication successful
```

```
quit
```



```
221 2.0.0 Bye
```

9.7.3 用 Thunderbird 测试 SMTP 认证

```
$ sudo tail -f /var/log/mail.log
```

```
Dec 8 21:06:10 mail postfix/smtpd[19476]: 4FDA342BA9: client=unknown[192.168.1.11 9], sasl_method=PLAIN, sasl_username=bajie@mytest.com
...
Dec 8 21:06:10 mail postfix/qmgr[19473]: 4FDA342BA9: from=<bajie@mytest.com>, siz e=499, nrcpt=1 (queue active)
...
Dec 8 21:06:42 mail postfix/smtp[19489]: 4FDA342BA9: to=<hiweed@gmail.com>, relay =gmail-smtp-in.l.google.com[209.85.143.27]:25, delay=32, delays=0.12/0.18/0.45/31, dsn=5.7.1, status=bounced ...
```

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
```

9.8 强迫用户使用 TLS 加密连接 SMTP

```
$ sudo postconf -e smtpd_tls_security_level=encrypt
```

9.9 使用自己创建的安全证书

```
$ sudo openssl req -new -x509 -days 3650 -nodes -out /etc/ssl/certs/mytest.com.pem -keyout /etc/ssl/private/mytest.com.key
```

```
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Shandong
Locality Name (eg, city) []:Qingdao
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Hiweed Linux Ltd
Organizational Unit Name (eg, section) []:Devel Dept.
Common Name (eg, YOUR name) []:ubox.mytest.com <-- 重要! 这里输入DNS全名
Email Address []:postmaster@mytest.com
```

```
$ sudo postconf -e smtpd_tls_cert_file=/etc/ssl/certs/mytest.com.pem
$ sudo postconf -e smtpd_tls_key_file=/etc/ssl/private/mytest.com.key
```

```
$ sudo nano /etc/dovecot/dovecot.conf
```

```
ssl_cert_file = /etc/ssl/certs/mytest.com.pem  
ssl_key_file = /etc/ssl/private/mytest.com.key
```

```
$ sudo /etc/init.d/dovecot restart
```

9.10 利用 Dovecot 实现 Quota（磁盘限额）

9.10.1 启用 quota 插件

```
$ sudo nano /etc/dovecot/dovecot.conf
```

```
protocol imap {  
    [...]  
    mail_plugins = quota imap_quota  
    mail_plugin_dir = /usr/lib/dovecot/modules/imap  
    [...]  
}  
protocol pop3 {  
    [...]  
    mail_plugins = quota  
    mail_plugin_dir = /usr/lib/dovecot/modules/pop3  
    [...]  
}  
protocol lda {  
    [...]  
    mail_plugins = quota  
    mail_plugin_dir = /usr/lib/dovecot/modules/lda  
    [...]  
}
```



9.10.2 配置 quota

1. 全局 quota 配置

```
$ sudo nano /etc/dovecot/dovecot.conf
```

```
plugin {  
  [...]  
  quota = maildir:storage=102400:messages=1000  
  [...]  
}
```

```
quota = maildir:storage=1048576
```

2. 个别用户的 quota 配置

```
$ sudo nano /etc/dovecot/dovecot.conf
```

```
[...]  
userdb sql {  
  args = /etc/dovecot/dovecot-sql.conf  
}  
  
userdb static {  
  [...]
```

```
$ sudo nano /etc/dovecot/dovecot-sql.conf
```

```
user_query = SELECT  '/var/mail/virtual/%d/%n/Maildir' AS home, 'vmail' AS uid,  
'v mail' AS gid, concat('maildir:storage=', quota) AS quota FROM users WHERE  
user = '%n'
```

```
$ sudo /etc/init.d/dovecot restart
```

9.11 垃圾邮件、病毒过滤

9.11.1 配置 SpamAssassin

```
$ sudo nano /etc/spamassassin/local.cf
```

```
bayes_auto_expire 0
```

```
use_pyzor 1
pyzor_path /usr/bin/pyzor

use_razor2 1
razor_config /etc/razor/razor-agent.conf

use_bayes 1
use_bayes_rules 1
bayes_auto_learn 1
```

```
$ spamassassin --lint
```

```
$ sudo sa-update --no-gpg
```

9.11.2 配置 AMaViSd

1. 启用内容过滤

```
$ sudo nano /etc/amavis/conf.d/15-content_filter_mode
```

```
use strict;
@bypass_virus_checks_maps = (
    \%bypass_virus_checks, \@bypass_virus_checks_acl, \$bypass_virus_checks_re);
@bypass_spam_checks_maps = (
    \%bypass_spam_checks, \@bypass_spam_checks_acl, \$bypass_spam_checks_re);
1; # ensure a defined return
```

2. 修改 Debian 默认配置

```
$ sudo nano /etc/amavis/conf.d/20-debian_defaults
```

```
[...]
$final_spam_destiny      = D_PASS;
[...]
```



```
$sa_spam_subject_tag = '***SPAM*** ';
```

```
$sa_tag_level_deflt = 2.0;
```

```
$sa_tag2_level_deflt = 6.31;
```

```
$sa_kill_level_deflt = 6.31;
```

```
$banned_filename_re = new_RE(
[... ]
qr'\.[^./]*\.(exe|vbs|pif|scr|bat|cmd|com|cpl|dll)\.?$'i,
[... ]
qr'\.(exe|vbs|pif|scr|bat|cmd|com|cpl)$'i, # banned extension - basic
[... ]
qr'^\.(exe-ms)$', # banned file(1) types
);
```

3. QUARANTINEDIR 设置

```
#$QUARANTINEDIR = "$MYHOME/virusmails";
$QUARANTINEDIR = undef ;
```

```
$ sudo nano /etc/cron.daily/clean-amavis
#!/bin/bash

if [ -d /var/lib/amavis/virusmails/ ]; then
    find /var/lib/amavis/virusmails/ -mtime +7 | xargs rm -rf
fi

exit 0
```

```
$ sudo chmod +x /etc/cron.daily/clean-amavis
```

4. 域名搜索配置

```
$ sudo nano /etc/amavis/conf.d/50-user
```

```
@lookup_sql_dsn = (
['DBI:mysql:database=maildb;host=127.0.0.1;port=3306',
'mailadmin',
'mailadminPassword']);
```

```
$sql_select_policy = 'SELECT name FROM virtual_domains WHERE CONCAT("@",name) IN (%k)';
```

```
$ sudo chmod o= /etc/amavis/conf.d/50-user
```

```
$ sudo /etc/init.d/amavis restart
```

5. 使 AMaViSd 能和 ClamAV 通话

```
$ sudo adduser clamav amavis
```

```
$ sudo /etc/init.d/clamav-daemon restart
$ sudo /etc/init.d/clamav-freshclam restart
```

9.11.3 配置 Postfix，将邮件交给 AMaViSd 过滤

1. 为 Postfix 创建 smtp-amavis 服务

```
$ sudo netstat -nap | grep 10024
```

```
tcp        0      0 127.0.0.1:10024      0.0.0.0:*             LISTEN      6014/amavisd
(maste
```

```
$ sudo nano /etc/postfix/master.cf
```

```
smtp-amavis unix -      -      n      -      2      smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
-o max_use=20
```

```
$ sudo postconf -e content_filter=smtp-amavis:[127.0.0.1]:10024
```

2. 为 Postfix 创建 10025 服务

```
$ sudo nano /etc/postfix/master.cf
```

```
127.0.0.1:10025 inet n      -      -      -      smtpd
-o content_filter=
-o local_recipient_maps=
```




```
-o relay_recipient_maps=  
-o smtpd_restriction_classes=  
-o smtpd_delay_reject=no  
-o smtpd_tls_security_level=  
-o smtpd_client_restrictions=permit_mynetworks,reject  
-o smtpd_helo_restrictions=  
-o smtpd_sender_restrictions=  
-o smtpd_recipient_restrictions=permit_mynetworks,reject  
-o smtpd_data_restrictions=reject_unauth_pipelining  
-o smtpd_end_of_data_restrictions=  
-o mynetworks=127.0.0.0/8  
-o smtpd_error_sleep_time=0  
-o smtpd_soft_error_limit=1001  
-o smtpd_hard_error_limit=1000  
-o smtpd_client_connection_count_limit=0  
-o smtpd_client_connection_rate_limit=0  
-o receive_override_options=no_header_body_checks,no_unknown_recipient_checks  
-o local_header_rewrite_clients=
```

```
$ sudo postconf -e receive_override_options=no_address_mappings
```

```
$ sudo /etc/init.d/postfix restart
```

```
$ sudo netstat -tap
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	localhost:10024	*:*	LISTEN	26386/amavisd (mast
tcp	0	0	localhost:10025	*:*	LISTEN	26308/master

[...]

9.11.4 垃圾邮件测试

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

```
$ sendmail bajie@mytest.com < /usr/share/doc/SpamAssassin/examples/sample-spam.  
Tx t
```

```
X-Virus-Scanned: Debian amavisd-new at ubox.mytest.com  
X-Spam-Flag: NO  
X-Spam-Score: 3.229  
X-Spam-Level: ***  
X-Spam-Status: No, score=3.229 tagged_above=0 required=6.31 tests=[ALL_TRUSTED=-  
1.44, DATE_IN_PAST_12_24=1.77, TVD_SPACE_RATIO=2.899]
```

```
X-Virus-Scanned: Debian amavisd-new at ubox.mytest.com  
X-Spam-Flag: YES  
X-Spam-Score: 1003.029  
X-Spam-Level: *****  
X-Spam-Status: Yes, score=1003.029 tagged_above=0 required=6.31 tests=[ALL_TRUST  
ED=-1.44, AWL=0.200, DATE_IN_PAST_12_24=1.77, GTUBE=1000, RAZOR2_CF_RANGE_51_100=0.  
5, RAZOR2_CF_RANGE_E4_51_100=1.5, RAZOR2_CHECK=0.5]
```

9.11.5 非法附件测试

```
BANNED message from you (multipart/mixed | application/x- msdownload,  
.exe,exe,exe) (exe)
```

```
BANNED message from you (multipart/mixed | application/octet-  
stream,.zip,filename.zip | .empty,filename.BAT)
```

9.11.6 将 Spam 自动转存到“垃圾”文件夹

1. 在服务器端实现 Spam 自动转存

```
$ sudo nano /etc/dovecot/dovecot.conf
```

```
protocol lda {  
    [...]  
    mail_plugins = quota cmusieve  
    global_script_path = /var/mail/virtual/spam-move.sieve  
    [...]  
}
```

```
$ sudo nano /var/mail/virtual/spam-move.sieve
```



```
require ["fileinto"];
if header :contains "X-Spam-Flag" ["YES"] {
    fileinto "Junkmail";
    stop;
}
```

```
$ sudo /etc/init.d/dovecot restart
```

3. Spam 自动转存测试

```
$ sendmail bajie@mytest.com < /usr/share/doc/spamassassin/examples/sample-
spam.tx t
```

```
deliver(bajie@mytest.com): 2008-12-11 14:17:59 Info: msgid=<GTUBE1.
1010101@exampl e.net>: saved mail to Junkmail
```

9.12 Webmail 的实现

9.12.1 配置 SquirrelMail

```
$ sudo squirrelmail-configure
```

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

```
-----  
Main Menu --
```

1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

```
Command >>
```

```
SquirrelMail Configuration : Read: config.php (1.4.0)
```

```
-----  
Language preferences
```

1. Default Language : **zh_CN**
2. Default Charset : **gb2312**
3. Enable lossy encoding : false

R Return to Main Menu
C Turn color on
S Save data
Q Quit

```
$ sudo nano /var/lib/locales/supported.d/local
```

```
zh_CN GB2312
```

```
$ sudo dpkg-reconfigure locales
```

```
$ sudo ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.
```



```
conf
```

```
$ sudo /etc/init.d/apache restart
```

9.13 修改系统别名/etc/aliases

```
login: hiweed
password:
Linux mail 2.6.24-22-server #1 SMP Mon Nov 24 19:14:19 UTC 2008 i686
[...]
No mail. (或者: You have new mail.)
hiweed@ubox:~$
```

```
$ mail
```

```
Mail version 8.1.2 01/15/2001. Type ? for help.
"/var/mail/hiweed": 2 messages 2 new
>N 1 hiweed@mail.mytes Tue Dec 9 05:13 14/459 hi hiweed
N 2 hiweed@mail.mytes Tue Dec 9 05:13 14/453 a? ha?
&
```

```
$ sudo nano /etc/aliases
```

```
[...]
Postmaster: root
root: hiweed@mytest.com
[...]
```

```
$ sudo newaliases
```

```
$ sudo /etc/init.d/postfix restart
```

9.14 Web 管理工具

9.14.1 安装 Virtual Mail Manager

```
$ cd ~
$ wget http://www.grs-service.ch/pub/grs_mailmgr_v1_6.tgz
$ sudo mkdir /var/www/mailadmin/
$ cd /var/www/mailadmin/
$ sudo tar xfvz /home/hiweed/grs_mailmgr_v1_6.tgz
```

```
$ sudo cp conf/cnf_main_template.php conf/cnf_main.php
$ sudo nano conf/cnf_main.php
```



```
$grs_db_name      = 'maildb'; // database name
$grs_db_host       = 'localhost'; // database server
$grs_db_username   = 'mailadmin'; // database user
$grs_db_password   = 'mailadminPassword'; // database password
```

```
$ mysql maildb -u root -p
```

```
mysql> CREATE TABLE `domain_admins` (
  `id` int(11) NOT NULL auto_increment,
  `domain_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
mysql> INSERT INTO `maildb`.`domain_admins` (
  `id` ,
  `domain_id` ,
  `user_id`
)
VALUES (
  NULL , '1', '1'
);
```

```
mysql> CREATE TABLE `languages` (
  `id` int(3) NOT NULL auto_increment,
  `name` varchar(50) NOT NULL,
  `active` tinyint(1) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

mysql> INSERT INTO `languages` (`id`, `name`, `active`) VALUES
(1, 'English', 1),
(2, 'Deutsch', 1);

mysql> exit
```

```
$ wget http://www.grs-service.ch/pub/scripts/sql_3.txt
$ mysql maildb < sql_3.txt -u root -p
```


第 10 章

最佳邮件列表：Mailman

10.1 安装 Mailman

```
$ sudo apt-get install apache2 postfix mailman
```

```
* Site list for mailman (usually named mailman) missing.  
* Please create it; until then, mailman will refuse to start.
```

10.2 配置 Mailman

10.2.1 修改主机名

```
$ sudo nano /etc/hostname
```

```
lists
```

```
$ sudo /etc/init.d/hostname.sh
```

```
192.168.1.10    lists.mytest.com
```

10.2.2 配置 Apache

```
$ sudo ln -s /etc/mailman/apache.conf /etc/apache2/sites-enabled/mailman
```

```
$ ls -l /etc/apache2/sites-enabled/mailman
```

```
lrwxrwxrwx 1 root root 24 2009-03-22 22:28 /etc/apache2/sites-enabled/mailman ->  
/ etc/mailman/apache.conf
```

```
$ sudo /etc/init.d/apache2 restart
```

10.2.3 配置 Postfix

1. 配置 main.cf

```
$ sudo nano /etc/postfix/main.cf
```

```
myhostname = lists
[...]
mydestination = lists, localhost.localdomain, , localhost
```

```
relay_domains = lists.mytest.com
transport_maps = hash:/etc/postfix/transport
mailman_destination_recipient_limit = 1
```

2. 配置 master.cf

```
mailman unix - n n - - pipe
 flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
 ${nexthop} ${user}
```

3. 配置 transport

```
$ sudo nano /etc/postfix/transport
```

```
lists.mytest.com mailman:
```

```
$ sudo postmap -v /etc/postfix/transport
```

```
[...]
postmap: dict_eval: const lists
postmap: dict_eval: const localdomain
postmap: dict_eval: const Postfix
[...]
postmap: dict_eval: const lists, localhost.localdomain, , localhost
postmap: dict_eval: expand $myhostname -> lists
[...]
postmap: dict_eval: const lists.mytest.com
postmap: dict_eval: expand $relay_domains -> lists.mytest.com
[...]
postmap: inet_addr_local: configured 2 IPv4 addresses
postmap: open hash /etc/postfix/transport
postmap: Compiled against Berkeley DB: 4.6.21?
postmap: Run-time linked against Berkeley DB: 4.6.21?
```

```
$ sudo /etc/init.d/postfix restart
```



10.2.4 创建默认邮件列表

```
$ sudo newlist mailman
```

Enter the email of the person running the list:hiweed@hiweed.com<--输入你的 E-mail 地址

Initial mailman password: <-- 输入密码

To finish creating your mailing list, you must edit your /etc/aliases (or equivalent) file by adding the following lines, and possibly running the 'newaliases' program:

```
## mailman mailing list
mailman:                "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin:          "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces:        "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm:        "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join:           "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave:          "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:          "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request:        "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe:      "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe:    "|/var/lib/mailman/mail/mailman unsubscribe mailman"
```

Hit enter to notify mailman owner...

