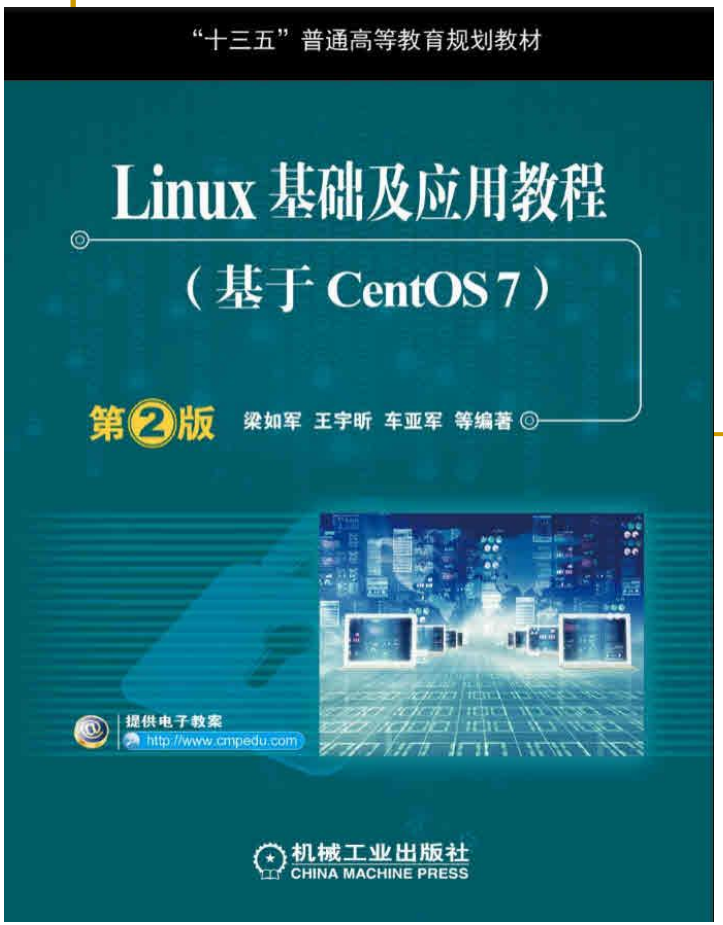
CentOS
The Community ENTerprise Operating System



第6章
基础架构服务

主讲人： 梁如军

2015-05-05

本章内容要点

- 守护进程
- 使用systemctl管理服务
- 安排周期性任务
- 日志系统与系统日志
- SSH与OpenSSH

本章学习目标

- 理解并管理守护进程
- 掌握周期性任务的设置方法
- 掌握日志系统的配置
- 查看系统日志
- 理解SSH协议体系结构
- 理解SSH基于主机和用户的安全验证
- 掌握OpenSSH服务的配置
- 掌握OpenSSH的用户密钥管理

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

管理守护进程

梁如军（linuxbooks@126.com）

守护进程（Daemon）

CentOS
The Community ENTerprise Operating System

- 始终在后台运行并响应合法请求的程序称为守护（Daemon）进程。
  - 守护进程不是由用户启动运行的，也不与终端关联。
  - 一个实际运行中的系统一般会有多个守护进程在运行，且各个系统中运行的守护进程都不尽相同。
  - 除非程序异常中止或者人为终止，否则它们将一直运行下去直至系统关闭。
- UNIX/Linux的守护进程在Windows系统中被称作“服务”。

梁如军（linuxbooks@126.com）

系统初始化进程

- **SysVinit**
  - 为 UNIX System V 系统创建的
  - RHEL/CentOS 5及之前的版本一直使用
- **Upstart**
  - 由Ubuntu创建的
  - RHEL/CentOS 6 使用Upstart
- **Systemd**
  - 先进的初始化系统
  - RHEL/CentOS 7使用Systemd

梁如军（linuxbooks@126.com）

使用systemctl管理服务（1）

- 显示当前已运行的所有服务
  - systemctl --type service 或
  - systemctl -t service
- 显示已加载的所有服务
  - systemctl --type service --all 或
  - systemctl -at service
- 显示已加载的但处于failed状态的服务
  - systemctl --type service --failed 或
  - systemctl -t service --failed

梁如军（linuxbooks@126.com）

使用systemctl管理服务（2）

- 启动名为ServiceName的服务
  - **systemctl start** <ServiceName>
- 停止名为ServiceName的服务
  - **systemctl stop** <ServiceName>
- 重启名为ServiceName的服务
  - **systemctl restart** <ServiceName>
- 重新加载名为ServiceName服务的配置文件
  - **systemctl reload** <ServiceName>
- 查看名为ServiceName服务的状态信息
  - **systemctl status** <ServiceName>

梁如军（linuxbooks@126.com）

使用systemctl管理服务（3）

- 在启动系统时启用名为ServiceName的服务
  - **systemctl enable** <ServiceName>
- 在启动系统时停用名为ServiceName的服务
  - **systemctl disable** <ServiceName>
- 查看名为ServiceName的服务是否在启动系统时启用
  - **systemctl is-enabled** <ServiceName>
- 查看所有服务是否在启动系统时启用
  - **systemctl list-unit-files -t service**

安排自动化任务

梁如军（linuxbooks@126.com）

自动安排进程任务

- 为什么要安排调度进程任务
- 调度任务的守护进程
  - atd
  - crond
- 安排调度任务的几个命令
  - at　　安排作业在某一时刻执行一次
  - batch　安排作业在系统负载不重时执行一次
  - cron　　安排周期性运行的作业

梁如军（linuxbooks@126.com）

CentOS 7中的cron

- 软件包名： cronie
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/crond.service
- 守护进程： /usr/sbin/crond
- 配置文件
  - /etc/sysconfig/crond
  - /etc/cron.d/0hourly
  - /etc/cron.deny
  - /etc/pam.d/crond

梁如军（linuxbooks@126.com）

与cronie相关的软件包

- 周期性计划任务由cronie软件提供
- 与cronie相关的软件包
  - cronie：提供crond守护进程及其配置目录/etc/cron.d以及用户使用的crontab命令
  - cronie-anacron：提供由crond调用的anacron程序及其配置文件/etc/anacrontab
  - crontabs：提供run-parts脚本以及依赖于此脚本的系统计划任务配置文件/etc/crontab和配置目录/etc/cron.{daily,weekly,monthly}

梁如军（linuxbooks@126.com）

cron与anacron

- cron假定服务器是24*7全天候运行的
  - 当系统时间变化或有一段关机时间就会遗漏这段时间应该执行的cron任务
- anacron （anachronistic cron）
  - 是针对非全天候运行而设计的
  - 是cron的一个连续时间版本
  - 不会因为时间不连续而遗漏计划任务的执行
  - 在CentOS 7中，anacron是cronie的一部分，且由crond调用的，仅用于运行系统常规计划任务

crond守护进程

- crond守护进程负责监控周期性任务的执行
- crond守护进程的执行参数配置文件
  - /etc/sysconfig/crond
- 控制普通用户的使用
  - 若/etc/cron.allow存在，仅列在其中的用户允许使用
  - 若/etc/cron.allow 不存在，检查/etc/cron.deny，没有列于其中的所有用户允许使用
  - 若两个文件均不存在，仅允许root用户使用
  - 空的/etc/cron.deny文件，表示允许所有用户使用（默认值）

梁如军（linuxbooks@126.com）

cron的工作过程

- crond启动以后，每分钟唤醒一次，检测如下文件的变化并将其加载到内存
  - **/etc/crontab**：是crontab格式（man 5 crontab）的文件
  - **/etc/cron.d/***：是crontab格式（man 5 crontab）的文件
  - **/var/spool/cron/***：是crontab格式（man 5 crontab）的文件
  - **/etc/anacrontab**：是anacrontab格式（man 5 anacrontab）的文件
- 一旦发现配置文件中安排的cron任务的时间和日期与系统的当前时间和日期符合时，就执行相应的cron任务
- 当cron任务执行结束后，任何输出都将作为邮件发送给安排cron任务的所有者，或者是配置文件中指定的MAILTO环境变量中指定的用户

梁如军（linuxbooks@126.com）

cron的工作过程（续）

- /var/spool/cron目录下的crontab文件
  - 是由用户使用**crontab**命令编辑创建的
  - 当每个用户使用crontab命令安排了cron任务之后，在/var/spool/cron目录下就会存在一个与用户同名的crontab文件
  - 例如一个用户名为osmond的用户，它所对应的crontab文件就是 /var/spool/cron/osmond

