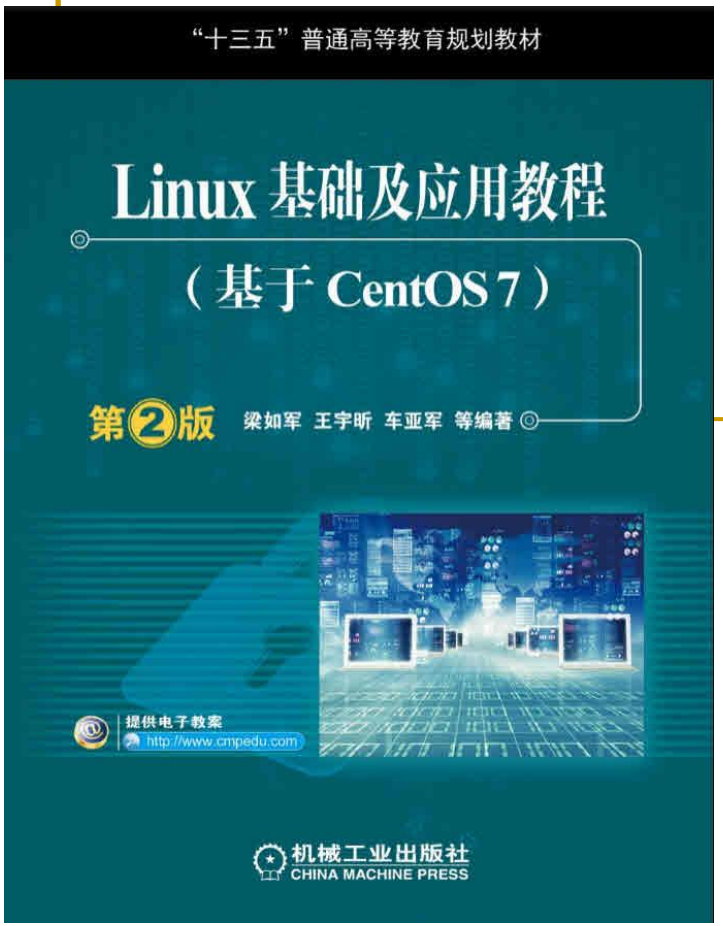
CentOS
The Community ENTerprise Operating System



第8章
服务器安全基础

主讲人： 梁如军

2015-05-05

本章内容要点

- 基本的系统安全
- 物理安全和登录安全
- 禁用root登录和sudo
- 口令安全和口令策略
- SSL与OpenSSL
- TCP Wrappers

梁如军（linuxbooks@126.com）

本章学习目标

- 掌握基本的系统安全配置
- 掌握物理安全和登录安全配置
- 掌握sudo的配置和sudo命令的使用
- 理解PAM机制及其执行过程
- 学会使用PAM模块增强口令安全
- 理解SSL协议体系结构
- 掌握自签名证书和私钥的创建过程
- 掌握TCP Wrappers的配置

梁如军（linuxbooks@126.com）

Linux 服务器安全的一般性原则

- 最少的正在运行的应用程序
  - 安装和运行最少的软件，以尽量减少漏洞。
  - 保持软件更新。
- 开放所需的最少端口
  - 关闭不必要的服务
  - 在单独的系统上运行不同的网络服务。最大限度地减少风险，避免一个服务的问题影响到其他服务。
- 最小特权:
  - 为用户帐户和软件执行任务赋予所必需的最低权限。

梁如军（linuxbooks@126.com）

Linux 服务器安全的一般性原则（续）

- 维护用户帐户
  - 创建一个良好的密码策略，并强制执行。
  - 最少的必要账户，删除已不使用的用户帐户。
- 配置系统备份
  - 制定灾难恢复计划并检测备份的有效性
- 启用 远程/集中式 系统日志（定期复查日志！）
  - 发送日志到一个专用的日志服务器
  - 这可以防止入侵者轻易地修改本地日志