```
$ sudo nano /etc/mailman/mm_cfg.py
```

```
#-----
# Default domain for email addresses of newly created MLs
DEFAULT_EMAIL_HOST = 'lists.mytest.com'
#-----
# Default host for web interface of newly created MLs
DEFAULT_URL_HOST   = 'lists.mytest.com'
#-----
```

```
$ sudo /etc/init.d/mailman start
```

10.3 管理 Mailman

10.3.2 通过命令行管理 Mailman

1. 创建邮件列表

```
$ sudo newlist listname1
```

2. 显示所有邮件列表

```
$ sudo list_lists
```

```
2 matching mailing lists found:
  Listname1 - [no description available]
  Mailman - [no description available]
```

3. 删除邮件列表

```
$ sudo rmlist -a listname1
```

To finish removing your mailing list, you must edit your `/etc/aliases` (or equivalent) file by removing the following lines, and possibly running the `'newaliases'` program:

```
## listname1 mailing list
listname1:                "|/var/lib/mailman/mail/mailman post listname1"
listname1-admin:          "|/var/lib/mailman/mail/mailman admin listname1"
listname1-bounces:        "|/var/lib/mailman/mail/mailman bounces listname1"
listname1-confirm:        "|/var/lib/mailman/mail/mailman confirm listname1"
listname1-join:           "|/var/lib/mailman/mail/mailman join listname1"
listname1-leave:          "|/var/lib/mailman/mail/mailman leave listname1"
listname1-owner:          "|/var/lib/mailman/mail/mailman owner listname1"
listname1-request:        "|/var/lib/mailman/mail/mailman request listname1"
listname1-subscribe:       "|/var/lib/mailman/mail/mailman subscribe listname1"
listname1-unsubscribe:    "|/var/lib/mailman/mail/mailman unsubscribe listname1"

Removing list info
Removing private archives
Removing private archives
Removing public archives
listname1 public archives not found as /var/lib/mailman/archives/public/
listname1.mbox
```

4. 添加成员

```
$ cat mylist.txt
```

```
hiweedleng@163.com
hiweedleng@126.com
kanakaleng@yeah.net
```

```
$ sudo add_members --regular-members-file=mylist.txt --welcome-msg=y mailman
```

```
已订阅: hiweedleng@163.com
已订阅: hiweedleng@126.com
已订阅: kanakaleng@yeah.net
```

```
$ tail /var/log/mail.log
```

5. 显示成员

```
$ sudo list_members mailman
```

```
hiweedleng@163.com
hiweedleng@126.com
kanakaleng@yeah.net
```



6. 克隆成员

```
$ sudo clone_member --remove hiweedtest@163.com hiweedNew@163.com
```

```
processing mailing list: mailman  
clone address added: hiweedNew@163.com  
original address removed: hiweedtest@163.com
```

7. 搜索成员

```
$ sudo find_member hiweedtest@163.com
```

```
hiweed@163.com found in:  
mailman
```

8. 删除成员

```
$ sudo remove_members --file=remove.txt listname1
```

```
$ sudo remove_members --fromall hiweedtest@163.com
```

9. 同步成员

```
$ sudo sync_members -f mylist.txt mailman
```

```
Added : hiweedtest@163.com  
Removed: hiweedNew@163.com
```

第 11 章

最佳 FTP 服务器方案

11.3 Pure-FTPd 的安装、配置

11.3.1 安装 Pure-FTPd

```
$ sudo apt-get install pure-ftpd-mysql mysql-server
```

11.3.2 配置 Pure-FTPd

1. 添加用户和组

```
$ sudo groupadd -g 2001 ftpgroup
$ sudo useradd -u 2001 -s /bin/false -d /dev/null -c "Pure-FTPd User" -g ftpgroup
ftpuser
```

2. Chroot 设置

```
$ sudo sh -c "echo 'yes' > /etc/pure-ftpd/conf/ChrootEveryone"
```

3. 手工创建用户目录

```
$ sudo sh -c "echo 'No' > /etc/pure-ftpd/conf/CreateHomeDir"
```

4. 为 Pure-FTPd 创建 MySQL 数据库

```
$ mysql -u root -p
```

```
mysql> CREATE DATABASE ftpusers;
```

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON ftpusers.* TO
'ftpadm in'@'localhost' IDENTIFIED BY 'ftpadminPassword';
```

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON ftpusers.* TO
'ftpadm in'@'localhost.localdomain' IDENTIFIED BY 'ftpadminPassword';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> USE ftpusers;

mysql> CREATE TABLE IF NOT EXISTS `users` (
  `User` varchar(16) NOT NULL default '',
  `Password` varchar(32) NOT NULL default '',
  `Uid` int(11) NOT NULL,
  `Gid` int(11) NOT NULL,
  `Dir` varchar(128) NOT NULL default '',
  `QuotaFiles` int(10) NOT NULL default '500',
  `QuotaSize` int(10) NOT NULL default '30',
  `ULBandwidth` int(10) NOT NULL default '80',
  `DLBandwidth` int(10) NOT NULL default '80',
  `Ipaddress` varchar(15) NOT NULL default '*',
  `Comment` tinytext,
  `Status` enum('0','1') NOT NULL default '1',
  `ULRatio` smallint(5) NOT NULL default '1',
  `DLRatio` smallint(5) NOT NULL default '1',
  PRIMARY KEY (`User`),
  UNIQUE KEY `User` (`User`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

mysql> quit
```

5. 配置 Pure-FTPd 的 mysql.conf

```
$ sudo mv /etc/pure-ftpd/db/mysql.conf /etc/pure-ftpd/db/mysql.conf_orig
```

```
$ sudo nano /etc/pure-ftpd/db/mysql.conf
```

```
MYSQLServer      127.0.0.1
MYSQLSocket      /var/run/mysqld/mysqld.sock
MYSQLUser        ftpadmin
MYSQLPassword    ftpadminPassword
MYSQLDatabase    ftpusers
MYSQLCrypt       md5
MYSQLGetPW       SELECT Password FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetUID      SELECT Uid FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetGID      SELECT Gid FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetDir      SELECT Dir FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetQTAFS    SELECT QuotaFiles FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetQTASZ    SELECT QuotaSize FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetRatioUL  SELECT ULRatio FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
MYSQLGetRatioDL  SELECT DLRatio FROM users WHERE User="\L" AND Status="1" AND
(Ipaddress = "*" OR Ipaddress LIKE "\R")
```



```
MySQLGetBandwidthUL SELECT ULBandwidth FROM users WHERE User="\L" AND Status="1"
AND (Ippaddress = "*" OR Ippaddress LIKE "\R")
MySQLGetBandwidthDL SELECT DLBandwidth FROM users WHERE User="\L" AND Status="1"
AND (Ippaddress = "*" OR Ippaddress LIKE "\R")
```

```
$ sudo chmod g=o= /etc/pure-ftpd/db/mysql.conf
```

```
$ sudo /etc/init.d/pure-ftpd-mysql restart
```

11.4 实现 FTP 用户的 Web 管理

11.4.1 安装 User manager for PureFTPd

```
$ cd ~
$ wget http://machiell.generaal.net/files/pureftpd/ftp_v2.1.tar.gz
$ cd /var/www
$ sudo tar xfvz ~/ftp_v2.1.tar.gz
```

11.4.2 配置 User manager for PureFTPd

```
$ sudo nano /var/www/ftp/config.php
```

```
[...]
$LANG = "Chinese";
$LocationImages = "images";
$DBHost = "127.0.0.1";
$DBLogin = "ftpadmin";
$DBPassword = "ftpadminPassword";
$DBDatabase = "ftpusers";
$FTPAddress = "ubox.mytest.com:21";
$DEFUserID = "2001";
$DEFGroupID = "2001";
$UsersFile = "/etc/passwd";
$GroupFile = "/etc/group";
$StyleSheet = "style/default.css.php";
$EnableQuota = 1;
$EnableRatio = 1;
[...]
```

11.4.3 设置 User manager for PureFTPd 管理员

```
$ mysql -u root -p
```



```
mysql> USE ftpusers;

mysql> CREATE TABLE IF NOT EXISTS `admin` (
  `Username` varchar(35) NOT NULL default '',
  `Password` char(32) NOT NULL default '',
  PRIMARY KEY (`Username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
INSERT INTO `admin` (`Username`, `Password`) VALUES
('ftpadmin', MD5('ftpadminPassword'));

mysql> quit
```

11.5 Pure-FTPd 配置选项介绍

11.5.3 字符串型配置选项

1. 任意字符串

```
ftp.ubox.org - jedi [13/Dec/2009:19:36:39] "GET /ftp/linux.tar.bz2" 200 21809338
```

11.6 实现 TLS 认证

11.6.1 证书设置

```
$ sudo openssl req -x509 -nodes -newkey rsa:1024 -keyout /etc/ssl/private/pure-
ftpd.pem -out /etc/ssl/private/pure-ftpd.pem
```



11.6.2 服务器的 TLS 设置

```
$ sudo sh -c "echo '2' > /etc/pure-ftpd/conf/TLS"
```

```
$ sudo /etc/init.d/pure-ftpd-mysql restart
```

11.7 FXP 协议支持

```
$ sudo sh -c "echo 'Yes' > /etc/pure-ftpd/conf/AllowUserFXP"  
$ sudo /etc/init.d/pure-ftpd-mysql restart
```

11.8 允许匿名访问

11.8.1 Pure-FTPd 设置

```
$ sudo sh -c "echo 'No' > /etc/pure-ftpd/conf/NoAnonymous"
```

```
$ sudo sh -c "echo 'Yes' > /etc/pure-ftpd/conf/AnonymousCantUpload"
```

```
$ sudo sh -c "echo 'Yes' > /etc/pure-ftpd/conf/AnonymousCanCreateDirs"
```

```
$ sudo sh -c "echo 'Yes' > /etc/pure-ftpd/conf/AnonymousOnly"
```

```
$ sudo /etc/init.d/pure-ftpd-mysql restart
```

11.8.2 添加系统用户

```
$ sudo groupadd ftp  
$ sudo useradd ftp -s /bin/false -d /var/ftp -m -c "anonymous ftp" -g ftp
```

第 12 章

最佳 NFS 服务器方案

12.2 NFS 服务器的安装及配置

12.2.2 安装 NFS 服务器软件

```
$ sudo apt-get install nfs-kernel-server nfs-common portmap nis
```

12.2.3 Portmap 安全

```
$ sudo nano /etc/hosts.deny
```

```
portmap mountd nfsd statd lockd rquotad : ALL
```

```
$ sudo nano /etc/hosts.allow
```

```
portmap mountd nfsd statd lockd rquotad : 192.168.1.10 192.168.1.100  
portmap ypserv ypbind : 192.168.1.10 192.168.1.100
```

```
$ sudo /etc/init.d/portmap restart
```

12.2.4 NIS 服务器配置

```
$ sudo nano /etc/default/nis
```

```
NISSERVER=master
```

```
$ sudo nano /etc/yp.conf
```

```
domain mytest.com server ubox.mytest.com
```



```
host 192.168.1.100
host 192.168.1.101
host 192.168.1.102
[...]
```

```
myclients (hibox,,) (client2,,) ...
```

```
$ sudo /usr/lib/yp/ypinit -m
```

```
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating
shadow.byname...
```

```
$ sudo /etc/init.d/nis restart
```

```
$ sudo make -C /var/yp
```

12.2.5 用/etc/exports 配置共享目录

```
$ sudo nano /etc/exports
```

```
/home    *(rw,async,no_subtree_check)
```

```
/home    192.168.1.0/24(rw,sync,insecure,no_subtree_check)
```

```
/var/lib hibox(rw,sync,no_subtree_check)  sbox(ro,sync,no_subtree_check)
```

```
$ sudo /etc/init.d/nfs-kernel-server restart
```

12.3 NFS 客户端的安装及配置

12.3.2 安装 NFS 客户端

```
$ sudo apt-get install nfs-common portmap nis
```

```
$ sudo dpkg-reconfigure nis
```

12.3.3 配置 NFS 客户端

1. Portmap 安全设置

```
$ sudo nano /etc/hosts.deny
```

```
portmap : ALL
```

```
$ sudo nano /etc/hosts.allow
```

```
portmap : 192.168.1.10
```

2. 配置名字服务

```
$ sudo nano /etc/passwd
```

```
+:::~:
```

```
$ sudo nano /etc/group
```

```
+:::
```

```
$ sudo nano /etc/passwd
```

```
+:::~:
```

3. 修改/etc/yp.conf

```
$ sudo nano /etc/yp.conf
```



```
ypserver 192.168.1.10
```

```
$ sudo /etc/init.d/nis restart
```

4. NFS 挂载

```
$ sudo mount 192.168.1.10:/home /home
```

```
$ sudo nano /etc/fstab
```

```
192.168.1.10:/home /home nfs rsize=8192,wsiz=8192,timeo=14,intr
```

第 13 章

与 Windows 共舞：Samba

13.2 安装 Samba 并测试

13.2.1 安装 Samba

```
$ sudo apt-get install samba
```

13.3 Samba 配置

13.3.1 最简单的 Samba 配置

```
$ sudo mv /etc/samba/smb.conf /etc/samba/smb.conf-orig
```

```
$ sudo nano /etc/samba/smb.conf
```

```
[global]
security=share

[myshare]
path=/usr/share/doc/samba
public=yes
```

```
$ testparm
```

```
Load smb config files from /etc/samba/smb.conf
Processing section "[myshare]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
```

```
[global]
    security = SHARE

[myshare]
    path = /usr/share/doc/samba
```

```
guest ok = Yes
```

```
$ sudo /etc/init.d/samba restart
```

13.3.2 Samba 的安全认证

```
$ sudo nano /etc/samba/smb.conf
```

```
[global]
security=user

[myshare]
path=/usr/share/doc/samba
valid users=hiweed
public=no
```

```
$ sudo /etc/init.d/samba restart
```

1. 在 Ubuntu 中创建用户

```
$ sudo adduser hiweed
```

```
Adding user `hiweed' ...
Adding new group `hiweed' (1001) ...
Adding new user `hiweed' (1001) with group `hiweed' ...
Creating home directory `/home/hiweed' ...
Copying files from `/etc/skel' ...
Enter new UNIX password: <-- 输入密码
Retype new UNIX password: <-- 再次输入密码
passwd: password updated successfully
Changing the user information for hiweed
Enter the new value, or press ENTER for the default
  Full Name []: Hiweed Leng <-- 输入用户全名 (可选)
  Room Number []: <-- 输入房间号码 (可选)
  Work Phone []: <-- 输入工作电话号码 (可选)
  Home Phone []: <-- 输入家庭电话号码 (可选)
  Other []:
Is the information correct? [y/N] <-- 按 y 键确认
```

2. 在 Samba 中创建用户

```
$ sudo smbpasswd -a hiweed
```

```
New SMB password: <-- 创建 Samba 密码
Retype new SMB password: <-- 再次输入密码
Added user hiweed.
```

```
$ sudo /etc/init.d/samba restart
```




4. Samba 用户密码修改

```
$ sudo smbpasswd hiweed
```

```
New SMB password: <-- 输入新密码
```

```
Retype new SMB password: <-- 再次输入新密码
```

13.3.4 文件写入实验

```
$ sudo nano /etc/samba/smb.conf
```

```
[global]
security=user

[myshare]
path=/usr/share/doc/samba
valid users=hiweed
writeable=yes
public=no
```

```
$ sudo /etc/init.d/samba restart
```

```
$ ls /usr/share/doc/ -l |grep sam
```

```
drwxr-xr-x 2 root root 4096 2009-03-25 23:01 samba
```

```
$ sudo nano /etc/samba/smb.conf
```

```
[global]
security=user

[myshare]
path=/home/hiweed
valid users=hiweed
writeable=yes
public=no
```

```
$ sudo /etc/init.d/samba restart
```

```
$ ls -l
```

```
-rw-r--r-- 1 hiweed hiweed 70 2009-10-13 03:17 mylist.txt
-rwxr--r-- 1 hiweed hiweed 70 2009-10-13 03:17 复件 mylist.txt
```

```
$ sudo chown hiweed -R /path/to/your/directory
```

13.4 基本的家目录共享方案



13.4.1 创建私人目录

```
$ ls -al /etc/skel
total 20
drwxr-xr-x  2 root root 4096 2009-03-31 16:17 .
drwxr-xr-x 69 root root 4096 2009-03-31 02:11 ..
-rw-r--r--  1 root root  220 2008-05-12 14:33 .bash_logout
-rw-r--r--  1 root root 2940 2008-05-12 14:33 .bashrc
-rw-r--r--  1 root root  586 2008-05-12 14:33 .profile
```

```
$ sudo mkdir /etc/skel/personal
$ ls -l /etc/skel
total 4
drwxr-xr-x 2 root root 4096 2009-03-31 16:23 personal
```

```
$ sudo chmod g=o /etc/skel/personal
$ ls -l /etc/skel
total 4
drwx----- 2 root root 4096 2009-03-31 16:29 personal
```

13.4.2 创建新用户

1. 创建系统用户

```
$ sudo adduser shangning
Adding user `shangning' ...
Adding new group `shangning' (1002) ...
Adding new user `shangning' (1002) with group `shangning' ...
Creating home directory `/home/shangning' ...
Copying files from `/etc/skel' ...      <-- 系统正在复制/etc/skel/目录下的内容
Enter new UNIX password:                <-- 请输入密码
Retype new UNIX password:               <-- 请再次输入密码
passwd: password updated successfully
Changing the user information for shangning
Enter the new value, or press ENTER for the default
  Full Name []:                        <-- 这里你可以输入用户的全名（可选）
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [y/N] y      <-- 按 y 键确认
```