梁如军（linuxbooks@126.com）

crontab文件的格式

- 注释行以 # 开头
- 详情参见 man 5 crontab
- 每一行由5个时间字段及命令组成

  minute hour day-of-month month-of-year day-of-week commands
- 五个时间字段
  - minute： 一小时中的哪一分钟 **[0～59]**
  - hour： 一天中的哪个小时 **[0～23]**
  - day-of-month： 一月中的哪一天 **[1～31]**
  - month-of-year： 一年中的哪一月 **[1～12]**
  - day-of-week： 一周中的哪一天 **[0～6]**

梁如军（linuxbooks@126.com）

crontab文件的书写注意事项

- 这些项都**不能为空**，必须填入
- 如果用户不需要指定其中的几项时间，那么可以**使用统配符*表示任何时间**
- 每个时间字段都可以指定多个值，它们之间**用”,”间隔**，如：1,3,5
- 每个时间字段都可以指定范围，它们之间**用“-”间隔**，如：12-20
- 每个时间字段都可以使用***/n表示每隔n**，如：*/2
- 命令应该给出**绝对路径**，或**设置PATH环境变量**
- 用户必须**具有运行**所对应的命令或程序的**权限**

梁如军（linuxbooks@126.com）

run-parts与/etc/cron.{hourly,daily,weekly,monthly}目录

- crontabs软件包
  - run-parts：执行指定目录中的所有可执行文件

    # run-parts <directory>
  - /etc/cron.{hourly,daily,weekly,monthly}目录
    - **/etc/cron.hourly/**：存放每小时要执行的任务脚本文件
    - **/etc/cron.daily/**：存放每天要执行的任务脚本文件
    - **/etc/cron.weekly/**：存放每周要执行的任务脚本文件
    - **/etc/cron.monthly/**：存放每月要执行的任务脚本文件
- 例如 /etc/cron.d/0hourly

  01 * * * * root run-parts /etc/cron.hourly

梁如军（linuxbooks@126.com）

anacron 任务的执行

- anacron由crond调用间接执行
  - /etc/cron.hourly/0anacron

/usr/sbin/anacron -s

- anacron执行时会读取其配置文件
  - /etc/anacrontab

梁如军（linuxbooks@126.com）

anacron 的执行过程

- 对于每项任务，anacron 先判定该任务是否已在配置文件的周期字段中指定的期间内被执行了。如果它在给定周期内还没有被执行，anacron 会等待延迟字段中指定的分钟数，然后再次尝试执行命令字段中指定的命令。
- 当任务完成后，anacron 会将此日期记录在 /var/spool/anacron 目录的时间戳（Timestamp）文件中， 默认的时间戳文件有三个：cron.daily，cron.monthly 和 cron.weekly。

```
# ls /var/spool/anacron/
```


梁如军（linuxbooks@126.com）

anacron的配置文件——/etc/anacrontab

- 详情参见 man 5 anacrontab
- 每一行由4个字段组成

  period　delay　job-identifier　command
  - 周期（period）：命令执行的时间间隔（天数）。
  - 延迟（delay）：重启多少分钟后执行任务。可以使用这个设置防止在系统启动时集中执行作业。
  - 作业标识符（job-identifier）：任务的描述，用在 anacron 的消息中，并作为作业时间戳文件的名称，只能包括非空白的字符（斜线除外）。
  - 命令（command）：实际执行的任务。

梁如军（linuxbooks@126.com）

CentOS默认的/etc/anacrontab文件

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
RANDOM_DELAY=45
START_HOURS_RANGE=3-22

若1天之内没有运行“日任务”，则 5 分钟之后运行它
1 5 cron.daily nice run-parts /etc/cron.daily
若7天之内没有运行“周任务”，则 25 分钟之后运行它
7 25 cron.weekly nice run-parts /etc/cron.weekly
若一个月之内没有运行“月任务”，则 45 分钟之后运行它
@monthly 45 cron.monthly nice run-parts /etc/cron.monthly

计划任务的安排方法

- 管理员可在/etc/crontab 文件和/etc/cron.d/*目录的文件中书写crontab格式的文件
- 管理员可在/etc/cron.{hourly,daily,weekly,monthly}目录下安排每小时、每天、每周、每月要执行的脚本
- 管理员可以修改/etc/anacrontab文件配置非每天、每周、每月要执行的计划任务
- 每个用户都可以使用crontab命令安排自己的cron任务

修改系统crontab文件 安排计划任务

- 编辑 /etc/crontab

  0 */2 * * * netstat -a | mail osmond@mydomain.com

  0 2 * * 0 root du -sh /home/* | sort -nr | head -10

梁如军（linuxbooks@126.com）

创建/etc/cron.d/目录下的文件安排计划任务

- 例如：创建/etc/cron.d/ddns-update文件

```
*/5 * * * * root /usr/bin/wget -O /dev/null
--http-user=YOURNAME
--http-passwd=YOURPASSWORD
http://www.3322.org/dyndns/ update?system=dyndns&
    hostname=YOURHOSTNAME.f3322.org"
```


梁如军（linuxbooks@126.com）

直接编写任务脚本
安排计划任务

- 直接编写目录下的脚本安排计划任务
  - /etc/cron.{hourly,daily,weekly,monthly}
- 例如
  - # vi /etc/cron.daily/cleanup-backups

    find /backup -mtime +60 ! -name lost+found -exec /bin/rm –rf {} \;
  - # chmod +x /etc/cron.daily/cleanup-backups

梁如军（linuxbooks@126.com）

使用crontab命令
安排用户自己的cron任务

- crontab命令功能
  - 用于生成cron进程所需要的用户crontab文件
- crontab命令格式

  crontab [-u user] file

  crontab [-u user] {-l|-r|-e}
  - -l　在标准输出上显示当前的crontab
  - -r　删除当前的crontab
  - -e　使用编辑器编辑当前的crontab文件
  - 当结束编辑离开时，将自动安装/var/spool/cron目录下
- 任何被允许的用户都可以使用crontab安排任务

cron的使用举例

- 使用命令**crontab -e**加载任务
  - 在编辑器中编辑

```
00 02 * * * rm -rf  ~/temp/*
```

  - 存盘退出
- 用户的cron任务加载以后
  - 可以使用**crontab -l**命令查看
  - 可以到/var/spool/cron目录确认

梁如军（linuxbooks@126.com）

系统日常的cron任务

- tmpwatch
  - 清除指定目录中的文件，不让/tmp目录处于满状态
- logrotate
  - 日志滚动，让日志文件不要变得太大
  - 配置文件：/etc/logrotate.conf
- logwatch
  - 一个日志文件分析程序，提供系统活动摘要，报告可疑信息
  - 配置文件：/etc/logwatch/conf/logwatch.conf

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

日志系统和系统日志

梁如军（linuxbooks@126.com）

系统日志简介

- 日志的用途
  - 系统审计、监测追踪和分析统计。
- 日志的功能
  - 用于记录系统、程序运行中发生的各种事件
  - 通过阅读日志，有助于诊断和解决系统故障
  - 帮助系统管理员寻找攻击者留下的痕迹
- 日志的分类
  - 文件日志：将日志记录到文件中
  - 数据库日志：将日志记录关系数据库中
  - 管道日志：将日志通过管道传递给应用程序处理

梁如军（linuxbooks@126.com）

日志系统简介

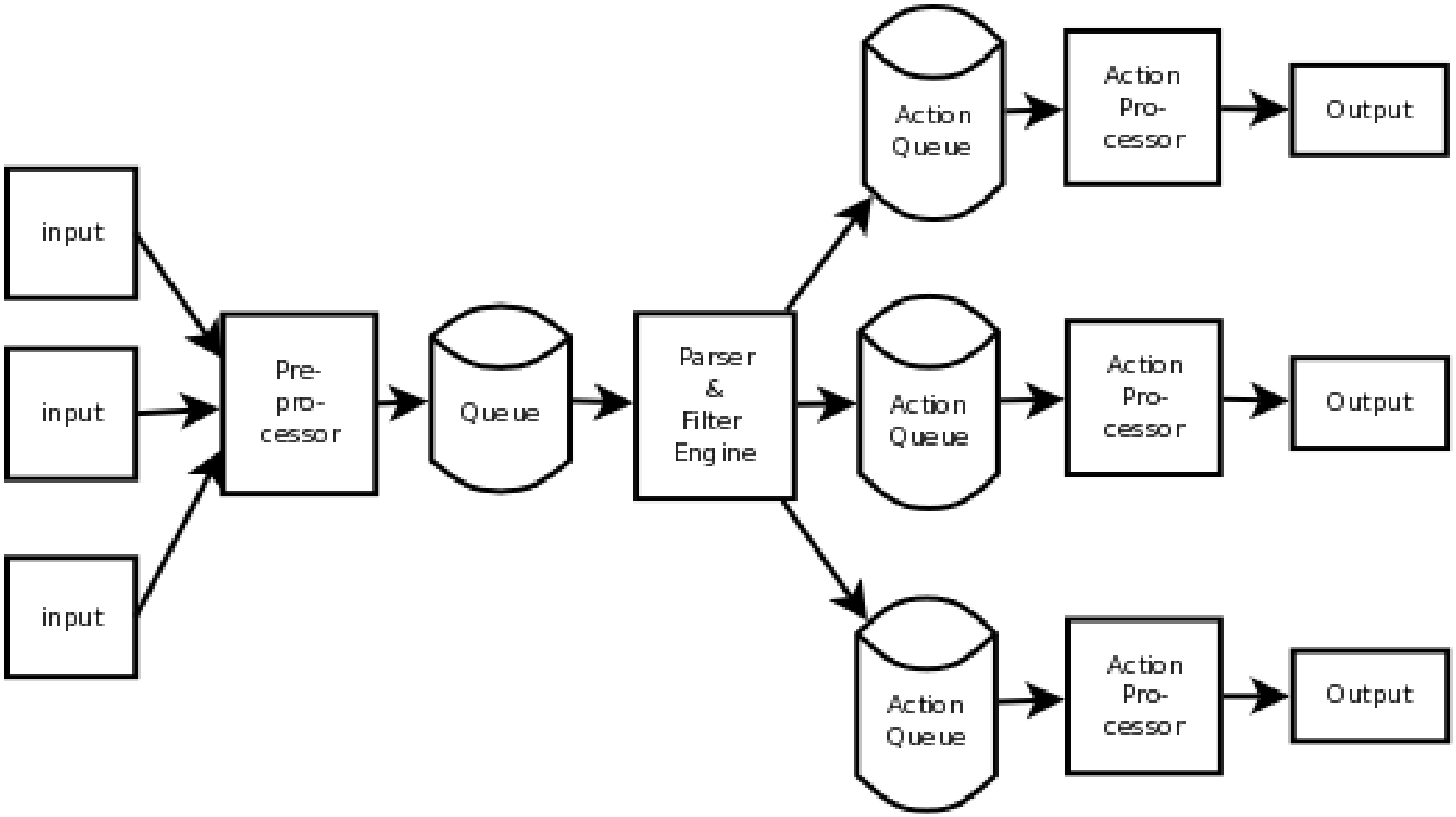
- 日志系统
  - 负责系统日志和内核消息捕捉的日志记录系统
- 日志系统的主要功能
  - 分类存放日志、方便日志管理
  - 可将日志消息记录到远程主机
- 日志系统的常用软件
  - syslog（ http://www.syslog.org/ ）
  - syslog-ng（http://www.balabit.com/network-security/syslog-ng）
  - rsyslog（http://www.rsyslog.com/）【默认安装】

rsyslog

- rsyslog采用模块化设计，是syslog的替代品
- 特点
  - 实现了基本的syslog协议
  - 直接兼容syslogd的syslog.conf 配置文件
  - 在同一台机器上支持多个rsyslogd进程
  - 丰富的过滤功能，可将消息过滤后再转发
  - 灵活的配置选项，配置文件中可写简单的逻辑判断
  - 增加了重要的功能，如使用TCP进行消息传输
  - 有现成的前端web展示程序

梁如军（linuxbooks@126.com）

rsyslog的体系结构



梁如军（linuxbooks@126.com）

rsyslog的消息流与模块

- 输入模块
  - imklg、imsock、imfile、imtcp
- 预处理模块
- 主队列
- 过滤模块
- 执行队列
- 输出模块
  - omudp、omtcp、omfile、omprog、ommysql、omruleset

梁如军（linuxbooks@126.com）

CentOS 7中的rsyslog

- 软件包名： rsyslog
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/rsyslog.service
- 守护进程：/sbin/rsyslogd
- 配置文件
  - /etc/rsyslog.conf 和 /etc/rsyslog.d/
  - /etc/sysconfig/rsyslog

梁如军（linuxbooks@126.com）

/etc/rsyslog.conf 的组成部分

- 全局指令（Global directives）
  - 设置全局参数，如：主消息队列尺寸、加载扩展模块等
- 模板（Templates）
  - 指定记录的消息格式，也用于动态文件名称生成
- 输出通道（Output channels）
  - 对用户期望的消息输出进行预定义
- 规则（Rules）【selector + action】
  - 指定消息规则
  - 在规则中可以引用之前定义的模板和输出通道
- 以 # 开始的行为注释，所有空行将被忽略

/etc/rsyslog.conf 的规则

- 规则由 selector 和 action 两部分组成
  - 每一行的格式为

    facility.priority　　action

    设备.级别　　动作
- 详情 **man 5 rsyslog.conf**

梁如军（linuxbooks@126.com）

默认的/etc/rsyslog.conf

 # egrep -v '^#|^$' /etc/rsyslog.conf

```
# provides support for local system logging (e.g. via logger command)
$ModLoad imuxsock
# provides kernel logging support (previously done by rklogd)
$ModLoad imklog
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
$IncludeConfig /etc/rsyslog.d/*.conf
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
authpriv.*                                              /var/log/secure
mail.*                                                  -/var/log/maillog
cron.*                                                  /var/log/cron
*.emerg                                                 *
uucp,news.crit                                          /var/log/spooler
local7.*                                                /var/log/boot.log
```


梁如军（linuxbooks@126.com）

设备类别（facility）

- **authpriv**：与安全、认证相关的信息
- **cron**：与cron和at有关的信息
- **daemon**：没有明确设备定义的守护进程的信息
- **ftp**：报告FTP守护进程的信息
- **kern**：报告与内核有关的信息
- **lpr**：报告与打印服务有关的信息
- **mail**：报告与邮件服务有关的信息
- **news**：报告与网络新闻服务有关的信息
- **syslog**：报告由syslog生成的信息
- **user**：报告一般的用户级别信息
- **uucp**：由UUCP子系统生成的信息
- **local0~local7**：保留给本地其他应用程序使用

梁如军（linuxbooks@126.com）

日志级别（priority）

- 0 **EMERG**（紧急）：会导致主机系统不可用的情况
- 1 **ALERT**（警告）：必须马上采取措施解决的问题
- 2 **CRIT**（严重）：比较严重的情况
- 3 **ERR**（错误）：运行出现错误
- 4 **WARNING**（提醒）：可能会影响系统功能的事件
- 5 **NOTICE**（注意）：不会影响系统但值得注意
- 6 **INFO**（信息）：一般信息
- 7 **DEBUG**（调试）：程序或系统调试信息等

facility.priority语法

- 可以在同一行中设置多个“设备.级别”组合，每组之间用分号隔开
- 可以同时指定多个设备或级别，用逗号间隔
- 可以使用通配符“*”表示所有的设备或级别
- 级别（priority）的指定
  - 直接指定：记录比指定级别高的所有级别的消息
  - 使用=前缀：只记录指定级别的日志消息
  - 使用!前缀：不记录比指定级别高的所有级别的消息
  - 使用!=前缀：不记录指定级别的日志消息
  - none：用于禁止任何日志消息

梁如军（linuxbooks@126.com）

facility.priority语法举例

- mail.info
- mail.=info
- kern.*
- kern.info;kern.!err
- mail.*;mail.!=info
- mail,news.=info
- *.=crit;kern.none
- *.=info;mail,news.none

梁如军（linuxbooks@126.com）

目标动作（ action ）

- 常规文件
  - 将日志记录于文件：以“/”开始的全路径文件名
  - **[-]/path/filename**
  - 文件名之前的减号（-）表示对文件的写入同时不立即同步到磁盘，这样可以加快日志写入速度
- 终端设备
  - 将日志发送到终端：**/dev/console，/dev/ttyX**
- 命名管道
  - 将日志记录到命名管道，用于日志调试非常方便
  - **| named_pipe**