```
total 4
drwx----- 2 shangning shangning 4096 2009-03-31 16:34 personal
```

```
$ ls -l /home/shangning/personal/
```

```
ls: cannot open directory /home/shangning/personal/: Permission denied
```

2. 将用户加入 Samba

```
$ sudo smbpasswd -a shangning
```

```
New SMB password:          <-- 请输入密码
Retype new SMB password:   <-- 再次输入密码
Added user shangning.
```

13.4.3 配置 Samba

1. 最简单的[homes]共享配置

```
$ sudo mv /etc/samba/smb.conf /etc/samba/conf.conf-backup
$ sudo nano /etc/samba/smb.conf
```

```
[global]
workgroup = HIWEEDGROUP

[homes]
guest ok = no
read only = no
```

```
$ sudo /etc/init.d/samba restart
```

```
$ sudo nano /etc/samba/smb.conf
```

```
[global]
workgroup = HIWEEDGROUP

[homes]
browseable = no
guest ok = no
read only = no
```

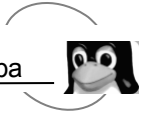
```
$ sudo /etc/init.d/samba restart
```

13.5 其他共享方案

13.5.1 共享光驱

```
$ sudo nano /etc/fstab
```

```
/dev/scd0    /cdrom    udf,iso9660 user,noauto,exec,utf8 0    0
```



```
$ sudo nano /etc/samba/smb.conf
```

```
[cdrom]
read only = yes
locking = no
path = /cdrom
guest ok = no
preexec = /bin/mount /cdrom
postexec = /bin/umount /cdrom
```

```
$ sudo /etc/init.d/samba restart
```

13.5.2 小组共享

1. 创建财务组

```
$ sudo addgroup caiwu
```

```
Adding group `caiwu' (GID 1003) ...
Done.
```

2. 将用户加入财务组

```
$ sudo adduser hiweed caiwu
```

```
Adding user `hiweed' to group `caiwu' ...
Adding user hiweed to group caiwu
Done.
```

3. 创建共享目录

```
$ sudo mkdir /home/samba/caiwu -p
```

4. 配置 Samba

```
$ sudo nano /etc/samba/smb.conf
```

```
[caiwu]
comment = 闲人免进 —— 财务部
path = /home/samba/caiwu
read only = no
guest ok = no
browseable = yes
create mask = 0660
directory mask = 0770
valid users = @caiwu
force group = caiwu
```

```
$ sudo /etc/init.d/samba restart
```

5. 设置目录权限

```
$ sudo chgrp caiwu /home/samba/caiwu/
```

```
$ sudo chmod 770 /home/samba/caiwu/
```

第 14 章

最佳虚拟化方案：OpenVZ

14.2 安装 OpenVZ

14.2.1 安装前的准备

```
$ sudo ln -sf /bin/bash /bin/sh
```

```
$ sudo /etc/init.d/apparmor stop  
$ sudo update-rc.d -f apparmor remove  
$ sudo apt-get remove apparmor apparmor-utils
```

14.2.2 安装 OpenVZ

```
$ sudo apt-get install linux-openvz vzctl vzquota
```

14.2.3 配置 OpenVZ

1. 内核参数调整

```
$ sudo nano /etc/sysctl.conf
```

```
[...]  
net.ipv4.conf.all.rp_filter=1  
net.ipv4.icmp_echo_ignore_broadcasts=1  
net.ipv4.conf.default.forwarding=1  
net.ipv4.conf.default.proxy_arp = 0  
net.ipv4.ip_forward=1  
kernel.sysrq = 1  
net.ipv4.conf.default.send_redirects = 1  
net.ipv4.conf.all.send_redirects = 0  
net.ipv4.conf.eth0.proxy_arp=1  
[...]
```

```
$ sudo sysctl -p
```



2. 修改 VE 全局配置

```
$ sudo nano /etc/vz/vz.conf
```

```
[...]  
NEIGHBOUR_DEVS=all  
[...]
```

3. 修改 vps.basic 配置文件

```
$ sudo nano /etc/vz/conf/ve-vps.basic.conf-sample
```

```
[...]  
KMEMSIZE="213770490:268435456"  
PRIVVMPAGES="655360:696320"  
NUMPROC="480:480"  
TCPSNDBUF="4703360:4703360"  
TCPRCVBUF="4703360:4703360"  
DGRAMRCVBUF="462144:462144"  
NUMOTHERSOCK="1000:1000"  
DISKSPACE="20485760:21530240"  
DISKINODES="2000000:2200000"  
[...]
```

```
CAPABILITY="CHOWN:on DAC_READ_SEARCH:on SETGID:on SETUID:on NET_BIND_ SERVICE:on  
NET_ADMIN:on SYS_CHROOT:on SYS_NICE:on"
```

4. 重新启动服务器

```
$ sudo reboot
```

```
$ uname -r
```

```
2.6.24-19-openvz
```

14.3 虚拟机的基本操作

14.3.1 虚拟机的创建

1. 下载操作系统模板

```
$ wget http://download.openvz.org/template/precreated/old/ubuntu-8.04-i386-minimal.tar.  
$ sudo mv ubuntu-8.04-i386-minimal.tar.gz /var/lib/vz/template/cache/
```

2. 创建虚拟机

```
$ sudo vzctl create 101 --ostemplate ubuntu-8.04-i386-minimal
```


3. 修改虚拟机配置

```
$ sudo vzctl set 101 --hostname test.mytest.com --save
$ sudo vzctl set 101 --ipadd 192.168.1.200 --save
```

```
$ sudo vzctl set 101 --numothersock 480 --save
```

```
$ sudo vzctl set 101 --nameserver 202.102.128.68 --nameserver 202.102.134.68
-sav e
```

```
$ man vzctl
```

14.3.2 虚拟机的启停

1. 启动虚拟机

```
$ sudo vzctl start 101
```

2. 进入/退出虚拟机

```
$ sudo vzctl enter 101
```

```
$ exit
```



3. 停止虚拟机

```
$ sudo vzctl stop 101
```

4. 删除虚拟机

```
$ sudo vzctl destroy 101
```

14.4 vzctl 用法详解

14.4.1 vzctl 基本用法

```
vzctl [flags] 子命令 虚拟机编号 [参数 1] [参数 2] [参数 3...]
```

14.4.2 创建虚拟机

```
vzctl [flags] create veid --ostemplate name [--config name] [--private path] [--root path] [--ipadd addr] [--hostname name]
```

14.4.3 虚拟机的启停等操作

```
vzctl [flags] start | stop | restart | enter | destroy | mount | umount | status | veid
```

VEID 虚拟机编号 是否存在 是否加载 是否运行

14.4.4 设置虚拟机参数

```
vzctl [flags] set veid [要设置的选项、值] [--save]
```

3. 资源限制选项

```
vzctl set veid --privvmpages 5M:6M
```

14.4.5 其他命令和参数

```
vzctl [flags] exec | exec2 veid command [arg ...]
```

```
$ vzctl exec 1000 /bin/ls -la
```

```
$ vzctl exec 1000 'ls -l / | sort'
```

```
vzctl runscript veid <script>
```

```
vzctl --help | --version
```

14.6 VE 的备份与恢复



14.6.1 安装 vldump

```
$ wget http://www.proxmox.com/cms_proxmox/cms/upload/vldump/vldump_1.1-1_all.deb
```

```
$ sudo dpkg -i vldump_1.1-1_all.deb
```

```
$ sudo apt-get -f install
```

14.6.2 vldump 的用法

```
vldump 选项 [--all | VEID]
```

14.6.3 备份 VE

```
$ sudo v2vdump 777
```

```
$ sudo v2vdump --suspend 777
```

```
$ sudo v2vdump --suspend --all --mailto root
```

```
$ sudo v2vdump --dumpdir /path/to/backup --snapshot 777
```

14.6.4 恢复 VE

```
$ sudo v2vdump --restore /path/to/v2vdump-777.tar 600
```

14.7 OpenVZ 排错

```
$ cat /proc/user_beancounters
```

uid resource	held	maxheld	barrier	limit	failcnt
230:kmemsize	12260500	25274790	213770490	213770490	0
lockedpages	0	0	256	256	0
privmpages	180676	473024	655360	696320	0
shmpages	16	2928	21504	21504	0
dummy	0	0	0	0	0
numproc	143	287	360	480	88
physpages	135350	401356	0	2147483647	0
vmguarpages	0	0	33792	2147483647	0
oomguarpages	135350	401356	26112	2147483647	0
numtcpsock	93	342	360	360	0
numflock	43	133	188	206	0
numpty	1	4	16	16	0
numsiginfo	0	148	256	256	0
tcpsndbuf	584704	4011648	4703360	4703360	0
tcprcvbuf	559488	3383488	4703360	4703360	0
othersockbuf	507520	1132416	1126080	2097152	3055
dgramrcvbuf	0	337600	462144	462144	0
numothersock	324	724	1000	1000	43
dcachesize	507150	841995	3409920	3624960	0
numfile	3720	7871	9312	9312	0
dummy	0	0	0	0	0
dummy	0	0	0	0	0
dummy	0	0	0	0	0
numiptent	10	10	128	128	0



15.1 安装 Bind9

```
$ sudo apt-get install bind9 dnsutils bind9-doc
```

15.3 配置 Bind9

15.3.1 Bind9 配置文件介绍

```
directory "/var/cache/bind";
```

15.3.2 DNS 记录类型

1. A 记录

```
www      IN      A       192.168.1.10
```

2. 别名记录（CNAME）

```
www      IN      A       192.168.1.10
mail     IN      CNAME   www
```

3. MX 记录

```
          IN      MX       mail.mytest.com.
mail     IN      A       192.168.1.100
```

4. NS 记录

```
          IN      NS       ns.mytest.com.
ns       IN      A       192.168.1.200
```

15.3.3 DNS 缓存服务器的配置

1. 转发配置

```
$ sudo nano /etc/bind/named.conf.options
```

```
forwarders {
    202.102.128.68;
```

```
202.102.134.68;  
};
```

```
$ sudo /etc/init.d/bind9 restart
```

2. 测试

```
$ sudo nano /etc/resolv.conf
```

```
search mytest.com  
nameserver 192.168.1.10
```

```
$ dig google.com
```

```
; <<>> DiG 9.4.2-P2 <<>> google.com  
;; global options: printcmd  
;; Got answer:  
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 52542  
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 4, ADDITIONAL: 0  
  
;; QUESTION SECTION:  
;google.com.                IN      A  
  
;; ANSWER SECTION:  
google.com.                125     IN      A      74.125.67.100  
google.com.                125     IN      A      209.85.171.100  
google.com.                125     IN      A      74.125.45.100  
  
;; AUTHORITY SECTION:  
google.com.                324418  IN      NS      ns1.google.com.  
google.com.                324418  IN      NS      ns2.google.com.  
google.com.                324418  IN      NS      ns3.google.com.  
google.com.                324418  IN      NS      ns4.google.com.  
  
;; Query time: 29 msec  
;; SERVER: 192.168.1.10#53(192.168.1.10)  
;; WHEN: Tue Feb 10 04:32:30 2009  
;; MSG SIZE rcvd: 148
```

15.3.4 主 DNS 服务器的配置

1. 创建正向 Zone 文件

```
$ sudo nano /etc/bind/named.conf.local
```

```
zone "mytest.com" {  
    type master;  
    file "db.mytest.com";  
};
```




```
$ sudo cp /etc/bind/db.local /var/cache/bind/db.mytest.com
```

```
$ sudo nano /var/cache/bind/db.mytest.com
```

```
;
; BIND data file for mytest.com
;
$TTL      604800
@         IN      SOA      mytest.com. root.mytest.com. (
                        1      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.
@         IN      A        192.168.1.10
ns        IN      A        192.168.1.10
ubox      IN      A        192.168.1.10
www       IN      CNAME    ubox
@         IN      AAAA     ::1
```

```
$ sudo /etc/init.d/bind9 restart
```

2. 创建反向 Zone 文件

```
$ sudo nano /etc/bind/named.conf.local
```

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "reverse/db.192.168.1";
};
```

```
$ sudo mkdir /var/cache/bind/reverse/
$ sudo cp /etc/bind/db.127 /var/cache/bind/reverse/db.192.168.1
```

```
$ sudo nano /var/cache/bind/reverse/db.192.168.1
```

```
;
; BIND reverse data file for 192.168.1
;
$TTL      604800
@         IN      SOA      ns.mytest.com. root.mytest.com. (
```

```
                1           ; Serial
                604800       ; Refresh
                86400        ; Retry
                2419200      ; Expire
                604800 )     ; Negative Cache TTL
;
@      IN      NS       ns.
125    IN      PTR      ns.mytest.com
125    IN      PTR      ubox.mytest.com
125    IN      PTR      www.mytest.com
```

```
$ sudo /etc/init.d/bind9 restart
```

3. 测试

```
$ ping mytest.com
```

```
$ dig www.mytest.com
$ dig 1.168.192.in-addr.arpa. AXFR
```

```
$ named-checkzone mytes.com db.mytest.com
```

```
$ named-checkzone mytes.com reverse/db.192.168.1
```

15.3.5 从 DNS 服务器的配置

1. 主服务器的配置

```
$ sudo nano /etc/bind/named.conf.local
```

```
zone "mytest.com" {
    type master;
    file "db.mytest.com";
    allow-transfer {192.168.1.100};
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "reverse/db.192.168.1";
    allow-transfer {192.168.1.100};
};
```

2. 从服务器的配置

```
$ sudo apt-get install bind9 dnsutils
```



```
$ sudo mkdir /var/cache/bind/reverse
$ sudo chown bind.bind /var/cache/bind/reverse/
```

```
$ sudo nano /etc/bind/named.conf.local
```

```
zone "mytest.com" {
    type slave;
    file "db.mytest.com";
    masters {192.168.1.125;};
};

zone "1.168.192.in-addr.arpa" {
    type slave;
    file "reverse/db.192.168.1";
    masters {192.168.1.125;};
};
```

```
$ sudo /etc/init.d/bind9 restart
```

```
$ ls /var/cache/bind/ -R
```

```
/var/cache/bind/:
db.mytest.com  reverse

/var/cache/bind/reverse:
db.192.168.1
```

```
$ sudo tail -n50 /var/log/syslog
```

```
named[4801]: zone mytest.com/IN: sending notifies (serial 6)
named[4801]: zone 1.168.192.in-addr.arpa/IN: Transfer started.
named[4801]: transfer of '1.168.192.in-addr.arpa/IN' from 192.168.1.125#53:
connected using 192.168.1.140#58618
named[4801]: zone 1.168.192.in-addr.arpa/IN: transferred serial 1
named[4801]: transfer of '1.168.192.in-addr.arpa/IN' from 192.168.1.125#53: end
of transfer
named[4801]: zone 1.168.192.in-addr.arpa/IN: sending notifies (serial 1)
```

15.4 让 Bind9 运行在 Chroot 环境

```
$ sudo /etc/init.d/bind9 stop
```



15.4.1 创建 Chroot 环境

```
$ sudo mkdir /var/lib/bind/etc  
$ sudo mkdir /var/lib/bind/dev  
$ sudo mkdir -p /var/lib/bind/var/run/bind/run
```

```
$ sudo mv /etc/bind /var/lib/bind/etc/  
$ sudo mv /var/cache/bind/ /var/lib/bind/var/cache/
```

```
$ sudo ln -s /var/lib/bind/etc/bind /etc/bind
```

```
$ sudo mknod /var/lib/bind/dev/null c 1 3  
$ sudo mknod /var/lib/bind/dev/random c 1 8  
$ sudo chmod 666 /var/lib/bind/dev/null /var/lib/bind/dev/random
```

```
$ sudo cp /etc/localtime /var/lib/bind/etc/
```

```
$ sudo chown -R bind:bind /var/lib/bind/var/*  
$ sudo chown -R bind:bind /var/lib/bind/etc/bind
```

15.4.2 Bind9 配置

```
$ sudo nano /etc/default/bind9
```

```
OPTIONS="-u bind -t /var/lib/bind"
```

15.4.3 日志路径设置

```
$ sudo nano /etc/default/syslogd
```

```
SYSLOGD=" -a /var/lib/bind/dev/log"
```

15.4.4 测试

```
$ sudo /etc/init.d/bind9 start
```

15.5 Bind9 排错

15.5.1 DNS 测试

1. /etc/resolv.conf

```
nameserver 192.168.1.10  
nameserver 192.168.1.100
```

2. ping 工具

```
$ ping mytest.com
```

```
PING mytest.com (192.168.1.10) 56(84) bytes of data.  
64 bytes from ns (192.168.1.10): icmp_seq=1 ttl=64 time=0.800 ms  
64 bytes from ns (192.168.1.10): icmp_seq=2 ttl=64 time=0.803 ms  
64 bytes from ns (192.168.1.10): icmp_seq=3 ttl=64 time=0.794 ms  
64 bytes from ns (192.168.1.10): icmp_seq=4 ttl=64 time=0.810 ms
```

3. dig 工具

```
$ dig -x 127.0.0.1
```



```
;; Query time: 1 msec
;; SERVER: 192.168.1.10#53 (192.168.1.10)
```

```
$ dig ubuntu.com
```

```
;; Query time: 47 msec
;; SERVER: 192.168.1.10#53 (192.168.1.10)
```

```
;; Query time: 1 msec
```

4. named-checkzone 工具

```
$ named-checkzone mytest.com /etc/bind/db.mytest.com
```

```
zone mytest.com/IN: loaded serial 5
OK
```

```
$ named-checkzone mytest.com /etc/bind/db.192.168.1
```

```
zone mytest.com/IN: loaded serial 5
OK
```

15.5.2 日志文件

```
logging {
    category default { default_syslog; default_debug; };
    category unmatched { null; };
};
```

```
$ sudo nano /etc/bind/named.conf.local
```

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3;
    };
    category queries { query.log; };
};
```

```
$ sudo touch /var/log/query.log  
$ sudo chown bind /var/log/query.log
```

```
$ sudo nano /etc/apparmor.d/usr.sbin.named
```

```
/var/log/query.log w,
```

```
$ cat /etc/apparmor.d/usr.sbin.named | sudo apparmor_parser -r
```

```
$ sudo /etc/init.d/bind9 restart
```


第 16 章

DNS 轮询

16.4 DNS 轮询的测试

```
$ nslookup
> server 192.168.1.10
> www.mytest.net
```

```
Name:    www.mytest.net
Address: 192.168.1.1
Name:    www.mytest.net
Address: 192.168.1.2
Name:    www.mytest.net
Address: 192.168.1.3
Name:    www.mytest.net
Address: 192.168.1.4
```

第 17 章

最佳 DHCP 服务器方案

17.3 安装 DHCP 服务器软件

```
$ sudo apt-get install dhcp3-server
```

On what network interfaces should the DHCP server listen? <-- 输入eth0

Please configure the DHCP server as soon as the installation finishes. <-- **Ok**
The version 3 DHCP server is now non-authoritative by default <-- **Ok**

```
* Starting DHCP server dhcpd3 [fail]
invoke-rc.d: initscript dhcp3-server, action "start" failed.
```



17.4 配置 DHCP 服务器

17.4.1 网络环境介绍

```
$ cat /etc/network/interfaces

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.10
    gateway 192.168.1.1
    netmask 255.255.255.0
```

17.4.2 DHCP 配置

```
$ sudo mv /etc/dhcp3/dhcpd.conf /etc/dhcp3/dhcpd.conf-back
```

```
$ sudo touch /etc/dhcp3/dhcpd.conf
```

```
$ sudo nano /etc/dhcp3/dhcpd.conf
```

```
default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.1;
option domain-name-servers 192.168.1.10, 192.168.1.100;
option domain-name "mytest.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.20 192.168.1.99;
    range 192.168.1.150 192.168.1.240;
}
```

```
$ sudo /etc/init.d/dhcp3-server start
```

17.4.3 测试

```
$ cat /etc/network/interfaces

# The primary network interface
auto eth0
```

```
iface eth0 inet dhcp
```

```
$ sudo /etc/init.d/networking restart
```

```
[...]  
DHCP OFFER of 192.168.1.240 from 192.168.1.10  
[...]  
DHCP ACK of 192.168.1.240 from 192.168.1.10
```

```
$ ifconfig eth0
```

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:57:6b:21  
          inet addr:192.168.1.240  Bcast:192.168.1.255  Mask:255.255.255.0  
[...]
```

```
$ cat /etc/resolv.conf
```

```
search mytest.com  
nameserver 192.168.1.10  
nameserver 192.168.1.100
```

17.5 DHCP 排错

```
$ ps aux | grep dhcpd
```

```
dhcpd      5373  0.0  2.0  2868 1272 ?        Ss   21:40   0:00 /usr/sbin/dhcpd3 -q  
-pf /var/run/dhcp3-server/dhcpd.pid -cf /etc/dhcp3/dhcpd.conf
```

```
$ sudo netstat -uap | grep dhcpd
```

```
udp        0      0 *:bootps          *:.*                5373/dhcpd3
```

```
$ sudo tail -n 100 /var/log/syslog | grep dhc
```

```
$ sudo cat /var/lib/dhcp3/dhcpd.leases
```

```
[...]  
lease 192.168.1.240 {  
  starts 5 2009/02/13 04:43:45;  
  ends 5 2009/02/13 04:53:45;  
  binding state active;  
  next binding state free;  
  hardware ethernet 00:0c:29:57:6b:21;
```



```
}  
[...]
```

第 18 章

负载均衡、高可用的 Web 集群

18.3 架构的实现

18.3.1 Web 服务器的安装及配置

1. 安装 Apache2

```
$ sudo apt-get install apache2
```

2. 修改 apache2.conf

```
$ sudo nano /etc/apache2/apache2.conf
```

```
[...]
#LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Age nt}i\"" combined
[...]
```

3. 创建文件 check.txt

```
$ sudo touch /var/www/check.txt
```

4. 修改虚拟主机配置

```
$ sudo nano /etc/apache2/sites-available/default
```

```
[...]
SetEnvIf Request_URI "^/check\.txt$" dontlog
CustomLog /var/log/apache2/access.log combined env=!dontlog
[...]
```

```
$ sudo /etc/init.d/apache2 restart
```

18.3.2 HAProxy 的安装及配置

1. 安装 HAProxy

```
$ sudo apt-get install haproxy
```



2. 配置 haproxy.cfg

```
$ sudo mv /etc/haproxy.cfg /etc/haproxy.cfg-back
$ sudo nano /etc/haproxy.cfg
```

```
global
    log 127.0.0.1    local0
    log 127.0.0.1    local1 notice
    maxconn 4096
    user haproxy
    group haproxy

defaults
    log        global
    mode       http
    option     httplog
    option     dontlognull
    retries    3
    redispatch
    maxconn    2000
    timeout    5000
    clitimeout 50000
    srvtimeout 50000

listen webfarm 192.168.1.14:80
    mode http
    stats enable
    stats auth admin:password
    balance roundrobin
    cookie JSESSIONID prefix
    option httpclose
    option forwardfor
    option httpchk HEAD /check.txt HTTP/1.0
    server webA 192.168.1.12:80 cookie A check
    server webB 192.168.1.13:80 cookie B check
```

3. 修改/etc/sysctl.conf

```
$ sudo nano /etc/sysctl.conf
```

```
net.ipv4.ip_nonlocal_bind=1
```

```
$ sudo sysctl -p
```

4. 让 HAProxy 自动启动

```
$ sudo nano /etc/default/haproxy
```

```
# Set ENABLED to 1 if you want the init script to start haproxy.
ENABLED=1
# Add extra flags here.
#EXTRA_OPTS="-de -m 16"
```

18.3.3 Keepalived 的安装及配置

1. 安装 Keepalived

```
$ sudo apt-get install keepalived
```

2. 配置 Keepalived

```
$ sudo nano /etc/keepalived/keepalived.conf
```

```
vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 2                # 每 2 秒钟检查一次
    weight 2
}

vrrp_instance VI_1 {
    interface eth0
    state MASTER
    virtual_router_id 51
    priority 101              # 101为“主”，100为“从”
    virtual_ipaddress {
        192.168.1.14
    }
    track_script {
        chk_haproxy
    }
}
```

```
$ sudo /etc/init.d/keepalived start
```

```
$ sudo nano /etc/keepalived/keepalived.conf
```

```
vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 2                # 每 2 秒钟检查一次
    weight 2
}

vrrp_instance VI_1 {
    interface eth0
    state MASTER
    virtual_router_id 51
    priority 100              # 101为“主”，100为“从”
    virtual_ipaddress {
        192.168.1.14
    }
    track_script {
        chk_haproxy
    }
}
```




```
$ sudo /etc/init.d/keepalived start
```

```
$ ip addr sh eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 00:0c:29:4e:67:1a brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.10/24 brd 192.168.1.255 scope global eth0  
    inet 192.168.1.14/24 brd 192.168.1.255 scope global secondary eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 00:0c:29:34:d7:7e brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.11/24 brd 192.168.1.255 scope global eth0
```

```
$ sudo /etc/init.d/haproxy start
```

18.4 测试

18.4.2 负载均衡节点故障模拟

```
$ sudo /etc/init.d/keepalived stop
```

```
$ ping 192.168.1.14
```

```
[...]  
64 bytes from 192.168.1.14: icmp_seq=20 ttl=64 time=0.901 ms  
64 bytes from 192.168.1.14: icmp_seq=21 ttl=64 time=217 ms  
From 192.168.1.11 icmp_seq=22 Destination Host Unreachable  
64 bytes from 192.168.1.14: icmp_seq=23 ttl=64 time=1961 ms  
64 bytes from 192.168.1.14: icmp_seq=24 ttl=64 time=975 ms
```

18.5 HAProxy 的 Web 统计页面

```
stats enable
stats auth admin:password
```

第 19 章

负载均衡、高可用的 MySQL 集群

19.2 管理节点（MGM）的安装及配置

19.2.1 安装 MySQL

```
$ sudo apt-get update install mysql-server
```

19.2.2 配置 ndb_mgmd.cnf

```
$ sudo nano /etc/mysql/ndb_mgmd.cnf
```

```
[NDBD DEFAULT]
NoOfReplicas=2

[MYSQLD DEFAULT]
[NDB_MGMD DEFAULT]
[TCP DEFAULT]

[NDB_MGMD]    # 管理节点
HostName=192.168.1.10    # 本机（管理节点）的 IP 地址

[NDBD]    # 存储节点 1
HostName=192.168.1.13
DataDir=/var/lib/mysql-cluster
BackupDataDir=/var/lib/mysql-cluster/backup

[NDBD]    # 存储节点 2
HostName=192.168.1.14
DataDir=/var/lib/mysql-cluster
BackupDataDir=/var/lib/mysql-cluster/backup

# 有几个存储节点，就写几行 [MYSQLD]
[MYSQLD]
[MYSQLD]
```

```
$ sudo /etc/init.d/mysql-ndb-mgm start
```

19.3 存储节点（NDB）的安装及配置

19.3.1 安装 MySQL

```
$ sudo apt-get update install mysql-server
```

```
$ sudo /etc/init.d/mysql stop
```

19.3.2 配置 my.cnf

```
$ sudo mv /etc/mysql/my.cnf /etc/mysql/my.cnf-back
```

```
$ sudo nano /etc/mysql/my.cnf
```

```
[client]
socket = /var/run/mysqld/mysqld.sock
port   = 3306

[mysqld]
ndbcluster
ndb-connectstring=192.168.1.10 # 管理节点的 IP 地址
default-storage-engine=NDBCLUSTER

[mysql_cluster]
ndb-connectstring=192.168.1.10 # 管理节点的 IP 地址
```

```
$ sudo /etc/init.d/mysql-ndb start-initial
```

```
* Starting MySQL NDB Data Node ndbd                                error=2350
2009-02-15 22:20:55 [ndbd] INFO      -- Error handler restarting system
2009-02-15 22:20:55 [ndbd] INFO      -- Error handler shutdown completed -
exiting
sphase=0
exit=-1
```

```
$ ps aux|grep ndb|grep -v grep
```

```
$ sudo /etc/init.d/mysql start
```



19.4 阶段测试

19.4.1 集群连接状态测试

```
$ ndb_mgm
```

```
-- NDB Cluster -- Management Client --  
ndb_mgm>
```

```
ndb_mgm> show
```

```
Connected to Management Server at: localhost:1186  
Cluster Configuration  
-----  
[ndbd(NDB)] 2 node(s)  
id=2 @192.168.1.13 (Version: 5.0.51, Nodegroup: 0)  
id=3 @192.168.1.14 (Version: 5.0.51, Nodegroup: 0, Master)  
  
[ndb_mgmd(MGM)] 1 node(s)  
id=1 @192.168.1.10 (Version: 5.0.51)  
  
[mysqld(API)] 2 node(s)  
id=4 @192.168.1.13 (Version: 5.0.51)  
id=5 @192.168.1.14 (Version: 5.0.51)
```

```
ndb_mgm> quit
```

19.4.2 测试

1. 数据同步测试

```
$ mysql -u root -p
```

```
mysql> CREATE DATABASE clustertest;
```

```
Query OK, 1 row affected (0.24 sec)
```

```
mysql> USE clustertest;
```

```
Database changed
```

```
mysql> CREATE TABLE testtable (Count INT) ENGINE=NDBCLUSTER;
```

```
Query OK, 0 rows affected (0.24 sec)
```

```
mysql> INSERT INTO testtable () VALUES (1);
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM testtable;
```

```
+-----+
```

```
| Count |
```

```
+-----+
```

```
|      1 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
$ mysql -u root -p
```

```
mysql> CREATE DATABASE clustertest;
```

```
Query OK, 1 row affected (0.24 sec)
```

```
mysql> USE clustertest;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> SELECT * FROM testtable;
```

```
+-----+
```

```
| Count |
```

```
+-----+
```

```
|      1 |
```

```
+-----+
```

```
1 row in set (0.03 sec)
```

```
mysql> INSERT INTO testtable () VALUES (2);
```

```
Query OK, 1 row affected (0.23 sec)
```

```
mysql> quit
```

```
Bye
```

```
mysql> SELECT * FROM testtable;
```

```
+-----+
```

```
| Count |
```

```
+-----+
```

```
|      2 |
```

```
|      1 |
```



```
+-----+  
2 rows in set (0.00 sec)
```

```
mysql> quit
```

```
Bye
```

2. 故障模拟测试

```
$ sudo /etc/init.d/mysql-ndb stop
```

```
$ ndb_mgm
```

```
-- NDB Cluster -- Management Client --
```

```
ndb_mgm> show
```

```
Connected to Management Server at: localhost:1186
```

```
Cluster Configuration
```

```
-----
```

```
[ndbd(NDB)] 2 node(s)
```

```
id=2 (not connected, accepting connect from 192.168.1.13)
```

```
id=3 @192.168.1.14 (Version: 5.0.51, Nodegroup: 0, Master)
```

```
[ndb_mgmd(MGM)] 1 node(s)
```

```
id=1 @192.168.1.10 (Version: 5.0.51)
```

```
[mysqld(API)] 2 node(s)
```

```
id=4 @192.168.1.13 (Version: 5.0.51)
```

```
id=5 @192.168.1.14 (Version: 5.0.51)
```

```
ndb_mgm> quit
```

```
$ mysql -u root -p
```

```
mysql> USE clustertest;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> SELECT * FROM testtable;
```

```
+-----+
```

```
| Count |
```

```
+-----+
```

```
| 2 |
```

```
| 1 |
```

```
+-----+
```

```
2 row in set (0.03 sec)
```

```
mysql> INSERT INTO testtable () VALUES (3);
```

```
Query OK, 1 row affected (0.89 sec)
```




```
mysql> SELECT * FROM testtable;
```

```
+-----+
| Count |
+-----+
|      1 |
|      2 |
|      3 |
+-----+
```

```
3 rows in set (0.00 sec)
```

```
mysql> quit
```

```
Bye
```

```
$ sudo /etc/init.d/mysql-ndb start
```

```
$ mysql -u root -p
```

```
mysql> USE clustertest;
```

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> SELECT * FROM testtable;
```

```
+-----+
| Count |
+-----+
|      2 |
|      3 |
|      1 |
+-----+
```

```
3 rows in set (0.00 sec)
```

```
mysql> quit
```

```
Bye
```

19.5 实现负载均衡

19.5.2 让内核支持 IPVS

```
$ sudo modprobe ip_vs_dh
$ sudo modprobe ip_vs_ftp
$ sudo modprobe ip_vs
$ sudo modprobe ip_vs_lblc
$ sudo modprobe ip_vs_lblcr
$ sudo modprobe ip_vs_lc
$ sudo modprobe ip_vs_nq
$ sudo modprobe ip_vs_rr
```

```
$ sudo modprobe ip_vs_sed
$ sudo modprobe ip_vs_sh
$ sudo modprobe ip_vs_wlc
$ sudo modprobe ip_vs_wrr
```

```
$ sudo nano /etc/modules
```

```
ip_vs_dh
ip_vs_ftp
ip_vs
ip_vs_lblc
ip_vs_lblcr
ip_vs_lc
ip_vs_nq
ip_vs_rr
ip_vs_sed
ip_vs_sh
ip_vs_wlc
ip_vs_wrr
```

```
$ sudo nano /etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1
```

```
$ sudo sysctl -p
```

19.5.3 安装 heartbeat、ldirectord 等软件

下面我们来安装 heartbeat、ldirectord，以及要用到的其他软件。在 mysql-lb1.mytest.com 和 mysql-lb2.mytest.com 上，执行下面的命令：

```
$ sudo apt-get install heartbeat ldirectord
$ sudo apt-get install libdbi-perl libdbd-mysql-perl libmysqlclient15-dev
```

19.5.4 配置 heartbeat

```
$ sudo nano /etc/hosts
```

```
127.0.0.1      localhost
192.168.1.11   mysql-lb1.mytest.com  mysql-lb1
192.168.1.12   mysql-lb2.mytest.com  mysql-lb2
```



```
$ sudo nano /etc/ha.d/ha.cf
logfacility          local0
bcast               eth0
mcast eth0 225.0.0.1 694 1 0
auto_failback off
node                mysql-lb1
node                mysql-lb2
respawn hacluster /usr/lib/heartbeat/ipfail
apiauth ipfail gid=haclient uid=hacluster
```