目标动作（ action ）续

- 远程主机
  - **@hostname**
  - 将信息发送到可解析的远程主机hostname或IP
  - 该主机必须正在运行rsyslogd，并可以识别rsyslog的配置文件
  - rsyslog 使用 **udp:514** 端口传送日志信息
- 用户列表
  - **:omusrmsg:User1,User2,…Usern**
  - 发送信息到指定的登录用户，*表示所有用户

梁如军（linuxbooks@126.com）

action语法举例

kern.*	/var/adm/kernel
mail.*	-/var/log/maillog
mail.=info	/dev/tty12
*.alert	root,joey
*.=emerg	*
.	@finlandia

梁如军（linuxbooks@126.com）

将日志发往另一台主机

- 传统的UDP传输
  - 很有可能会丢失消息，但是传统的标准方法
  - ***.* @192.168.0.1**
- 基于TCP传输
  - 仅在某些特殊情况下丢失消息，被广泛使用
  - ***.* @@192.168.0.1**
- 基于 RELP 传输
  - 从不丢失消息，但要求 rsyslogd 的版本大于 3.15.0
  - ***.* :omrelp:192.168.0.1:2514**

梁如军（linuxbooks@126.com）

日志管理策略

- 及时作好备份和归档
- 延长日志保存期限
- 控制日志访问权限
  - 日志中可能会包含各类敏感信息，如账户、口令等
- 集中管理日志
  - 便于日志信息的统一收集、整理和分析
  - 杜绝日志信息的意外丢失、恶意篡改或删除

梁如军（linuxbooks@126.com）

日志滚动

- 为什么进行日志滚动
  - 对日志文件进行定期清理以免造成磁盘空间的不必要的浪费
  - 加快了管理员查看日志所用的时间
- 日志滚动程序
  - RHEL/CentOS 使用程序 logrotate 处理日志滚动
  - logrotate 能够自动完成日志的压缩、备份、删除工作
  - 系统默认把 logrotate 加入到系统每天执行的计划任务中

梁如军（linuxbooks@126.com）

logrotate 的配置文件

 # cat /etc/logrotate.conf

// 每周清理一次日志文件
weekly
// 保存过去四周的日志文件
rotate 4
// 清除旧日志文件的同时，创建新的空日志文件
create
// 包含/etc/logrotate.d目录下的所有配置文件
include /etc/logrotate.d

/etc/logrotate.d目录下的配置文件

 # ls /etc/logrotate.d

acpid cups mgetty rpm syslog named samba

……

- 每个文件的格式为：

```
# 注释
/full/path/to/logfile {
    配置语句1
    …
    配置语句n
}
```


梁如军（linuxbooks@126.com）

以cron方式运行logrotate

CentOS
The Community ENTerprise Operating System

 # cat /etc/cron.daily/logrotate

```
#!/bin/sh

/usr/sbin/logrotate /etc/logrotate.conf
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally
    with [$EXITVALUE]"
fi
exit 0
```


梁如军（linuxbooks@126.com）

主要日志文件简介

- 日志保存位置
  - 默认位于：/var/log 目录下
- 主要日志文件介绍
  - 内核及常规消息日志：/var/log/messages
  - 计划任务日志：/var/log/cron
  - 系统引导日志：/var/log/dmesg
  - 邮件系统日志：/var/log/maillog
  - 用户登录日志：/var/log/lastlog、/var/log/secure、/var/log/wtmp、/var/run/utmp
  - ……

梁如军（linuxbooks@126.com）

常规日志文件

/var/log/messages

- 是纯文本文件，可以使用 cat、tail [-f] 查看
- 文件中每条消息的格式
  - 时间标签　主机名　消息子系统　消息
- 例如

```
May 15 18:57:15 centos1 syslogd 1.4.1: restart.
May 15 19:20:13 centos1 sshd(pam_unix)[2968]: session opened for user root
by (uid=0)
May 15 20:02:07 centos1 sshd(pam_unix)[763]: session closed for user root
```

梁如军（linuxbooks@126.com）

用户登录日志

- 保存了用户登录、退出系统等相关信息
  - /var/log/lastlog：最近的用户登录事件
  - /var/log/wtmp：用户登录、注销及系统开、关机事件
  - /var/run/utmp：当前登录的每个用户的详细信息
  - /var/log/secure：与用户验证相关的安全性事件
- 分析工具
  - who、w、lastlog、last、ac

梁如军（linuxbooks@126.com）

应用程序日志

- 由相应的应用程序独立进行管理
  - Web服务：/var/log/httpd/
    - access_log、error_log
  - FTP服务：/var/log/vsftpd.log
  - ……
- 分析工具
  - 文本查看、grep过滤检索
  - awk、sed 等文本过滤、格式化编辑工具
  - Webalizer、Awstats等专用日志分析工具

梁如军（linuxbooks@126.com）

日志分析工具LogWatch

- LogWatch是一个对历史日志进行分析的工具
  - LogWatch由Perl语言编写
  - 主页为 http://www.logwatch.org/
  - 在RHEL/CentOS 中 LogWatch工具由RPM包logwatch提供，且是默认安装的
  - 默认情况下，logwatch以cron任务方式每日运行一次。

OPENSSH

梁如军（linuxbooks@126.com）

SSH

CentOS
The Community ENTerprise Operating System

- 为什么使用 SSH
- SSH 2 体系结构
- CentOS 上的 OpenSSH
- 生成密钥对
- 复制公钥到目标
- 服务器配置
- 客户端配置

漏洞

CentOS
The Community ENTerprise Operating System

- 未加密的网络连接容易嗅探，欺骗，劫持
- 需要认证保护
  - 抵御针对凭证的捕获攻击
  - 允许认证方法的替代方案
- 需要保护数据
  - 提供交互式登录安全
  - 为所传输的数据提供安全保障