```
$ sudo nano /etc/ha.d/haresources
mysql-lb1 \
    ldirectord::ldirectord.cf \
    LVSSyncDaemonSwap::master \
    IPaddr2::192.168.1.15/24/eth0/192.168.1.255
```

```
$ sudo nano /etc/ha.d/authkeys
auth 3
3 md5 A46fsdgCH
```

```
$ sudo chmod 600 /etc/ha.d/authkeys
```

19.5.5 配置 ldirectord

```
$ sudo nano /etc/ha.d/ldirectord.cf
# Global Directives
checktimeout=10
checkinterval=2
autoreload=no
logfile="local0"
quiescent=yes
virtual = 192.168.1.15:3306
    service = mysql
    real = 192.168.1.13:3306 gate
    real = 192.168.1.14:3306 gate
    checktype = negotiate
    login = "ldirector"
    passwd = "ldirectorpassword"
    database = "ldirectordb"
    request = "SELECT * FROM connectioncheck"
    scheduler = wrr
```

```
$ sudo update-rc.d -f ldirectord remove
$ sudo update-rc.d -f heartbeat remove
$ sudo update-rc.d heartbeat start 90 2 3 4 5 . stop 05 0 1 6 .
```

19.5.6 NDB 节点配置

1. 为 ldirector 创建数据库

```
$ mysql -u root -p
```

```
mysql> GRANT ALL ON ldirectordb.* TO 'ldirector'@'%' IDENTIFIED BY  
'ldirectorpasswd';  
mysql> FLUSH PRIVILEGES;  
mysql> CREATE DATABASE ldirectordb;  
mysql> USE ldirectordb;  
mysql> CREATE TABLE connectioncheck (Status INT) ENGINE=NDBCLUSTER;  
mysql> INSERT INTO connectioncheck () VALUES (1);  
mysql> quit
```

```
$ mysql -u root -p
```

```
mysql> GRANT ALL ON ldirectordb.* TO 'ldirector'@'%' IDENTIFIED BY  
'ldirectorpasswd';  
mysql> FLUSH PRIVILEGES;  
mysql> CREATE DATABASE ldirectordb;  
mysql> quit
```

2. 设置 IP 路由

```
$ sudo apt-get install iproute
```

```
$ sudo nano /etc/sysctl.conf
```

```
net.ipv4.conf.all.arp_ignore = 1  
net.ipv4.conf.eth0.arp_ignore = 1  
net.ipv4.conf.all.arp_announce = 2  
net.ipv4.conf.eth0.arp_announce = 2
```

```
$ sudo sysctl -p
```

3. 设置虚拟 IP 地址

```
$ sudo nano /etc/network/interfaces
```



```
auto lo:0
iface lo:0 inet static
    address 192.168.1.15
    netmask 255.255.255.255
pre-up sysctl -p > /dev/null
```

```
$ sudo ifup lo:0
```

19.5.7 测试

```
$ sudo /etc/init.d/ldirectord stop
$ sudo /etc/init.d/heartbeat start
```

```
$ sudo reboot
```

1. ldirectord 状态检查

```
$ ldirectord ldirectord.cf status
```

```
ldirectord for /etc/ha.d/ldirectord.cf is running with pid: 4584
```

```
ldirectord is stopped for /etc/ha.d/ldirectord.cf
```

2. 虚拟 IP 状态检查

```
$ ip addr sh eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:16:3e:45:fc:f8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.0.255 scope global eth0
    inet 192.168.1.15/24 brd 192.168.0.255 scope global secondary eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:16:3e:16:c1:4e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.12/24 brd 192.168.0.255 scope global eth0
```

3. IPVS 状态检查

```
$ sudo ipvsadm -L -n
```

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.1.15:3306 wrr
  -> 192.168.1.13:3306             Route    1      0          0
  -> 192.168.1.14:3306             Route    1      0          0
```

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
```

```
$ sudo /etc/ha.d/resource.d/LVSSyncDaemonSwap master status
```

```
master running
(ipvs_syncmaster pid: 4704)
```



```
master stopped  
(ipvs_syncbackup pid: 1440)
```

4. MySQL 测试

```
$ mysql -h 192.168.1.15 -u ldirector -p
```

5. 故障模拟测试

```
$ ping 192.168.1.15
```

```
[...]  
64 bytes from 192.168.1.15: icmp_seq=22 ttl=64 time=0.416 ms  
64 bytes from 192.168.1.15: icmp_seq=23 ttl=64 time=0.901 ms  
64 bytes from 192.168.1.15: icmp_seq=24 ttl=64 time=217 ms  
From 192.168.1.11: icmp_seq=25 Redirect Host(New nexthop: 192.168.1.15)  
From 192.168.1.11 icmp_seq=26 Destination Host Unreachable  
[...]  
64 bytes from 192.168.1.15: icmp_seq=50 ttl=64 time=1961 ms  
64 bytes from 192.168.1.15: icmp_seq=51 ttl=64 time=975 ms
```

第 20 章

最佳远程控制方案：SSH

20.1 关于公钥认证

20.1.1 为什么要用公钥认证

```
$ grep sshd /var/log/auth.log.0
```

```
sshd[15738]: Failed password for root from x.x.x.x port 57087 ssh2
sshd[15740]: Address x.x.x.x maps to server.xxxxxx.com, but this does not map
back to the address - POSSIBLE BREAK-IN ATTEMPT!
sshd[15740]: (pam_unix) authentication failure; logname= uid=0 euid=0 tty=ssh
ruser= rhost=x.x.x.x user=root
```

20.2 SSH 的安装

20.2.1 安装 SSH 服务器和客户端

```
$ sudo apt-get install openssh-server
```

```
$ sudo apt-get install openssh-client
```

20.2.2 测试

```
$ ssh localhost
```

```
$ exit
```




20.3 SSH 配置

20.3.1 生成密钥对

```
$ ssh-keygen -t rsa -C "Hiweed's Key"
```

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/hiweed/.ssh/id_rsa): <-- 回车确认
```

```
Enter passphrase (empty for no passphrase): <-- 输入密码
```

```
Enter same passphrase again: <-- 再次输入密码
```

```
Your identification has been saved in /home/hiweed/.ssh/id_rsa.
```

```
Your public key has been saved in /home/hiweed/.ssh/id_rsa.pub.
```

```
The key fingerprint is:
```

```
fd:eb:41:f7:11:ec:7d:38:5d:e1:40:53:50:8b:0e:56 Hiweed's Key
```

20.3.2 将公钥复制到服务器

```
$ ssh-copy-id -i .ssh/id_rsa.pub hiweed@192.168.1.10
```

20.3.3 SSH 登录测试

1. 从 Linux 登录

```
$ ssh hiweed@192.168.1.10
```

```
$ ssh -i .ssh/id_rsa hiweed@192.168.1.10
```

20.3.4 SSH 服务器配置

```
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.back
```

```
$ sudo nano /etc/ssh/sshd_config
```

```
PermitRootLogin yes
```

```
PasswordAuthentication yes
```

```
UsePAM yes
```

```
PermitRootLogin no
```

```
PasswordAuthentication no
```

```
UsePAM no
```

```
$ sudo /etc/init.d/ssh restart
```

20.4 SSH 小技巧

20.4.1 用 scp 远程复制文件

1. 将本地文件复制到 SSH 服务器

```
$ scp file1 file2 [...] username@mytest.com:
```

2. 将 SSH 服务器文件复制到本地

```
$ scp username@mytest.com:/path/to/file .
```

3. 两台 SSH 服务器之间复制文件

```
$ scp username@mytest.com:/path/to/file username@yourname.org:/path/to/file
```

4. 复制所有文件

```
$ scp * username@mytest.com:/path/to/
```

```
$ scp -r username@mytest.com:/path/to/ .
```

5. 使用公钥

```
$ scp -i /path/to/private.key username@mytest.com:/path/to/ .
```

6. 限制带宽

```
$ scp -l 256 username@mytest.com:/path/to/file.tar.gz .
```

20.4.2 在客户端上指定命令

```
$ ssh mytest.com 'echo $PATH'
```

```
/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
```

```
$ ssh mytest.com env
```

```
Enter passphrase for key '/home/hiweed/.ssh/id_rsa':  
SHELL=/bin/bash  
SSH_CLIENT=192.168.1.12 47143 22  
USER=hiweed  
MAIL=/var/mail/hiweed  
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games  
PWD=/home/hiweed  
SHLVL=1  
HOME=/home/hiweed  
LOGNAME=hiweed
```



```
SSH_CONNECTION=192.168.1.12 47143 192.168.1.10 22
_=/usr/bin/env
```

20.4.3 在服务器上限制所执行的命令

```
command="/path/to/some/command args..." ssh-rsa .....(key).....
```

```
$ nano -w ~/.ssh/authorized_keys
```

```
command="ls / > files.list" ssh-rsa yHCBA8quGjcd1U9FXv/X19eSQQk4uLdw4eSqSfwV6m
G6ri
37Aha8k6dSJmtJ9OSFqnZYK6iXW5Iv1c2hG1lHYfK19zTMH00EMaAAAB3NzaC1yc2EAAAABIwAAAQEA3+
QaDcFzr30fO24pLg2UQOuLNRxYKFcEGd9J36Ubbp5gR2IcgHhIWtgjnlR8iaMWbS0mUiLQO5HqIOtRC3O
m+RRQQjDF6Xbk4CUiQ6V09QSAYZn2P6sjtiv4dl5lCXdBMgwwzBivzTw9RhLsqC44wtzJT/rA9C7Q71j
JpxRCvmcq/vBHQtIE8EKr6A1+Q3SWH3R+05zlyr5+xd8k085/1r5DNOKYSeBk/Ba2ibiyM+61SFG0aVw
== Hiweed's Key
```

```
$ ssh 192.168.1.10
```

```
$ cat files.list
```

20.4.4 修改密钥口令

```
$ ssh-keygen -p
```

```
Enter file in which the key is (/home/hiweed/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/hiweed/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

20.4.5 将密钥放入内存

```
$ ssh-agent $SHELL
$ ssh-add
```

```
Enter passphrase for /home/hiweed/.ssh/id_rsa:
Identity added: /home/hiweed/.ssh/id_rsa (/home/hiweed/.ssh/id_rsa)
```

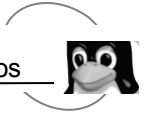
```
$ ssh-add -D
```

```
All identities removed.
```

```
$ exit
```


第 21 章

最佳服务器监控方案：Nagios



21.2 安装 Nagios

```
$ sudo apt-get install apache2
```

```
$ sudo apt-get install nagios2 nagios-plugins nagios-images
```

21.3 配置 Nagios

21.3.1 Nagios 初始化设置

1. 权限设置

```
$ sed 's/nagiosadmin/ng2admin/g' /etc/nagios2/cgi.cfg > /tmp/cgi.cfg  
$ sudo mv /tmp/cgi.cfg /etc/nagios2/cgi.cfg
```

```
$ sudo htpasswd -c /etc/nagios2/htpasswd.users ng2admin
```

2. 启用外部命令

```
$ sudo nano /etc/nagios2/nagios.cfg
```

```
[...]  
check_external_commands=1  
[...]
```

```
$ sudo /etc/init.d/nagios2 stop  
$ sudo dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios2/rw  
$ sudo dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios2  
$ sudo /etc/init.d/nagios2 start
```

21.3.2 Nagios 监控设置

```
$ sudo cp -r conf.d/ conf.d-back
```

1. 联系人设置

```
$ sudo nano /etc/nagios2/conf.d/contacts_nagios2.cfg
```

```
define contact{  
    contact_name          hiweed  
    alias                  Hiweed Leng  
    service_notification_period 24x7  
    host_notification_period 24x7  
    service_notification_options w,u,c,r
```

```
host_notification_options    d,r
service_notification_commands  notify-by-email
host_notification_commands    host-notify-by-email
email                        hiweed@hiweed.com
}

define contactgroup{
    contactgroup_name    admins
    alias                Nagios Administrators
    members              hiweed
}
```

2. 主机设置

```
define host {
    host_name    gateway
    alias        Default Gateway
    address      192.168.1.1
    use          generic-host
}
```

```
$ sudo nano /etc/nagios2/conf.d/hosts.cfg
```

```
define host{
    host_name    google
    alias        Internet Connection
    address      www.google.com
    use          generic-host
}

define host{
    host_name    mywangateway
    alias        ISP Gateway
    address      218.57.116.185
    parents      google
    use          generic-host
}

define host{
    host_name    mylangateway
    alias        My LAN Internet Gateway
    address      192.168.1.1
    parents      mywangateway
    use          generic-host
}

define host{
    host_name    webserver
    alias        Web Server
    address      192.168.1.12
    parents      mylangateway
    use          generic-host
}

define host{
    host_name    mailserver
    alias        Mail Server
    address      192.168.1.13
}
```



```
parents    mylangateway
use        generic-host
}
```

```
$ sudo nano /etc/nagios2/conf.d/extinfo_nagios2.cfg
```

```
[...]
define hostextinfo{
    host_name    gateway
    icon_image   base/ng-switch40.png
    statusmap_image base/ng-switch40.png
}
```

3. 主机组设置

```
define hostgroup {
    hostgroup_name all
    alias          All Servers
    members        *
}

define hostgroup {
    hostgroup_name ubuntu-servers
    alias          Ubuntu GNU/Linux Servers
    members        localhost, webmaster, mailserver
}

define hostgroup {
    hostgroup_name http-servers
    alias          HTTP servers
    members        localhost, webserver
}

define hostgroup {
    hostgroup_name ssh-servers
    alias          SSH servers
    members        localhost, webserver, mailserver
}

define hostgroup {
    hostgroup_name mailservers
    alias          Mail servers
    members        localhost, mailserver
}

define hostgroup {
    hostgroup_name ping-servers
    alias          Pingable servers
    members        gateway
}
```

```
$ sudo nano /etc/nagios2/conf.d/extinfo_nagios2.cfg
```

```
define hostextinfo{
    hostgroup_name  ubuntu-servers
    notes           Ubuntu GNU/Linux servers
    icon_image      base/debian.png
    icon_image_alt  Ubuntu GNU/Linux
    vrml_image      debian.png
}
```



```
statusmap_image base/debian.gd2
}
```

[...]

```
$ sudo /etc/init.d/nagios2 start
```

```
* Starting nagios2 monitoring daemon nagios2 [ OK ]
```



4. 服务设置

```
$ sudo nano /etc/nagios2/conf.d/services_nagios2.cfg
```

```
define service {
    hostgroup_name      http-servers
    service_description  HTTP
    check_command        check_http
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      ssh-servers
    service_description  SSH
    check_command        check_ssh
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      ping-servers
    service_description  PING
    check_command        check_ping!100.0,20%!500.0,60%
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      mailservers
    service_description  POP
    check_command        check_pop
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      mailservers
    service_description  IMAP
    check_command        check_imap
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      mailservers
    service_description  Secure POP
    check_command        check_spop
    use                  generic-service
    notification_interval 0
}

define service {
    hostgroup_name      mailservers
    service_description  Secure IMAP
    check_command        check_simap
}
```

```
use generic-service
notification_interval 0
}

define service {
    hostgroup_name mailservers
    service_description SMTP
    check_command check_smtp
    use generic-service
    notification_interval 0
}
```

21.5 Nagios 排错

```
$ sudo /etc/init.d/nagios2 restart
```

```
* Restarting nagios2 monitoring daemon nagios2
* already running!
```

[fail]

```
$ sudo /etc/init.d/nagios2 stop
```

```
* Stopping nagios2 monitoring daemon nagios2
```

```
$ sudo /etc/init.d/nagios2 start
```

```
[...]
```

```
Error: Could not find any hostgroup matching 'debian-servers'
```

```
Error: Could not expand hostgroups and/or hosts specified in extended host info
(config file '/etc/nagios2/conf.d/extinfo_nagios2.cfg', starting on line 5)
```

```
[...]
```

第 22 章

最佳 RAID 方案：RAID10

22.2 RAID10 的实现

22.2.3 分区复制

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
# sfdisk -d /dev/sda | sfdisk /dev/sdc
# sfdisk -d /dev/sda | sfdisk /dev/sdd
```

22.2.4 创建 RAID 阵列

```
# mdadm --create /dev/md0 --auto=yes --force -R --level=raid1 --raid-devices=4
/ dev/sd[a-d]1
```

```
# mdadm --create /dev/md1 --auto=yes --force -R --level=raid10 --raid-devices=4
/d ev/sd[a-d]2
```

```
# mdadm --create /dev/md2 --auto=yes --force -R --level=raid10 --raid-devices=4
/d ev/sd[a-d]3
```

22.3 RAID10 的日常维护

22.3.2 mdadm 的选项

5. 其他选项

```
$ man mdadm
```

22.3.3 创建 RAID 阵列

```
$ sudo mdadm --create /dev/md1 --auto=yes --force -R --level=raid10 --raid-
devices =4 /dev/sd[a-d]2
```



```
$ sudo mdadm --create /dev/md1 --auto=yes --force -R --level=raid10 --raid-devices=4 /dev/sd[a-d]2 --hot-spares=1 /dev/sde2
```