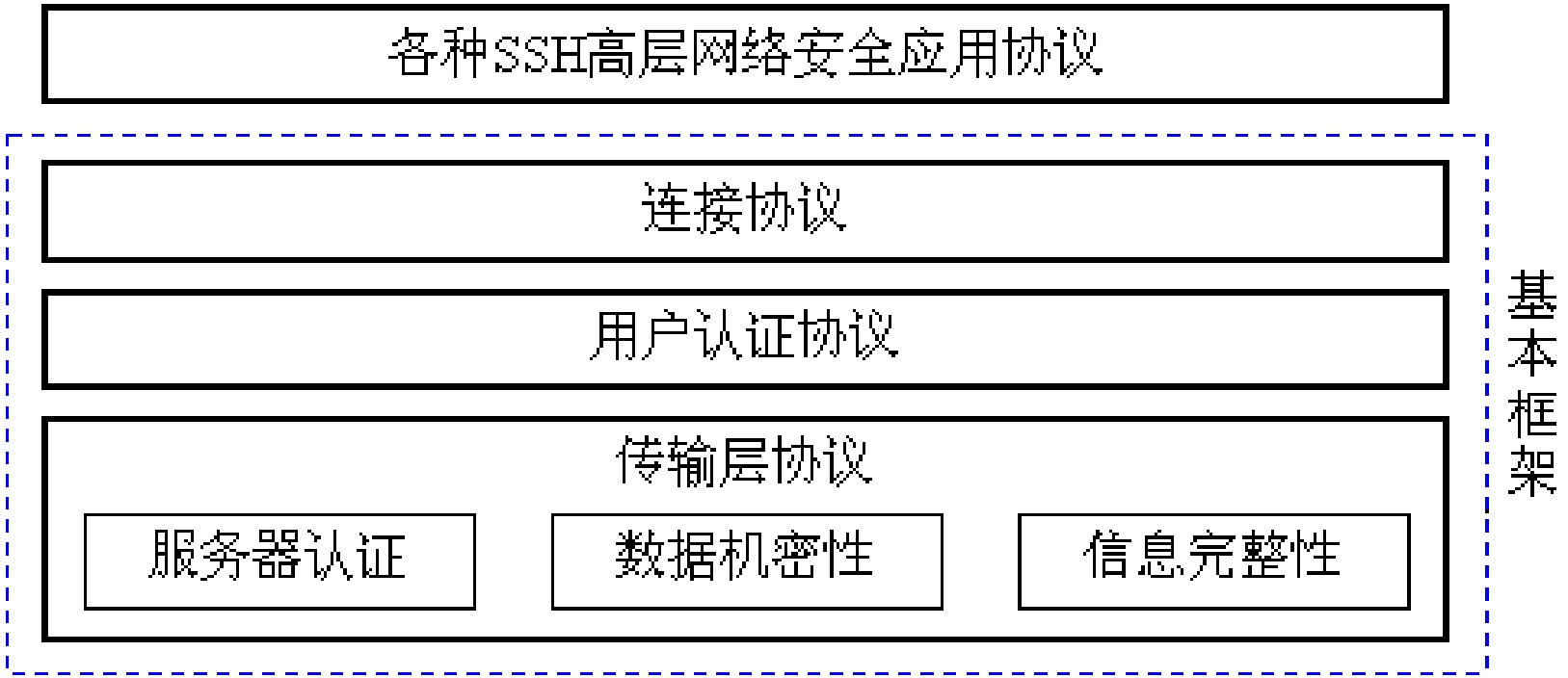
解决

- 使用SSH进行交互式登录会话
- 在可信任的主机之间配置用户的公钥认证
- 使用TCP端口转发保护未加密的数据通道

SSH简介

- SSH 的英文全称为 **Secure SHell**
- SSH 是IETF 的网络工作组所制定的协议
- SSH 是建立在应用层和传输层基础上的安全协议
- SSH（Secure SHell）协议是 C/S 模式协议
  - 分为 SSH 的客户端和服务器端
  - 一次成功的 SSH 会话需要两端通力合作来完成
- SSH 目的是要在非安全网络上提供安全的远程登录和其他安全服务
  - 所有使用 SSH 协议的通信，包括口令，都会被加密传输
  - 用于替代传统的 telnet、ftp、r族命令（rlogin、rsh、rcp）

SSH协议体系结构



梁如军（linuxbooks@126.com）

SSH 基于主机的安全验证

- 在 SSH 协议中每台主机都有一对或多对主机密钥
  - 首次启动 SSH 服务时会自动生成，一般无需变更
- SSH 通过严格的主机密钥检查
  - 用户可以核对来自服务器的公钥同之前所定义的密钥是否一致，防止了某个用户访问一个他没有相应公钥的主机
  - SSH 利用主机的公钥（而不是IP地址）实现主机身份认证，不容易受到IP地址欺骗的攻击

SSH 基于用户的安全验证1

- 基于口令的安全验证
  - 只要用户知道自己用户账号和口令，就可以登录到远程主机
  - 口令验证由 PAM 进行验证
  - SSH 对所有传输的数据进行加密传输（包括用户口令）
  - 不能避免受到“中间人”方式的攻击

梁如军（linuxbooks@126.com）

SSH 基于用户的安全验证2

- 基于密钥的安全验证
  - 每个用户都拥有自己的一对或多对密钥
    - 包括：公钥和私钥
    - 密钥协议： RSA 或 DSS/DSA
  - 每个用户自己的密钥对需用户自己生成
    - 可以使用不同的密钥协议创建多对密钥
    - 并将公密发布到需要访问的服务器上
  - 基于密钥的安全验证不需要在网络上传送用户口令
  - 可以避免“中间人”的攻击方式，因为“中间人”没有你的私钥

梁如军（linuxbooks@126.com）

用户密钥认证协议

CentOS
The Community ENTerprise Operating System

- RSA 和 DSS/DSA 认证承诺不必提供密码就能够同远程系统建立连接
- RSA/DSA 密钥认证协议的基本工作原理：
  - 密钥由一对组成：一把专用密钥（亦称私钥）和一把公用密钥（亦称公钥）。
  - 密钥对由客户端生成，私钥由用户自己保管，并将公钥散播到需要认证之处（登录服务器端）。
  - 公钥用于对消息进行加密，只有拥有私钥的人才能对该消息进行解密。
  - 公钥只能用于加密，而私钥只能用于对由匹配的公钥编码的消息进行解密。

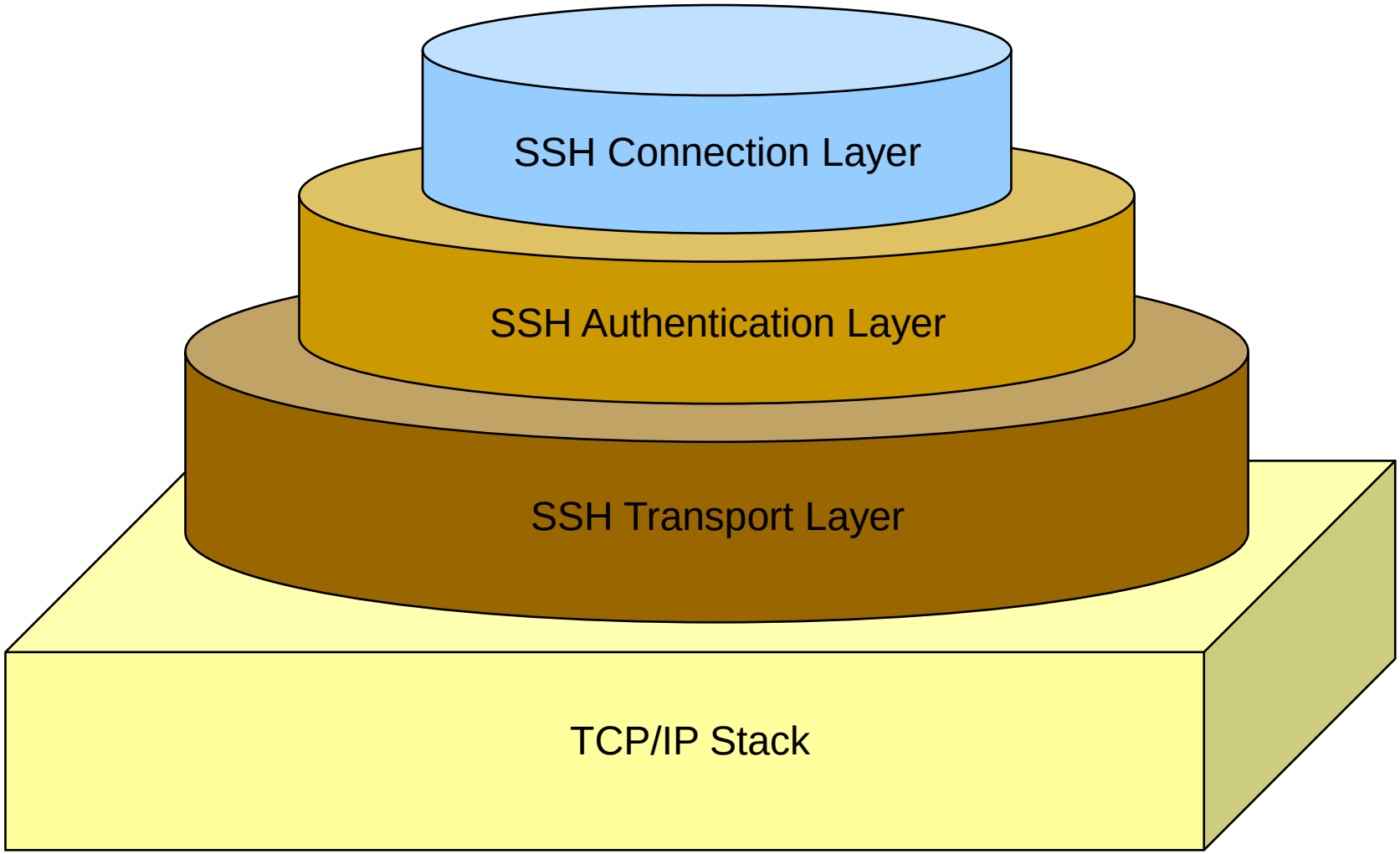
梁如军（linuxbooks@126.com）

用户密钥认证过程

- SSH 客户向 SSH 服务器提出用户密钥认证请求
- SSH 服务器在用户的家目录下寻找其公钥，然后把它和用户发送过来的公钥进行比较
  - 如果两个密钥一致
    - 服务器就用公钥生成加密“质询”（challenge）并把它发送给客户端软件
    - 客户端软件收到“质询”之后就使用用户自己的私钥解密再把它送还给服务器
    - 认证通过后，客户端向服务器发送会话请求开始双方的加密会话
  - 否则认证失败