22.3.4 查询 RAID 阵列

```
$ cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md2 : active raid10 sda3[0] sdd3[3] sdc3[2] sdb3[1]
      465664 blocks 64K chunks 2 near-copies [4/4] [UUUU]

md1 : active raid10 sda2[0] sdd2[3] sdc2[2] sdb2[1]
      16209408 blocks 64K chunks 2 near-copies [4/4] [UUUU]

md0 : active raid1 sda1[0] sdd1[3] sdc1[2] sdb1[1]
      48064 blocks [4/4] [UUUU]

unused devices: <none>
```

```
$ sudo mdadm --detail /dev/md0
```

```
$ sudo mdadm --examine /dev/sda2
```

```
$ sudo mdadm --examine /dev/sdb*
```

22.3.5 RAID 的监控

```
$ sudo nano /etc/mdadm/mdadm.conf
```

```
[...]
MAILADDR hiweed@hiweed.com
[...]
```

22.3.6 RAID 的启动/停止

```
$ sudo mdadm -A /dev/md0
```

```
$ sudo mdadm --stop /dev/md0
```

```
mdadm: fail to stop array /dev/md0: Device or resource busy
```

```
$ sudo umount /boot
$ sudo mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

```
$ sudo mdadm -A /dev/md0
mdadm: /dev/md0 has been started with 4 drives.
```

22.4 故障处理

22.4.1 从 RAID 中移除设备

1. 移除单个 RAID 物理卷

```
$ sudo mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
mdadm: set /dev/sda1 faulty in /dev/md0
mdadm: hot removed /dev/sda1
```

```
$ sudo mdadm --zero-superblock /dev/sda1
```

2. 移除整个硬盘

```
$ sudo mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
mdadm: set /dev/sda1 faulty in /dev/md0
mdadm: hot removed /dev/sda1
```

```
$ sudo mdadm /dev/md1 --fail /dev/sda2 --remove /dev/sda2
mdadm: set /dev/sda2 faulty in /dev/md1
mdadm: hot removed /dev/sda2
```

```
$ sudo mdadm /dev/md2 --fail /dev/sda3 --remove /dev/sda3
```



```
mdadm: set /dev/sda3 faulty in /dev/md2
mdadm: hot removed /dev/sda3
```

22.4.2 添加已有 RAID 物理卷

```
$ sudo mdadm /dev/md0 --add /dev/sda1
```

```
mdadm: re-added /dev/sda1
```

```
$ sudo mdadm /dev/md1 --add /dev/sda2
```

```
mdadm: re-added /dev/sda1
```

```
$ sudo mdadm /dev/md2 --add /dev/sda3
```

```
mdadm: re-added /dev/sda1
```

22.4.3 更换全新硬盘

1. 移除坏硬盘

```
$ sudo mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
```

```
$ sudo mdadm /dev/md1 --fail /dev/sda2 --remove /dev/sda2
```

```
$ sudo mdadm /dev/md2 --fail /dev/sda3 --remove /dev/sda3
```

```
$ cat /proc/mdstat
```

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
[raid10]
```

```
md2 : active raid10 sdd3[3] sdc3[2] sdb3[1]
      465664 blocks 64K chunks 2 near-copies [4/3] [_UUU]
```

```
md1 : active raid10 sdd2[3] sdc2[2] sdb2[1]
      16209408 blocks 64K chunks 2 near-copies [4/3] [_UUU]
```

```
md0 : active raid1 sdd1[3] sdc1[2] sdb1[1]
      48064 blocks [4/3] [_UUU]
```

```
unused devices: <none>
```

2. 插入硬盘

```
$ sudo halt
```

```
$ sudo fdisk -l
```

3. 新硬盘分区

```
$ sudo sfdisk -d /dev/sda | sudo sfdisk /dev/sdd
```

4. 将新分区加入 RAID

```
$ sudo mdadm --detail /dev/md1
```

```
[...]  
Active Devices : 3  
Working Devices : 3  
[...]  
   Number   Major   Minor   RaidDevice State  
     0         0         0         0    removed  
     1         8          2         1    active sync  /dev/sda2  
     2         8        18         2    active sync  /dev/sdb2  
     3         8        34         3    active sync  /dev/sdc2
```

```
$ sudo mdadm /dev/md1 --add /dev/sdd2
```

```
$ sudo mdadm --detail /dev/md1
```

```
[...]  
Rebuild Status : 7% complete  
[...]  
   Number   Major   Minor   RaidDevice State  
     4         8        50          0    spare rebuilding  /dev/sdd2  
[...]
```

```
$ sudo mdadm --detail /dev/md1
```

```
Active Devices : 4  
Working Devices : 4  
[...]  
   Number   Major   Minor   RaidDevice State  
     0         8        50          0    active sync  /dev/sdd2  
     1         8          2         1    active sync  /dev/sda2  
     2         8        18         2    active sync  /dev/sdb2  
     3         8        34         3    active sync  /dev/sdc2
```

```
$ sudo mdadm /dev/md0 --add /dev/sdd1
```




```
$ sudo mdadm /dev/md2 --add /dev/sdd3
```

5. 设置 grub

```
$ sudo grub
grub> root (hd3,0)
grub> setup (hd3)
grub> quit
```

22.5 添加备用硬盘

```
$ sudo mdadm --detail /dev/md1 | grep Spare
Spare Devices : 0
```

22.5.1 插入新硬盘

```
$ sudo fdisk -l
```

22.5.2 新硬盘分区

```
$ sudo sfdisk -d /dev/sda | sudo sfdisk /dev/sde
```

22.5.3 将新分区加入 RAID

```
$ sudo mdadm /dev/md0 --add /dev/sde1
$ sudo mdadm /dev/md1 --add /dev/sde2
$ sudo mdadm /dev/md2 --add /dev/sde3
```

```
$ sudo mdadm --detail /dev/md0
Total Devices : 5
[...]
Working Devices : 5
Failed Devices : 0
Spare Devices : 1
[...]
   Number   Major   Minor   RaidDevice State
[...]
     4         8      65         -    spare   /dev/sde1
```

22.5.4 设置 grub

```
$ sudo grub
grub> root (hd4,0)
grub> setup (hd4)
grub> quit
```

22.5.5 故障模拟

```
$ sudo mdadm /dev/md0 --fail /dev/sda1
```

```
mdadm: set /dev/sda1 faulty in /dev/md0
```

```
$ sudo mdadm --detail /dev/md0
```

```
[...]
```

```
Failed Devices : 1
```

```
Spare Devices : 0
```

```
UUID : 858ea4ab:1c224ab8:b41c754c:f8475f43
```

```
Events : 0.80
```

Number	Major	Minor	RaidDevice	State		
0	8	65	0	active	sync	/dev/sde1
1	8	17	1	active	sync	/dev/sdb1
2	8	33	2	active	sync	/dev/sdc1
3	8	49	3	active	sync	/dev/sdd1
4	8	1	-	faulty	spare	/dev/sda1

```
$ sudo mdadm /dev/md0 --remove /dev/sda1
```

```
$ sudo mdadm /dev/md0 --add /dev/sda1
```

第 23 章

最佳数据安全方案： RAID10+LVM

23.1 创建 RAID 物理卷

23.1.2 剩余硬盘的分区处理

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
# sfdisk -d /dev/sda | sfdisk /dev/sdc
# sfdisk -d /dev/sda | sfdisk /dev/sdd
```

23.2 创建 RAID 阵列

23.2.1 创建 RAID1 阵列

```
# mdadm --create /dev/md0 --auto=yes --force -R --level=raid1 --raid-devices=4
/ dev/sd[a-d]1
```

23.2.2 创建 RAID10 阵列

```
# mdadm --create /dev/md1 --auto=yes --force -R --level=raid10 --raid-devices=4
/d ev/sd[a-d]2
```



23.5 LVM 的相关命令

23.5.1 LVM 物理卷相关命令

1. 显示 LVM 物理卷信息

```
$ sudo pvdisplay
--- Physical volume ---
PV Name           /dev/md1
VG Name           ubox_lvm
PV Size           15.90 GB / not usable 512.00 KB
Allocatable       yes (but full)
PE Size (KByte)   4096
Total PE          4071
Free PE           0
Allocated PE       4071
PV UUID           E3MQdl-NwhH-zJUo-hSX1-J5uY-44b1-bYgV9P
```

2. 创建新的 LVM 物理卷

```
$ sudo pvcreate /dev/sde1 /dev/sdf
Physical volume "/dev/sde1" successfully created
Physical volume "/dev/sdf" successfully created
```

3. LVM 物理卷的扫描

```
$ sudo pvscan
PV /dev/md1   VG ubox_lvm   lvm2 [15.90 GB / 0   free]
Total: 1 [15.90 GB] / in use: 1 [15.90 GB] / in no VG: 0 [0   ]
```

```
$ sudo pvs
PV          VG          Fmt Attr PSize   PFree
/dev/md1    ubox_lvm   lvm2 a-   15.90G    0
```

4. LVM 物理卷容量的修改

```
$ sudo pvresize /dev/sda1
```

```
$ sudo pvresize --setphysicalvolumesize 30G /dev/sda1
```

5. LVM 物理卷的移动

```
$ sudo pvmove -v /dev/sda4
```

6. LVM 物理卷的删除

```
$ sudo pvremove /dev/sda4
```

23.5.2 LVM 卷组相关命令

1. 创建 LVM 卷组

```
$ sudo vgcreate ubuntu_vg /dev/sdk1 /dev/sdl1
```

2. 查询 LVM 卷组

```
$ sudo vgdisplay
```

```
--- Volume group ---
VG Name                ubox_lvm
System ID
Format                 lvm2
[...]
VG Size                15.90 GB
PE Size                4.00 MB
Total PE               4071
Alloc PE / Size        4071 / 15.90 GB
Free PE / Size         0 / 0
VG UUID                IZTYxO-L008-D39n-s33P-UAgJ-RYww-PKISqE
```

3. LVM 卷组扫描

```
$ sudo vgs
```

VG	#PV	#LV	#SN	Attr	VSize	VFree
ubox_lvm	1	5	0	wz--n-	15.90G	0

```
$ sudo vgscan
```

```
Reading all physical volumes. This may take a while...
Found volume group "ubox_lvm" using metadata type lvm2
```

4. LVM 卷组改名

```
$ sudo vgrename /dev/ubox_lvm /dev/my_volume_group
```

```
$ sudo vgrename ubox_lvm my_volume_group
```

5. LVM 卷组的拆分、合并

```
$ sudo vgsplit -v databases_vg new_vg /dev/sdk1 /dev/sdn1
```

```
$ sudo vgmerge -v databases_vg my_vg
```

6. LVM 卷组的导出、导入

```
$ sudo vgchange -an vg02
```



```
$ sudo vgexport vg02
```

```
$ sudo vgimport vg02 /dev/sd[b-h]5
```

7. LVM 卷组的扩容、收缩

```
$ sudo vgextend vg00 /dev/sda4 /dev/sdn1
```

```
$ sudo vgreduce vg00 /dev/sda4 /dev/sdn1
```

8. LVM 卷组属性修改

```
$ sudo vgchange -an vg02
```

```
$ sudo vgchange -a y
```

```
$ sudo vgchange -l 128 /dev/vg00
```

9. LVM 卷组 meta 数据的备份、恢复

```
$ sudo vgcfgbackup ubox_lvm
```

```
$ sudo vgcfgbackup
```

```
$ sudo vgcfgrestore --file /etc/lvm/backup/vg02 vg02
```

23.5.3 LVM 逻辑卷相关命令

1. 创建 LVM 逻辑卷

```
$ sudo lvcreate --name share_lv --size 40G vg00
```

2. 查询 LVM 逻辑卷

```
$ sudo lvdisplay
```

```
--- Logical volume ---  
LV Name                /dev/ubox_lvm/ubox_home
```

```
VG Name          ubox_lvm
LV UUID          oIDBUy-42e3-fxtC-VCei-yrQi-KKus-tKNpTh
LV Write Access  read/write
LV Status        available
# open           1
LV Size          2.00 GB
Current LE       512
Segments         1
Allocation       inherit
Read ahead sectors 0
Block device     254:0
[...]
```

3. LVM 逻辑卷扫描

```
$ sudo lvs
```

LV	VG	Attr	LSize	Origin	Snap%	Move	Log	Copy%
ubox_home	ubox_lvm	-wi-ao	2.00G					
ubox_root	ubox_lvm	-wi-ao	4.00G					
ubox_swap	ubox_lvm	-wi-ao	256.00M					
ubox_tmp	ubox_lvm	-wi-ao	1.00G					
ubox_var	ubox_lvm	-wi-ao	8.65G					

```
$ sudo lvscan
```

ACTIVE	['/dev/ubox_lvm/ubox_home']	[2.00 GB]	inherit
ACTIVE	['/dev/ubox_lvm/ubox_root']	[4.00 GB]	inherit
ACTIVE	['/dev/ubox_lvm/ubox_tmp']	[1.00 GB]	inherit
ACTIVE	['/dev/ubox_lvm/ubox_swap']	[256.00 MB]	inherit
ACTIVE	['/dev/ubox_lvm/ubox_var']	[8.65 GB]	inherit

4. LVM 逻辑卷改名

```
$ sudo lvrename /dev/ubox_lvm/ubox_home /dev/ubox_lvm/ubox_opt
```

```
$ sudo lvrename ubox_lvm ubox_home ubox_opt
```

5. LVM 逻辑卷的扩容

```
$ sudo umount /home
```

```
$ sudo lvextend -L10G /dev/ubox_lvm/ubox_home
```

```
$ sudo e2fsck -f /dev/ubox_lvm/ubox_home
```

```
$ sudo resize2fs /dev/ubox_lvm/ubox_home
```



6. LVM 逻辑卷的收缩

```
$ sudo resize2fs /dev/ubox_lvm/ubox_home 5G
```

```
$ sudo lvreduce -L5G /dev/ubox_lvm/ubox_home
```

7. LVM 逻辑卷属性修改

```
$ sudo lvchange -an /dev/ubox_lvm/ubox_home
```

```
$ sudo lvchange -ay
```

```
$ sudo lvchange -pr /dev/ubox_lvm/ubox_home
```

23.6 添加新硬盘

23.6.1 插入新硬盘

```
$ sudo fdisk -l
```

23.6.2 配置 RAID

1. 创建 RAID 物理卷

```
$ sudo cfdisk /dev/sde
```

```
cfdisk (util-linux-ng 2.13.1)
```

```
Disk Drive: /dev/sde
```

```
Size: 8589934592 bytes, 8589 MB
```

```
Heads: 255 Sectors per Track: 63 Cylinders: 1044
```

Name	Flags	Part Type	FS Type	[Label]	Size (MB)

		Pri/Log	Free Space		8587.20

```
[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]
[ Write ]
```


Print help screen

```
$ sudo sfdisk -d /dev/sde | sudo sfdisk /dev/sdf
```

2. 创建 RAID1 阵列

```
$ sudo mdadm --create /dev/md2 --auto=yes --force -R --level=raid1 --raid-  
devices= 2 /dev/sd[e-f]1
```

```
mdadm: array /dev/md2 started.
```

```
$ sudo mdadm --detail /dev/md2
```

```
/dev/md2:
```

```
[...]
```

```
    Raid Level : raid1
```

```
[...]
```

```
    Active Devices : 2
```

```
    Working Devices : 2
```

```
[...]
```

Number	Major	Minor	RaidDevice	State	
0	8	65	0	active sync	/dev/sde1
1	8	81	1	active sync	/dev/sdf1

23.6.3 在 RAID 上配置 LVM

1. 创建 LVM 物理卷

```
$ sudo pvcreate /dev/md2
```

```
Physical volume "/dev/md2" successfully created
```

```
$ sudo pvdisplay /dev/md2
```

```
--- NEW Physical volume ---
```

```
PV Name          /dev/md2
```

```
VG Name
```

```
[...]
```

2. 扩容现有 LVM 卷组

```
$ sudo vgextend ubox_lvm /dev/md2
```

```
Volume group "ubox_lvm" successfully extended
```

```
$ sudo pvdisplay
```

```
--- Physical volume ---
```

```
PV Name          /dev/md2
```

```
VG Name          ubox_lvm
```



3. 扩容现有 LVM 逻辑卷

```
$ sudo lvdisplay /dev/ubox_lvm/ubox_var | grep Size
```

```
LV Size                8.65 GB
```

```
$ sudo lvextend -L+7.9G /dev/ubox_lvm/ubox_var
```

```
Rounding up size to full physical extent 7.90 GB
Extending logical volume ubox_var to 16.55 GB
Logical volume ubox_var successfully resized
```

```
$ sudo lvdisplay /dev/ubox_lvm/ubox_var | grep Size
```

```
LV Size                16.55 GB
```

23.6.4 扩容文件系统

```
$ df -h | grep var
```

```
[...]
/dev/mapper/ubox_lvm-ubox_var
      8.7G  129M  8.6G   2% /var
```

```
$ sudo xfs_growfs /var
```

```
[...]
data blocks changed from 2268160 to 4339712
```

```
$ df -h | grep var
```

```
[...]
/dev/mapper/ubox_lvm-ubox_var
      17G  130M   17G   1% /var
```

23.8 LVM 分区备份

23.8.1 创建快照

```
$ sudo lvcreate -s -L90M -n var_snapshot /dev/ubox_lvm/ubox_var
```

```
Rounding up size to full physical extent 92.00 MB
Logical volume "var_snapshot" created
```