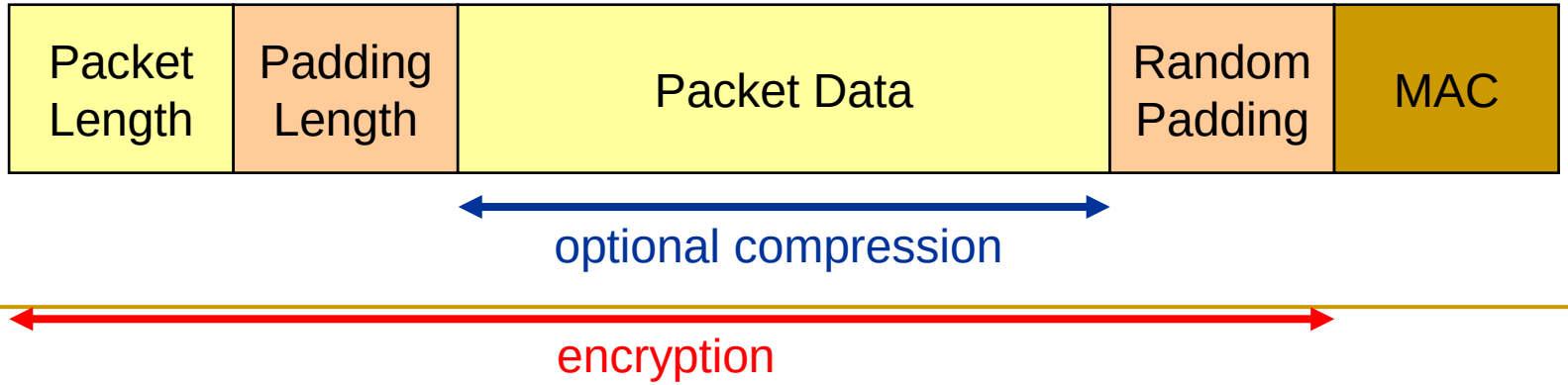
梁如军（linuxbooks@126.com）

SSH 2 体系结构


SSH Connection Layer
SSH Authentication Layer
SSH Transport Layer
TCP/IP Stack

SSH 2 – 传输层

- 传输层提供算法协商，密钥交换和服务器身份验证，并设置了加密的安全连接，提供完整性，保密性和可选的压缩。
- 密钥交换使用了一个1024位系数的 Diffie-Hellman 协议确保完美的保密。
- 服务器身份验证基于RSA或DSS签名，可以使用原始的公共密钥或X.509，PGP或SPKI证书。



初始服务器密钥发现

CentOS
The Community ENTerprise Operating System

- 首先，一个客户连接到 ssh 服务器并请求验证服务器的密钥

```
[djm@roku djm]$ ssh root@hachi.mindrot.org
The authenticity of host 'hachi.mindrot.org (203.36.198.102)'
    can't be established.
RSA key fingerprint is cd:41:70:30:48:07:16:81:e5:30:34:66:f1:56:ef:db.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
root@hachi.mindrot.org's password: xxxxxxxx
Last login: Tue Aug 27 10:56:25 2002
[root@hachi root]#
```

- 这样做是为了防止攻击者冒充服务器，这将让他们有机会捕捉到的密码或会话的内容。
- 一旦服务器的密钥已被验证，它被记录在 **~/.ssh/known_hosts** 里，因此它可以自动检查每个连接。

SSH 2 – 认证层

CentOS
The Community ENTerprise Operating System

- 认证层为用户身份验证提供了几种机制。这些措施包括传统的密码认证，以及公共密钥或基于主机的认证机制。
- 基于口令的认证: 用户名和口令的安全传输是基于加密的ssh传输层。在服务器上正常的基于口令的登录将会发生。
- 基于密钥的认证:用户使用其私钥签署一个由服务器发送的 challenge。用户的公钥文件 `id_rsa.pub`必须首先安装在服务器 `~/.ssh/authorized_keys` 文件中

SSH 2 – 连接层

CentOS
The Community ENTerprise Operating System

- 连接层 提供
  - 交互式登录会话:

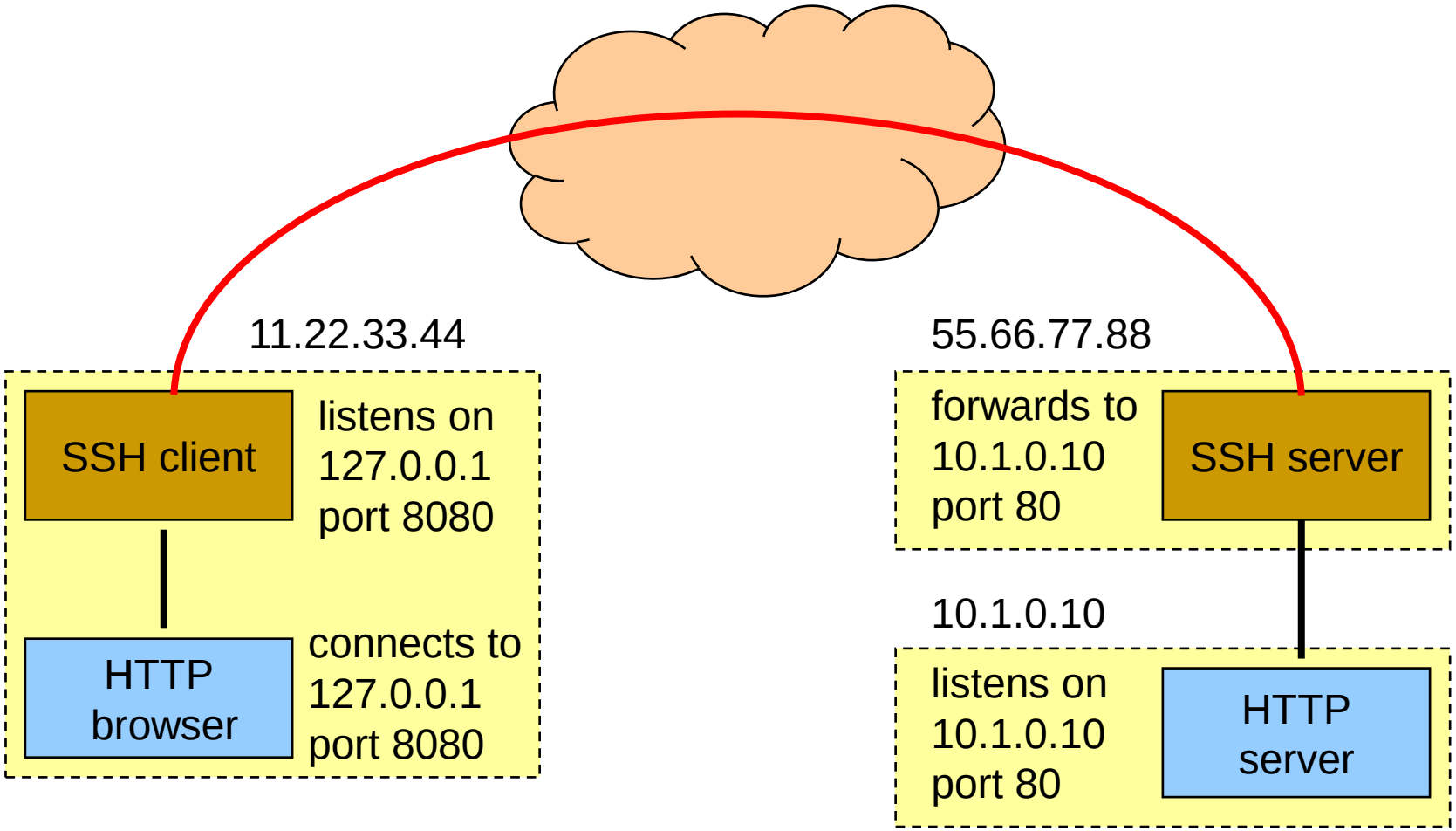
```
$ ssh -l antje srv.kool.net
```

  - 远程执行命令:

```
$ ssh antje@srv.kool.net "rm *"
```

  - 通过 scp 或 sftp 命令安全地实现远程文件或目录的复制
  - 转发 TCP / IP 连接
  - 和转发 X11 连接
- 所有这些通道被复用成一个单一的加密隧道。

SSH 2 – TCP/IP 端口转发




```
ssh -L8080:10.1.0.10:80  55.66.77.88
```

SSH 2 – 实现

CentOS
The Community ENTerprise Operating System

- OpenSSH for OpenBSD
  - http://www.openssh.org
- Portable OpenSSH for Linux, Unix, Mac OS X
  - http://www.openssh.org
- PuTTY for Windows
  - http://www.chiark.greenend.org.uk/~sgtatham/putty/
- WinSCP graphical Windows scp and sftp client
  - http://winscp.net/

客户端的主机认证

- /etc/ssh/ssh_known_hosts
  - 用户可能有 ~/.ssh/known_hosts
  - UserKnownHostsFile
- StrictHostKeyChecking
  - 默认为 ***ask***，让用户决定是否接受未知的服务器密钥
- ssh-keyscan
  - 收集指定主机的公钥