```
$ sudo lvm lvscan
```

```
ACTIVE                '/dev/ubox_lvm/ubox_home' [2.00 GB] inherit
ACTIVE                '/dev/ubox_lvm/ubox_root' [4.00 GB] inherit
```

```
ACTIVE      '/dev/ubox_lvm/ubox_tmp' [1.00 GB] inherit
ACTIVE      '/dev/ubox_lvm/ubox_swap' [256.00 MB] inherit
ACTIVE      Original '/dev/ubox_lvm/ubox_var' [16.55 GB] inherit
ACTIVE      Snapshot '/dev/ubox_lvm/var_snapshot' [92.00 MB] inherit
```

23.8.2 备份快照内容

```
$ sudo mount /dev/ubox_lvm/var_snapshot /mnt -o nouuid
```

```
$ ls /mnt
```

```
backups cache lib local lock log mail opt run spool tmp
```

```
$ sudo tar cfz /tmp/var.tar.gz /mnt
```

```
$ sudo umount /mnt
```

23.8.3 删除快照

```
$ sudo lvremove /dev/ubox_lvm/var_snapshot
```

```
Do you really want to remove active logical volume "var_snapshot"? [y/n]: y
Logical volume "var_snapshot" successfully removed
```

第 24 章

Ubuntu Server 系统安全

24.1 系统安全更新

24.1.1 订阅安全列表

```
$ sudo apt-get update && apt-get upgrade
```

24.1.2 自动更新

```
$ sudo nano /etc/apt/apt.conf.d/10periodic
```

```
APT::Periodic::Update-Package-Lists "1";  
APT::Periodic::Download-Upgradeable-Packages "1";  
APT::Periodic::AutocleanInterval "0";  
APT::Periodic::Unattended-Upgrade "1";
```

```
$ sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
```

```
Unattended-Upgrade::Allowed-Origins {  
    "ubuntu hardy-security";  
};
```

24.2 控制台安全

```
$ sudo nano /etc/event.d/control-alt-delete
```

```
[...]  
#exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

24.3 用户、密码管理

24.3.1 关于 root 用户

```
$ sudo passwd
```

```
[sudo] password for hiweed: <-- 你当前用户的密码  
Enter new UNIX password: <-- root 的密码  
Retype new UNIX password: <-- 再输入一遍 root 的密码  
passwd: password updated successfully
```

```
$ sudo passwd -l root
```

```
Password changed.
```

24.3.3 关于/etc/sudoers

用户名 主机名 = [(目的用户)] [NOPASSWD:] 命令列表

```
$ man sudoers
```

1. 指定运行命令的身份

```
hiweed ubox = (operator) /bin/ls, /bin/kill, /usr/bin/lprm
```

```
$ sudo -u operator /bin/ls
```

```
hiweed ubox = (operator) /bin/ls, (root) /bin/kill, /usr/bin/lprm
```

```
hiweed ubox = (:dailer) /usr/bin/tip, /usr/bin/cu
```

2. 有密码/无密码

```
hiweed ubox = NOPASSWD: /bin/ls, /bin/kill, /usr/bin/lprm
```

```
hiweed ubox = NOPASSWD: /bin/ls, PASSWD: /bin/kill, /usr/bin/lprm
```

```
hiweed ubox = NOPASSWD: ALL
```



24.3.4 密码策略

1. 密码长度设置

```
$ sudo nano /etc/pam.d/common-password
```

```
password requisite pam_unix.so nullok obscure md5
```

```
password requisite pam_unix.so nullok obscure md5 min=8
```

2. 密码有效期

```
$ sudo chage -l hiweed
```

```
Last password change           : Mar 14, 2009
Password expires                : never
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
```

```
$ sudo chage hiweed
```

```
Changing the aging information for hiweed
Enter the new value, or press ENTER for the default
```

```
Minimum Password Age [0]:           <-- 最短有效期
Maximum Password Age [99999]:       <-- 最长有效期
Last Password Change (YYYY-MM-DD) [2009-03-14]: <-- 密码最后修改日期
Password Expiration Warning [7]:    <-- 密码过期提前警告期
Password Inactive [-1]:             <-- 过期后密码是否锁定
Account Expiration Date (YYYY-MM-DD) [1969-12-31]: <-- 账号过期日
```

```
$ sudo chage -M 90 -W 14 -I 5 hiweed
```

```
$ sudo chage -E 2010-12-31 hiweed
```

24.4 ufw 防火墙

24.4.1 启用、禁用 ufw

```
$ sudo ufw status
Firewall not loaded
```

```
$ sudo enable ufw
Firewall started and enabled on system startup
```

```
$ sudo ufw status
Firewall loaded
```

```
$ sudo ufw disable
Firewall stopped and disabled on system startup
```

要启用 ufw 日志，运行命令：

```
$ sudo ufw logging on
Logging enabled
```

```
$ sudo ufw logging off
Logging disabled
```

24.4.2 基本规则设置

1. 开放端口

```
$ sudo ufw allow 53
```

```
$ sudo ufw allow 53/tcp
```

```
$ sudo ufw allow 53/udp
```

2. 关闭端口

```
$ sudo ufw deny 53
```



```
$ sudo ufw deny 53/tcp
```

```
$ sudo ufw deny 53/udp
```

3. 以服务名代替端口号

```
$ sudo ufw deny ssh  
$ sudo ufw allow ssh
```

```
$ less /etc/services
```

4. 删除规则

```
$ sudo ufw deny 53/udp
```

```
$ sudo ufw delete deny 53/udp
```

24.4.3 常用规则设置

1. 允许某个 IP 访问

```
$ sudo ufw allow from 10.10.100.100
```

2. 禁止某个 IP 访问

```
$ sudo ufw deny from 10.10.100.100
```

3. 允许某个网段访问

```
$ sudo ufw allow from 10.10.100.0/24
```

4. 禁止某个网段访问

```
$ sudo ufw deny from 10.10.100.0/24
```

5. 允许某 IP 访问某个端口

```
$ sudo ufw allow from 192.168.1.4 to any port 22
```

6. 禁止某 IP 访问某个端口

```
$ sudo ufw deny from 192.168.1.4 to any port 22
```


7. 禁止 ping

```
$ sudo nano /etc/ufw/before.rules
```

```
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

```
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

```
$ sudo /etc/init.d/ufw force-reload
```

```
* Stopping firewall: ufw... [ OK ]  
* Starting firewall: ufw... [ OK ]
```

24.4.4 高级规则设置

1. 挡掉某个 IP 地址

```
$ sudo ufw allow 80
```

```
$ sudo ufw deny 111.222.33.44
```

```
$ sudo nano /etc/ufw/before.rules
```

```
[...]  
# drop INVALID packets  
# uncomment to log INVALID packets  
#-A ufw-before-input -m conntrack --ctstate INVALID -j LOG --log-prefix "[UFW  
BLOCK INVALID]: "  
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP  
  
# Block IPs  
-A ufw-before-input -s 111.222.33.44 -j DROP  
[...]
```

2. 控制子网中的个别主机 (1)

```
$ sudo ufw deny from 192.168.1.1 to any port 22
```

```
Rule added
```

```
$ sudo ufw deny from 192.168.1.20 to any port 22
```

```
Rule added
```

```
$ sudo ufw allow from 192.168.1.0/24 to any port 22
```



```
Rule added
```

```
$ sudo ufw status
```

```
Firewall loaded
```

To	Action	From
--	-----	----
22:tcp	DENY	192.168.1.1
22:udp	DENY	192.168.1.1
22:tcp	DENY	192.168.1.20
22:udp	DENY	192.168.1.20
22:tcp	ALLOW	192.168.1.0/24
22:udp	ALLOW	192.168.1.0/24

3. 控制子网中的个别主机 (2)

```
$ sudo ufw delete allow from 192.168.1.0/24 to any port 22
```

```
Rule deleted
```

```
$ sudo ufw status
```

```
Firewall loaded
```

To	Action	From
--	-----	----
53:tcp	ALLOW	Anywhere
53:udp	ALLOW	Anywhere
22:tcp	DENY	192.168.1.1
22:udp	DENY	192.168.1.1
22:tcp	DENY	192.168.1.20
22:udp	DENY	192.168.1.20

```
$ sudo ufw deny from 192.168.1.9 to any port 22
```

```
Rule added
```

```
$ sudo ufw allow from 192.168.1.0/24 to any port 22
```

```
Rule added
```

```
$ sudo ufw status
```

```
Firewall loaded
```

To	Action	From
--	-----	----
53:tcp	ALLOW	Anywhere
53:udp	ALLOW	Anywhere
22:tcp	DENY	192.168.1.1

```
22:udp      DENY  192.168.1.1
22:tcp      DENY  192.168.1.20
22:udp      DENY  192.168.1.20
22:tcp      DENY  192.168.1.9
22:udp      DENY  192.168.1.9
22:tcp      ALLOW 192.168.1.0/24
22:udp      ALLOW 192.168.1.0/24
```

24.4.5 IP 伪装

1. 启用包转发

```
$ sudo nano /etc/default/ufw
```

```
DEFAULT_FORWARD_POLICY="DROP"
```

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

```
$ sudo nano /etc/ufw/sysctl.conf
```

```
net/ipv4/ip_forward=1
```

```
net/ipv6/conf/default/forwarding=1
```

2. 添加规则

```
$ sudo nano /etc/ufw/before.rules
```

```
# nat 规则
*nat
:POSTROUTING ACCEPT [0:0]

# 将来自 eth1 的数据包转发给 eth0
-A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE

# 不要删掉该 COMMIT 行, 否则 nat 规则不会生效
COMMIT
```

```
$ sudo /etc/init.d/ufw restart
```

```
* Stopping firewall: ufw... [ OK ]
* Starting firewall: ufw... [ OK ]
```



24.5 入侵检测

24.5.1 安装 LAMP

```
$ sudo apt-get install mysql-server libapache2-mod-php5 php5-mysql libphp-  
adodb
```

24.5.2 安装、配置 Snort

1. 安装 Snort

```
$ sudo apt-get install snort-mysql
```

```
Setting up snort-mysql (2.7.0-14) ...  
* Stopping Network Intrusion Detection System snort  
* No running snort instance found  
* Starting Network Intrusion Detection System snort  
* /etc/snort/db-pending-config file found  
* Snort will not start as its database is not yet configured.  
* Please configure the database as described in  
* /usr/share/doc/snort-{pgsql,mysql}/README-database.Debian  
* and remove /etc/snort/db-pending-config  
invoke-rc.d: initscript snort, action "start" failed.  
dpkg: error processing snort-mysql (--configure):  
 subprocess post-installation script returned error exit status 6  
Processing triggers for libc6 ...  
ldconfig deferred processing now taking place  
Errors were encountered while processing:  
 snort-mysql  
E: Sub-process /usr/bin/dpkg returned an error code (1)
```

```
$ less /usr/share/doc/snort-mysql/README-database.Debian  
$ zless /usr/share/doc/snort-mysql/README.database.gz
```

2. 为 Snort 创建数据库

```
$ mysql -uroot -p
```

```
mysql> CREATE DATABASE snortdb;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> grant CREATE, INSERT, SELECT, UPDATE on snortdb.* to snort@localhost;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SET PASSWORD FOR snort@localhost=PASSWORD('snortPassword');
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> exit
```

```
Bye
```

```
$ cd /usr/share/doc/snort-mysql
$ zcat create_mysql.gz | mysql snortdb -u snort -psnortPassword
```

```
$ sudo rm /etc/snort/db-pending-config
```

3. 配置 Snort

```
$ sudo nano /etc/snort/snort.conf
```

```
#var HOME_NET any
var HOME_NET 192.168.1.0/24
```

```
#var EXTERNAL_NET any
var EXTERNAL_NET !$HOME_NET
```

```
#output database: log, mysql,
output database: log, mysql, user=snort password=snortPassword dbname=snortdb
host=localhost
```

```
$ sudo chown snort /var/log/snort/alert
```

```
$ sudo snort -c /etc/snort/snort.conf
```

```
[...]
==== Initialization Complete ===

,,_      -*> Snort! <*-
o" )~    Version 2.7.0 (Build 35)
'''      By Martin Roesch & The Snort Team: http://www.snort.org/team.html
          (C) Copyright 1998-2007 Sourcefire Inc., et al.

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 1.6 <Build 11>
Preprocessor Object: SF_FTPTELNET Version 1.0 <Build 10>
Preprocessor Object: SF_DNS Version 1.0 <Build 2>
Preprocessor Object: SF_DCERPC Version 1.0 <Build 4>
Preprocessor Object: SF_SMTP Version 1.0 <Build 7>
Preprocessor Object: SF_SSH Version 1.0 <Build 1>
Not Using PCAP_FRAMES
```



```
$ sudo /etc/init.d/snort start
```

```
* Starting Network Intrusion Detection System snort [ OK ]
```

```
$ ps aux|grep snort
```

```
snort    7166  122  24.7  88572 63324 ?        Rs   04:04   0:02 /usr/sbin/snort -m  
02 7 -D -d -l /var/log/snort -u snort -g snort -c /etc/snort/snort.conf -S  
HOME_NET=[1 92.168.1.0/24] -i eth0
```

```
$ cat /var/log/syslog | grep snort | grep ERROR
```

```
Apr  4 03:52:49 ubuntu snort[7032]: FATAL ERROR: OpenAlertFile() => fopen() alert  
f ile /var/log/snort/alert: Permission denied  
Apr  4 04:01:12 ubuntu snort[7141]: FATAL ERROR: database: mysql_error: Access  
deni ed for user 'snort'@'localhost' (using password: YES)
```

24.5.3 安装、配置 BASE

1. 安装 acidbase 软件包

```
$ sudo apt-get install acidbase
```

2. 配置 Apache

```
$ sudo nano /etc/apache2/sites-available/default
```

```
Alias /acidbase "/usr/share/acidbase"
```

```
<Directory /usr/share/acidbase/>  
    Options +FollowSymLinks  
    AllowOverride None  
    Order allow,deny  
    allow from all  
</Directory>
```

```
$ sudo /etc/init.d/apache2 reload
```

```
* Reloading web server config apache2 [ ok ]
```

3. 配置 BASE

```
$ sudo mv /etc/acidbase/base_conf.php /etc/acidbase/base_conf.php-orig
```

```
$ sudo nano /etc/acidbase/base_conf.php
```

```
// $BASE_path = dirname(__FILE__);  
$BASE_path = "/usr/share/acidbase";
```

```
$ sudo chmod o= /etc/acidbase/base_conf.php  
$ sudo chgrp www-data /etc/acidbase/base_conf.php
```

24.6 肉鸡检测

24.6.1 chkrootkit 的使用

```
$ sudo apt-get install chkrootkit
```

```
$ sudo chkrootkit ps ls cron
```

```
ROOTDIR is '/'  
Checking `ps'... not infected  
Checking `ls'... not infected
```



```
Checking `cron'... not infected
```

```
$ sudo chkrootkit
```

```
$ sudo chkrootkit -q
```

```
$ sudo chkrootkit -x su | less
```

```
ROOTDIR is '/'  
###  
### Output of: /usr/bin/strings -a /bin/su  
###  
/lib/ld-linux.so.2  
libcrypt.so.1  
__gmon_start__  
_Jv_RegisterClasses  
libpam.so.0  
pam_start  
[...]
```

```
$ sudo chkrootkit -p /cdrom/bin:/cdrom/sbin:/cdrom/usr/bin
```

```
$ sudo chkrootkit -r /mnt/yourharddisk/
```

24.6.2 rkhunter 的使用

1. 安装 rkhunter

```
$ sudo apt-get install rkhunter postfix mailx
```

```
$ sudo rkhunter --update
```

```
[ Rootkit Hunter version 1.3.0 ]  
  
Checking rkhunter data files...  
  Checking file mirrors.dat [ No update ]  
  Checking file programs_bad.dat [ No update ]  
  Checking file backdoorports.dat [ No update ]  
  Checking file suspscan.dat [ No update ]  
  Checking file i18n/cn [ Updated ]  
  Checking file i18n/en [ Updated ]  
  Checking file i18n/zh [ Updated ]
```


Checking file i18n/zhutf

[Updated]

```
$ sudo rkhunter --list
```

```
$ sudo rkhunter -c
```

```
[ Rootkit Hunter version 1.3.0 ]
```

```
Checking system commands...
```

```
Performing 'strings' command checks
```

```
Checking 'strings' command
```

```
[ OK ]
```

```
Performing 'shared libraries' checks
```

```
Checking for preloading variables
```

```
[ None found ]
```

```
Checking for preload file
```

```
[ Not found ]
```

```
Checking LD_LIBRARY_PATH variable
```

```
[ Not found ]
```

```
Performing file properties checks
```

```
Checking for prerequisites
```

```
[ OK ]
```

```
/bin/bash
```

```
[ OK ]
```

```
[...]
```

2. 配置 rkhunter

```
BINDIR="/cdrom/bin /cdrom/usr/bin /cdrom/sbin /cdrom/usr/sbin"
```

```
$ sudo nano /etc/default/rkhunter
```

```
REPORT_EMAIL="hiweed@hiweed.com"
```

```
$ less /etc/cron.daily/rkhunter
```