客户端的用户认证

CentOS
The Community ENTerprise Operating System

- 用户身份验证方法请求
  - PasswordAuthentication
  - PubkeyAuthentication
    - Key files and ~/.ssh/authorized_keys
  - HostbasedAuthentication
    - Insecure, do not use
- 实际使用的方法基于远程服务器的控制

OpenSSH的主机密钥管理——主机密钥生成

- OpenSSH的服务启动脚本 **/etc/init.d/sshd**
  - 包含了主机密钥的生成命令
  - sshd首次启动时默认会生成三对主机密钥（SSH-1 RSA、SSH-2 RSA、SSH-2 DSA）
- 何时需要重新生成主机密钥
  - 若系统是从一个旧系统克隆而来
    - 如：硬盘对拷、复制的虚拟机文件等
- 在 RHEL/CentOS 上重新生成主机密钥

```
# rm -f /etc/ssh/ssh_host_*
# service sshd restart
```


梁如军（linuxbooks@126.com）

OpenSSH的主机密钥管理——搜集可信任主机的公钥

- **ssh-keyscan** 命令格式

  ssh-keyscan -t <rsa|ecdsa|ed25519> <Hostname|IPaddress> [<Hostname|IPaddress> …]

  - -t <rsa|ecdsa|ed25519>：用于指定密钥算法。可以同时指定多种算法（用逗号间隔）
  - Hostname|IPaddress：指定被信任主机的主机名或 IP 地址

- 例如：
  - $ ssh-keyscan -t rsa,ecdsa soho soho.ls-al.loc 192.168.0.200 > ~/.ssh/known_hosts

梁如军（linuxbooks@126.com）

OpenSSH的用户密钥管理——密钥生成和分发

- 密钥生成
  - ssh-keygen [**-t rsa|dsa|ecdsa**]
    - 默认使用rsa密钥认证类型
- 密钥分发
  - 私钥（private key）被保留在自己的系统上
    - 通常使用私钥短语保护私钥（推荐）
    - 重新设置私钥保护短语

      $ ssh-keygen -f ~/.ssh/id_rsa -p
  - 公钥（public key）被分发（复制）到目标系统

    $ ssh-copy-id -i ~/.ssh/id_rsa.pub [user@]host

梁如军（linuxbooks@126.com）

保护私钥

CentOS
The Community ENTerprise Operating System

- 使用私钥保护短语保护私钥
  - 是私钥丢失后的一道防线
- **ssh-agent** 存储私钥保护短语
  - 会话开始时输入私钥的保护短语
  - 使用 **ssh-add** 管理私钥保护短语
- 使密钥盗窃更难，但不是不可能

OpenSSH的用户密钥管理——ssh-agent和ssh-add

- ssh-agent
  - 是一个用户认证代理（authentication agent）
  - 用于在登录会话中缓存解密后的私钥
  - ssh/scp/sftp命令内置支持了同 ssh-agent 通信的机制
  - 使得私钥保护口令只需要输入一次
- ssh-add
  - 用于向认证代理的高速缓存中添加私钥

梁如军（linuxbooks@126.com）

OpenSSH的用户密钥管理——ssh-agent的运行方法

- 在Shell中运行

  $ eval $(ssh-agent)

  - 或

  $ ssh-agent bash

- 在用户登录脚本中运行

  $ vi ~/.bash_profile

  - 添加如下行

  eval $(ssh-agent)

梁如军（linuxbooks@126.com）

CentOS 7中的sshd

- 软件包名： openssh-server
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/sshd.service
- 守护进程： /usr/sbin/sshd
- 配置文件
  - /etc/ssh/sshd_config
  - /etc/sysconfig/sshd

梁如军（linuxbooks@126.com）

OpenSSH 服务

- 安装 OpenSSH 服务

  # yum install openssh-server

  - RHEL/CentOS 默认安装了openssh-server
- 配置文件是 **/etc/ssh/sshd_config**
  - 一般情况下无需修改，默认的配置即可工作良好
  - 配置文件中的语句说明参见
    - http://lamp.linux.gov.cn/OpenSSH/sshd_config.html
- SSH 服务安全
  - 使用 DenyHosts / fail2ban 实现访问控制
  - 使用 pam_access 模块实现口令认证的访问控制

梁如军（linuxbooks@126.com）

OpenSSH服务配置

- 用户访问控制
  - AllowUsers/AllowGroups
  - DenyUsers/DenyGroups
  - PermitRootLogin
- 认证方式 (Password or/and Keys)
  - PasswordAuthentication
- StrictModes
  - If a user's SSH configuration files or home directory are world-writable, deny access

/etc/ssh/sshd_config ——OpenSSH 服务的安全配置1

- 仅使用SSH版本2协议
  - **Protocol 2**
- 限制服务监听IP地址和端口（请根据需要修改）
  - **ListenAddress 192.168.0.222:22**
  - **ListenAddress 202.55.66.77:2222**
- 限制用户访问（请根据需要修改）
  - **DenyUsers　user1 user2 foo**
  - **AllowUsers　root osmond vivek**

梁如军（linuxbooks@126.com）

/etc/ssh/sshd_config ——OpenSSH 服务的安全配置2

- 禁止root登录（需配置sudo）
  - **PermitRootLogin no**
- 仅允许root用户仅通过密钥认证登录
  - **PermitRootLogin without-password**
- 设置所有用户仅能通过密钥认证登录
  - **PasswordAuthentication no**
- 启用PAM用户认证
  - **UsePAM yes**

/etc/ssh/sshd_config ——OpenSSH 服务的安全配置3

- 禁止模拟已被视为废弃的rsh的认证方式
  - 即不读取文件 ~/.rhosts 和 ~/.shosts
  - **ChallengeResponseAuthentication no**
  - **HostbasedAuthentication no**
  - **IgnoreRhosts yes**
- 设置用户登录会话空闲5分钟自动注销
  - **ClientAliveInterval 300**
  - **ClientAliveCountMax 0**

梁如军（linuxbooks@126.com）

安全隧道（stunnel）

- 为不安全的服务提供安全的访问
- 使用SSL，没有内置的加密技术
  - 需要一个证书
- 防止截取数据
- 防止数据篡改

端口转发

- *ssh* 和 *sshd* 能转发 TCP 流量
- 语法
  - -L clientport:host:hostport
  - -R serverport:host:hostport
- 可以用来绕过访问控制
  - 要求客户端成功验证远程的 sshd
  - AllowTcpForwarding

本章思考题

- 什么是守护进程？守护进程的运行方式和分类？
- 什么是cron任务？如何加载cron任务？如何安排cron任务?
- 简述crontab文件中各个字段的含义及书写规范。
- 什么是rsyslog？rsyslog功能？
- 可以使用哪些命令查看非文本日志文件？
- LogWatch和SEC的功能？
- 为什么要进行日志滚动？CentOS如何实现日志滚动？
- 简述SSH的工作过程。

梁如军（linuxbooks@126.com）

本章实验

- 学会使用守护进程管理工具。
- 学会安排用户自己的cron任务。
- 学会安排系统的cron任务。
- 学会配置远程日志。
- 学会配置日志滚动。
- 学会查看系统日志文件。
- 配置ssh服务以加强安全性
- 配置ssh/scp/sftp的密钥登录
- 使用ssh-agent和ssh-add

梁如军（linuxbooks@126.com）

进一步学习

- 学习systemctl的使用方法。
- 学习使用at命令安排一次性任务。
- 学习使用rsyslog配置中央日志服务器。
- 学习在中央日志服务器上安装和使用LogAnalyzer。
- 学习时间同步服务器（ntpdate和ntpd）的配置和使用。
- 学习使用SSH 隧道（Tunnel）实现
  - 端口转发（Port forward）的方法
  - https://www.ibm.com/developerworks/cn/linux/l-cn-sshforward/。
- 学习使用keychain完美实现ssh/scp/sftp无口令登录的方法。
- 学习使用denyhost或 fail2ban保护ssh服务。

进一步学习（续）

- 在Windows系统上安装Git（http://msysgit.github.io/），学习使用git命令操作本地仓库。
- 使用Git同时安装的OpenSSH创建自己的密钥对，在GitHub（https://github.com）上注册自己的账号并上传自己的公钥，学习操作远程git仓库。有关 Github 的使用，请参考蒋鑫所著的开源书 GotGitHub（http://www.worldhello.net/gotgithub/）。
- 学习使用etckeeper（EPEL仓库有提供）以git方式管理系统的/etc目录。