24.6.3 unhide 的使用

```
$ sudo apt-get install unhide
```

2. 检测隐藏进程

```
$ sudo unhide proc
```

```
Unhide 02-11-2007
```

```
yjesus@security-projects.com
```

```
[*]Searching for Hidden processes through /proc scanning
```



```
$ sudo unhide sys
```

```
Unhide 02-11-2007
```

```
yjesus@security-projects.com
```

```
[*]Searching for Hidden processes through getpriority() scanning
```

```
[*]Searching for Hidden processes through getpgid() scanning
```

```
[*]Searching for Hidden processes through getsid() scanning
```

```
[*]Searching for Hidden processes through sched_getaffinity() scanning
```

```
[*]Searching for Hidden processes through sched_getparam() scanning
```

```
[*]Searching for Hidden processes through sched_getscheduler() scanning
```

```
[*]Searching for Hidden processes through sched_rr_get_interval() scanning
```

```
[*]Searching for Hidden processes through sysinfo() scanning
```

```
HIDDEN Processes Found:2
```

```
$ sudo unhide brute
```

```
Unhide 02-11-2007
```

```
yjesus@security-projects.com
```

```
[*]Starting scanning using brute force against PIDS
```

```
Found HIDDEN PID: 31194
```

```
Found HIDDEN PID: 31200
```

```
Found HIDDEN PID: 31201
```

```
Found HIDDEN PID: 31202
```

```
Found HIDDEN PID: 31204
```

```
Found HIDDEN PID: 31211
```

```
Found HIDDEN PID: 31212
```

```
Found HIDDEN PID: 31215
```

```
Found HIDDEN PID: 31216
```

3. 检测隐藏的网络端口

```
$ sudo unhide-tcp
```

```
Unhide-TCP 28-12-2005
```

```
yjesus@security-projects.com
```

```
Starting TCP checking
```

```
Starting UDP checking
```

24.7 数据完整性检测

24.7.1 安装 Tripwire

```
$ sudo apt-get install tripwire
```

24.7.2 配置 Tripwire

1. 配置 twcfg.txt

```
ROOT            =/usr/sbin
POLFILE         =/etc/tripwire/tw.pol
DBFILE          =/var/lib/tripwire/${HOSTNAME}.twd
REPORTFILE      =/var/lib/tripwire/report/${HOSTNAME}-${DATE}.twr
SITEKEYFILE     =/etc/tripwire/site.key
LOCALKEYFILE    =/etc/tripwire/${HOSTNAME}-local.key
EDITOR          =/usr/bin/vi
LATEPROMPTING   =false
LOOSEDIRECTORYCHECKING =false
MAILNOVIOLATIONS =true
EMAILREPORTLEVEL =3
REPORTLEVEL     =3
SYSLOGREPORTING =true
MAILMETHOD      =SMTP
SMTPHOST        =localhost
SMTPPORT        =25
```

```
$ cd /etc/tripwire
$ sudo twadmin -m F -S site.key twcfg.txt

Please enter your site passphrase:
Wrote configuration file: /etc/tripwire/tw.cfg
```

```
$ sudo chmod 600 /etc/tripwire/twcfg.txt
```

```
$ sudo rm -f /etc/tripwire/twcfg.txt
```

```
$ sudo twadmin --print-cfgfile | sudo tee /etc/tripwire/twcfg.txt
```

2. 配置 twpol.txt

```
$ sudo nano /etc/tripwire/twpol.txt
```

```
#
```



```
# Critical devices
#
(
    rulename = "Devices & Kernel information",
    severity = $(SIG_HI),
    emailto = hiweed@hiweed.com
)
{
    /dev                -> $(Device) ;
    /proc/devices       -> $(Device) ;
    /proc/net           -> $(Device) ;
    /proc/sys           -> $(Device) ;
    /proc/cpuinfo       -> $(Device) ;
    /proc/modules       -> $(Device) ;
    /proc/mounts        -> $(Device) ;
    /proc/dma           -> $(Device) ;
    /proc/filesystems   -> $(Device) ;
    /proc/pci           -> $(Device) ;
    /proc/interrupts    -> $(Device) ;
    /proc/driver/rtc    -> $(Device) ;
    /proc/ioports       -> $(Device) ;
    /proc/scsi          -> $(Device) ;
    /proc/kcore         -> $(Device) ;
    /proc/self          -> $(Device) ;
    /proc/kmsg          -> $(Device) ;
    /proc/stat          -> $(Device) ;
    /proc/ksyms         -> $(Device) ;
    /proc/loadavg       -> $(Device) ;
    /proc/uptime        -> $(Device) ;
    /proc/locks         -> $(Device) ;
    /proc/version       -> $(Device) ;
    /proc/mdstat        -> $(Device) ;
    /proc/meminfo       -> $(Device) ;
    /proc/cmdline       -> $(Device) ;
    /proc/misc          -> $(Device) ;
}
```

```
$ cd /etc/tripwire
$ sudo twadmin -m P -S site.key twpol.txt
```

```
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
```

```
$ sudo chmod 600 /etc/tripwire/twpol.txt
```

```
$ sudo rm -f /etc/tripwire/twpol.txt
```

```
$ sudo twadmin --print-polfile | sudo tee /etc/tripwire/twpol.txt
```

```
$ sudo tripwire --test --email hiweed@hiweed.com
```

```
Sending a test message to: hiweed@hiweed.com
```

24.7.3 初始化 Tripwire 数据库

```
$ sudo tripwire -m i
```

```
Please enter your local passphrase:          <-- 输入 local 密钥的密码
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
### Warning: File system error.
### Filename: /etc/rc.boot
### No such file or directory
### Continuing...
### Warning: File system error.
### Filename: /root/mail
### No such file or directory
[...]
```

24.7.4 执行完整性检测

```
$ sudo tripwire --check --email-report
```

24.7.5 检测报告分析

```
$ sudo twprint -m r --twrfile /var/lib/tripwire/report/<your-file>.twr |less
```

检测报告的第一部分是摘要，列出了报告生成的时间等信息：

```
Note: Report is not encrypted.
Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:      root
Report created on:        Sat Apr  4 17:46:06 2009
Database last updated on:  Never

=====
Report Summary:
=====

Host name:                ubuntu
Host IP address:          127.0.1.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/ubuntu.twd
Command line used:        tripwire --check --quiet --email-report
```

```
=====
Rule Summary:
=====
```



Section: Unix File System				
Rule Name	Severity Level	Added	Removed	Modified
Invariant Directories	66	0	0	0
Tripwire Data Files	100	0	0	0
Other binaries	66	0	0	0
Tripwire Binaries	100	0	0	0
Other libraries	66	0	0	0
* Root file-system executables	100	0	0	1
System boot changes	100	0	0	0
Root file-system libraries (/lib)	100	0	0	0
Critical system boot files	100	0	0	0
* Other configuration files (/etc)	66	1	0	1
Boot Scripts	100	0	0	0
Security Control	66	0	0	0
Root config files	100	0	0	0
* Devices & Kernel information	100	212	84	0
Total objects scanned: 19623				
Total violations found: 299				

Object Detail:		
Section: Unix File System		
Rule Name: Root file-system executables (/sbin)		
Severity Level: 100		
Modified Objects: 1		
Modified object name: /sbin		
Property:	Expected	Observed
* Modify Time	Sat Apr 4 12:53:19 2009	Sat Apr 4 17:37:57 2009

24.7.6 查看 Tripwire 数据库内容

```
$ sudo twprint -m d --print-dbfile | less
Tripwire(R) 2.3.0 Database

Database generated by:      root
Database generated on:     Sat Apr 4 18:10:13 2009
Database last updated on:   Never
```

Database Summary:

```

Host name:          ubuntu
Host IP address:    127.0.1.1
Host ID:           None
Policy file used:   /etc/tripwire/tw.pol
Configuration file used: /etc/tripwire/tw.cfg
Database file used: /var/lib/tripwire/ubuntu.twd
Command line used:  tripwire -m i

```

Object Summary:

Section: Unix File System

Mode	UID	Size	Modify Time
-----	-----	-----	-----
/			
drwxr-xr-x	root (0)	XXX	XXXXXXXXXXXXXXXXXXXX
/bin			
drwxr-xr-x	root (0)	4096	Sat Apr 4 12:53:19 2009
/bin/bash			
-rwxr-xr-x	root (0)	702160	Mon May 12 14:33:24 2008
/bin/bunzip2			
-rwxr-xr-x	root (0)	26300	Fri Mar 21 06:32:33 2008
/bin/bzcat			
-rwxr-xr-x	root (0)	26300	Fri Mar 21 06:32:33 2008
/bin/bzcmp			
lrwxrwxrwx	root (0)	6	Wed Feb 25 22:20:19 2009

```
$ sudo twprint -m d --print-dbfile /bin/ls
```

Object name: /bin/ls

Property:	Value:
-----	-----
Object Type	Regular File
Device Number	2049
Inode Number	32848
Mode	-rwxr-xr-x
Num Links	1
UID	root (0)
GID	root (0)
Size	92376
Modify Time	Fri Apr 4 02:42:37 2008
Blocks	192
CRC32	CtpvTt
MD5	DliGdyfdJnMREIN3DJRFCZ



24.8 被入侵后的系统恢复

24.8.3 找到黑客入侵的方法

1. 找出非正常文件

```
$ sudo find / -xdev -ctime -10
```

```
$ sudo find / -name ".." -type d -print -xdev
$ sudo find / -name ".*" -print -xdev
$ sudo find / -name ".*,*" -print -xdev
```

```
$ sudo grep "x:0" /etc/passwd
$ sudo grep "x:0" /etc/passwd
```

```
$ sudo find / -user root -perm -4000 -print
```

4. 看看谁在服务器上

```
$ w
```

```
19:16:18 up 16:25, 2 users, load average: 1.00, 1.00, 1.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
hiweed    tty1     -                02:50    16:19  0.34s  0.29s -bash
hiweed    pts/0    hixp             10:11    0.00s  1.28s  0.38s sshd: hiweed [priv]
```

```
$ sudo netstat -nlp |grep ":22 "
```

```
(No info could be read for "-p": geteuid()=1000 but you should be root.)
```

```
tcp6      0      0 :::22                :::*             LISTEN        -
tcp6      0  52 192.168.1.10:22      192.168.1.119:1370 ESTABLISHED -
```

```
$ last -a
```

```
hiweed    pts/0    Sat Apr 4 10:11    still logged in    hixp
hiweed    pts/0    Sat Apr 4 02:58 - 10:07 (07:08)    hixp
hiweed    tty1     Sat Apr 4 02:50    still logged in
hiweed    tty1     Sat Apr 4 02:50 - 02:50 (00:00)
reboot    system boot Sat Apr 4 02:49 - 19:21 (16:32)    2.6.24-23-server
hiweed    tty1     Wed Feb 25 23:04 - down (00:00)
hiweed    tty1     Wed Feb 25 23:04 - 23:04 (00:00)
reboot    system boot Wed Feb 25 23:03 - 23:04 (00:00)    2.6.24-23-server
```

```
wtmp begins Wed Feb 25 23:03:33 2009
```



```
$ cat /var/log/auth.log | grep Accept
```

```
Apr  4 02:58:50 ubuntu sshd[4356]: Accepted password for hiweed from
192.168.1.119 port 3322 ssh2
Apr  4 10:11:38 ubuntu sshd[31004]: Accepted password for hiweed from
192.168.1.119 port 1370 ssh2
```

5. 查看当前网络状态

```
$ sudo netstat -nalp | less
```

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 192.168.1.10:22         192.168.1.119:1370     ESTABLISHED -
udp        0      0 0.0.0.0:68              0.0.0.0:*               -
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State      I-Node   PID/Program name  Path
unix    2      [ ]         DGRAM      upstart    5743     -                @/com/ ubuntu/
unix   11      [ ]         DGRAM      10577     -                /dev/ log
unix    2      [ ACC ]     STREAM    LISTENING  24411    -                /var/
run/m  ysqld/mysqld.sock
```

```
$ sudo netstat -plant | awk '$4 ~ /:80>/ {print}' | awk '{print $5}' | cut -f1
-d: | sort | uniq -c | sort -n
```

```
1 0.0.0.0
1 202.160.179.31
1 208.36.144.9
1 61.135.190.245
1 72.30.79.49
2 124.115.0.111
2 124.115.0.171
2 67.195.37.155
3 124.115.0.109
3 124.115.0.162
5 61.135.249.219
13 59.60.23.147
20 125.78.103.53
```

```
$ sudo netstat -plant | awk '{print $6}' | sort | uniq -c | sort -n
```

```
1 ESTABLISHED
1 Foreign
1 established)
17 LISTEN
38 TIME_WAIT
```

```
$ sudo lsof -i -n | less
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
---------	-----	------	----	------	--------	------	------	------



```

proftpd 1942 nobody 1u IPv4 44593574 TCP *:ftp (LISTEN)
inetd 2215 root 4u IPv4 4874 TCP *:auth (LISTEN)
sendmail- 2267 root 4u IPv4 45167849 TCP 127.0.0.1:smtp (LISTEN)
fcserver 2330 nobody 4u IPv4 5144 TCP *:1935 (LISTEN)
fcadmin 2338 root 1u IPv4 5148 TCP *:1111 (LISTEN)
java 2366 root 33u IPv4 5505 TCP 220.50.11.147:51127 (LISTEN)
asterisk 2414 root 14u IPv4 5437 TCP *:cisco-sccp (LISTEN)
nc 2433 root 3u IPv4 5352 TCP *:20710 (LISTEN)
mysqld 9504 mysql 13u IPv4 45256749 TCP *:mysql (LISTEN)
apache2 18375 www-data 4u IPv4 21204540 TCP *:www (LISTEN)
apache2 18375 www-data 23u IPv4 45434974 TCP 220.50.11.147:www->
124.115.0.162:32814 (ESTABLISHED)
sshd 19171 hiweed 3u IPv4 45432810 TCP 220.50.11.147:ssh->
119.165.92.251:3570 (ESTABLISHED)
rsync 26762 root 4u IPv4 45320044 TCP *:rsync (LISTEN)
named 27742 bind 513u IPv4 21257436 UDP 220.50.11.147:domain
sshd 32070 root 3u IPv4 21266250 TCP *:ssh (LISTEN)

```

```
$ sudo lsof -nP
```

```

COMMAND  PID  USER  FD  TYPE  DEVICE SIZE  NODE NAME
proftpd  1942  nobody 1u  IPv4  44593574  TCP *:21 (LISTEN)
inetd    2215  root   4u  IPv4  4874      TCP *:113 (LISTEN)
sendmail- 2267  root   4u  IPv4  45167849  TCP 127.0.0.1:25 (LISTEN)
fcserver 2330  nobody 4u  IPv4  5144      TCP *:1935 (LISTEN)
fcadmin  2338  root   1u  IPv4  5148      TCP *:1111 (LISTEN)
java     2366  root   33u IPv4  5505      TCP 220.50.11.147:51127 (LISTEN)
asterisk 2414  root   14u IPv4  5437      TCP *:2000 (LISTEN)
nc       2433  root   3u  IPv4  5352      TCP *:20710 (LISTEN)
mysqld   9504  mysql  13u IPv4  45256749  TCP *:3306 (LISTEN)
apache2  13769  root   4u  IPv4  21204540  TCP *:80 (LISTEN)
sshd     19171  hiweed 3u  IPv4  45432810  TCP 220.50.11.147:22->
119.165.92. 251:3570 (ESTABLISHED)
apache2  19500  www-data 4u  IPv4  21204540  TCP *:80 (LISTEN)
rsync    26762  root   4u  IPv4  45320044  TCP *:873 (LISTEN)
named    27742  bind   22u IPv4  21257438  TCP 127.0.0.1:953 (LISTEN)
sshd     32070  root   3u  IPv4  21266250  TCP *:22 (LISTEN)

```

6. 查看进程

```
$ ps -elf | less
```

```

4 S root      2414      1 0 80    0 - 7586 -      Jan17 ?          03:08:32 /usr/sbin/
as terisk -f
0 S root      2417      1 0 80    0 - 412 -      Jan17 tty2       00:00:00 /sbin/
getty 38400 tty2
0 S root      9465      1 0 80    0 - 619 -      06:29 ?          00:00:00 /bin/sh
/usr/ bin/mysqld_safe
4 S mysql    9504 9465 1 80    0 - 33428 -      06:29 ?          00:06:45 /usr/
sbin/my sld --basedir=/usr --datadir=/home/mysqldb --user=mysql --pid-
file=/var/ run/mysql ld/mysqld.pid --skip-external-locking --port=3306 --
socket=/var/run/ mysqld/mysqld. sock
0 S root      9505 9465 0 80    0 - 408 -      06:29 ?          00:00:00 logger -p
da emon.err -t mysqld_safe -i -t mysqld
5 S snort    13766 1 0 80    0 - 47709 -      06:30 ?          00:06:08 /usr/
sbin/sn ort -m 027 -D -d -l /var/log/snort -u snort -g snort -c
/etc/snort/snort.conf -S HO ME_NET=[210.51.1.136/32] -i eth0
5 S root      13769 1 0 80    0 - 10162 -      Feb20 ?          00:05:59 /usr/sbin/
ap ache2 -k start

```

```
5 S www-data 16622 13769 1 80 0 - 12283 - 15:49 ? 00:01:27 /usr/
sbin/a pache2 -k start
```

24.8.6 修复问题

```
$ sudo apt-get update && sudo apt-get dist-upgrade
```