DNS缓存服务（DNSMASQ）

梁如军（linuxbooks@126.com）

Dnsmasq

- Dnsmasq是一个轻量级的易于配置的DNS转发器（DNS forwarder）和DHCP服务器
- Dnsmasq为小型网络提供了DNS和可选择的DHCP功能。它服务那些只在本地适用的域名，这些域名是不会在全球的DNS服务器中出现的。
- Dnsmasq的设计目标是用于使用NAT通过ADSL连接因特网的家庭网络。对于那些需求低资源消耗且配置方便简单的小型网络（最多可支持1000台主机）也是一个很好的选择。

Dnsmasq的功能

- 使用/etc/hosts文件实现本地内网解析
- 从上游域名解析服务器中获取公网的IP地址-域名映射关系放入缓存
- 被配置用来向特定的上游DNS服务器发送特定的域名解析请求，从而可以简单的与私有的DNS服务器结合使用
- 集成的DHCP服务器支持静态和动态的DHCP租约服务，支持多网络和多IP地址范围，并且实现了 BOOTP/TFTP/PXE 协议用于支持无盘工作站或网络设备的远程启动

梁如军（linuxbooks@126.com）

CentOS 7中的dnsmasq

- 软件包名： dnsmasq
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/dnsmasq.service
- 守护进程： /usr/sbin/dnsmasq
- 配置文件
  - /etc/dnsmasq.conf
  - /etc/dnsmasq.d/

梁如军（linuxbooks@126.com）

Dnsmasq的配置文件

- 安装Dnsmasq
  - # yum install dnsmasq
- Dnsmasq的主配置文件是/etc/dnsmasq.conf
  - 在/etc/dnsmasq.conf中启用/etc/dnsmasq.d目录

    conf-dir=/etc/dnsmasq.d
- 检查Dnsmasq配置语法的正确性
  - # dnsmasq --test

梁如军（linuxbooks@126.com）

Dnsmasq的基本配置

- /etc/dnsmasq.d/common.conf

```
listen-address=127.0.0.1,192.168.0.254
user=dnsmasq
group=dnsmasq
log-queries
log-facility=/var/log/dnsmasq.log
pid-file=/var/run/dnsmasq.pid
```

配置Dnsmasq的日志滚动

```
# groupadd -r dnsmasq
# useradd -r -g dnsmasq dnsmasq
# cat >  /etc/logrotate.d/dnsmasq <<END
/var/log/dnsmasq.log {
        missingok
        notifempty
        delaycompress
        sharedscripts
        size 200M
        weekly
        create 0640 dnsmasq dnsmasq
        postrotate
                [ ! -f /var/run/dnsmasq.pid ] \
 || kill -USR2 \`cat /var/run/dnsmasq.pid\`
        endscript
}
END
```


梁如军（linuxbooks@126.com）

配置Dnsmasq的本地DNS缓存和转发器

- /etc/dnsmasq.d/dns.conf

```
domain-needed
bogus-priv
dns-forward-max=150
cache-size=1000
neg-ttl=3600
no-poll
no-resolv

local=/olabs.lan/
server=8.8.8.8
server=208.67.222.222

server=/baidu.com/114.114.114.114
address=/lan/192.168.0.254
```


梁如军（linuxbooks@126.com）

配置Dnsmasq的内网解析

- /etc/hosts

```
192.168.0.254        zbox
192.168.0.253        mele
192.168.0.252        soho
192.168.0.251        cubie
```


梁如军（linuxbooks@126.com）

配置Dnsmasq的DHCP和tftpd服务

- /etc/dnsmasq.d/dhcp-tftp.conf

```
dhcp-range=192.168.0.30,192.168.0.80,255.255.255.0,12h
dhcp-option=3,192.168.0.1
dhcp-option=6,192.168.0.254

dhcp-host=11:22:33:44:55:66,fred,192.168.0.60

dhcp-boot=pxelinux.0,192.168.0.254
enable-tftp
tftp-root=/var/lib/tftpboot
dhcp-authoritative
dhcp-ignore=tag:!known
dhcp-ignore=#known
no-dhcp-interface=eth1
```


梁如军（linuxbooks@126.com）

启动Dnsmasq

```
# service dnsmasq restart

# netstat -ltup|grep dnsmasq
tcp        0      0 *:domain      *:*         LISTEN      5278/dnsmasq
tcp        0      0 *:domain      *:*         LISTEN      5278/dnsmasq
udp        0      0 *:domain      *:*                     5278/dnsmasq
udp        0      0 *:bootps      *:*                     5278/dnsmasq
udp        0      0 *:tftp        *:*                     5278/dnsmasq
udp        0      0 *:domain      *:*                     5278/dnsmasq
```


梁如军（linuxbooks@126.com）

HTTP缓存服务（POLIPO）

梁如军（linuxbooks@126.com）

Polipo简介

- Polipo是一个为个人或小型网络设计的轻量级的Web缓存代理
- Polipo同时提供了一个Web接口
  - 用于查看文档、查看配置和本地磁盘缓存等
  - http://your_polipo_server:**8123**

梁如军（linuxbooks@126.com）

CentOS 7中的Polipo

- 软件包名： polipo（EPEL仓库）
- 服务类型：由Systemd启动的守护进程
- 配置单元： /usr/lib/systemd/system/polipo.service
- 守护进程： /usr/bin/polipo
- 配置文件
  - /etc/polipo/config
  - /etc/logrotate.d/polipo

梁如军（linuxbooks@126.com）

使用Polipo配置HTTP缓存代理

- /etc/polipo/config

```
daemonise = true
pidFile = /var/run/polipo/polipo.pid
proxyAddress = 0.0.0.0
allowedClients = 127.0.0.1,192.168.0.0/24
cacheIsShared = true
chunkHighMark = 50331648
objectHighMark = 16384
disableIndexing = false
dnsQueryIPv6 = no
```


梁如军（linuxbooks@126.com）

配置polipo的代理客户

- 设置 yum 使用代理

  # echo 'proxy=http://127.0.0.1:8123' >> /etc/yum.conf

- 设置 wget 使用代理

  # echo 'http-proxy = 127.0.0.1:8123' >> /etc/wgetrc

  - 或

  # echo 'http-proxy = 127.0.0.1:8123' >> ~/.wgetrc

梁如军（linuxbooks@126.com）

ATD守护进程和AT命令

梁如军（linuxbooks@126.com）

atd守护进程

- atd守护进程负责监控一次性任务的执行
- atd守护进程的执行参数 /etc/sysconfig/atd
- 控制普通用户的使用
  - 若/etc/at.allow存在，仅列在其中的用户允许使用
  - 若/etc/at.allow 不存在，检查/etc/at.deny，没有列于其中的所有用户允许使用
  - 若两个文件均不存在，仅允许root用户使用
  - 空的/etc/at.deny文件，表示允许所有用户使用（默认值）

梁如军（linuxbooks@126.com）

at命令的功能和格式

- 功能：安排一个或多个命令在指定的时间运行一次。
- at 命令格式及参数
  - at [-q 队列] [-f 文件名] 时间
  - at -d
  - at -l
  - at -c
- atq（at -q）
- atrm（at -d）

梁如军（linuxbooks@126.com）

at命令指定时间的方式

- 绝对计时方法
  - midnight noon teatime
  - hh:mm [today]
  - hh:mm tomorrow
  - hh:mm 星期
  - hh:mm mm/dd/yyyy
- 相对计时方法
  - now + n minutes
  - now + n hours
  - now + n days
  - now + n weeks

梁如军（linuxbooks@126.com）

at命令指定时间的方式举例

- 指定在今天下午 5:30 执行某命令（假设现在时间是中午12:30，2011年4月16）
- 命令格式如下：
  - at 5:30pm
  - at 17:30
  - at 17:30 today
  - at now + 5 hours
  - at now + 300 minutes
  - at 17:30 16.4.2011
  - at 17:30 4/16/2011
  - at 17:30 Apr 16

梁如军（linuxbooks@126.com）

at命令使用范例

- 交互方式

  $ at 4:00 6/20/2001

- 使用命令文件方式
  - 生成文件myatjob：

  $ echo “find / -name *.txt > ~/txtfiles”> myatjob
  - 使用at命令

  $ at -f myatjob 4:00 6/20/2011
  - or

  $ at < myatjob 4:00 6/20/2011