梁如军（linuxbooks@126.com）

Linux 服务器安全的一般性原则（续2）

- 加密传输数据
  - 加密认证信息（如密码）显得尤为重要
  - 使用密钥认证的远程登录服务（ssh）
  - 使用SSL/TLS协议加密应用协议（https, ftps, etc）
- 配置网络访问控制
  - TCP Wrappers and xinetd
  - Netfilter/iptables etc.
- 配置安全工具以提高系统的鲁棒性

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

系统安全

梁如军（linuxbooks@126.com）

基本系统安全

- 安全的磁盘布局
- 使用挂装选项提高文件系统的安全性
- 查找并取消文件/目录的非必要的特殊权限
- 避免安装不必要的软件包
- 配置软件包更新的Email通知
- 关闭不必要的服务

梁如军（linuxbooks@126.com）

磁盘布局

- /目录中必须包括 /etc、/lib、/bin、/sbin，即不能在此四个目录上使用独立的分区或逻辑卷
- 除了 / 、/boot 和 SWAP 之外您应该根据自己的需要尽量分离数据到不同的分区或逻辑卷
- 建议创建独立的 /usr、/var、/tmp、/var/tmp 文件系统
- 根据日志管理需要，您可能应该创建独立的 /var/log、/var/log/audit 文件系统
- 若所有普通用户数据存储在本机，您还应该创建独立的 /home 文件系统
- 若系统对外提供大量服务（如Web虚拟主机等），应该创建独立的 /srv 文件系统

梁如军（linuxbooks@126.com）

提高文件系统的安全性

- 常用如下三个挂装参数提高文件系统的安全性
  - noexec：不允许在本分区上执行二进制程序，即防止执行二进制程序但允许脚本执行
  - nodev：不解释本分区上的字符或块设备，即防止用户使用设备文件
  - nosuid：不允许在本分区上执行 SUID/SGID 的访问
- 例如

```
/dev/sda5   /srv/www     ext3    defaults,nosuid,nodev,noexec   1 2
/dev/sda6   /var/ftp     ext3    defaults,nosuid,nodev,noexec   1 2
/dev/sda7   /tmp         ext3    defaults,nosuid,nodev,noexec   1 2
```


梁如军（linuxbooks@126.com）

去除非必要的特殊权限

- 查找系统中权限为SUID/SGID 的文件

  # find / -xdev \(-perm -4000 -o -perm -2000 \) -ls

- 查找指定目录/dir下的所有用户可写的设置了粘着位（sticky-bit）的目录

  # find /dir -xdev -type d \(-perm -0002 -a ! -perm -1000 \) -ls

使用 chmod 命令去除文件或目录的非必要的特殊权限
chmod u-s
chmod g-s
chmod o-t

梁如军（linuxbooks@126.com）

检查重要的文件权限

- 查找指定目录/dir下的所有用户 可写的文件

  # find /dir -xdev -type f -perm -0002 -ls
- 查找指定目录/dir下的非有效用户或有效组的文件

  # find /dir -xdev \(-nouser -o -nogroup \) -ls

使用 chmod 命令修改权限
chmod o-w
使用 chown 命令修改文件属主或组
chown USER:GROUP

梁如军（linuxbooks@126.com）

避免安装不必要的软件包

- 在安装过程中仅仅安装必要的软件包，即**使用最小化安装**
- 使用如下命令查找、删除不必要的软件包

  # yum list installed

  # yum remove PackageName
- 通常服务器无需运行X系统，尤其是被托管的服务器。
- 在系统运行过程中，可以安装需要的软件包

  # yum install PackageName

梁如军（linuxbooks@126.com）

软件包的更新

- 保持系统中软件包的更新极为重要
  - 当软件的编制者发现软件的漏洞之后将对其进行修复，修复后的软件包就会发布到相应的yum仓库中
- 手动更新

  # yum check-update

  # yum -y update

  # yum -y update-minimal
- 自动更新
  - 启用yum-cron服务

梁如军（linuxbooks@126.com）

配置软件包更新的Email通知

- 安装 yum-cron

  # yum -y install yum-cron

- 配置 yum-cron

  # vi /etc/sysconfig/yum-cron

  update_cmd = security

  emit_via = email

  MAILTO = 13912345678@139.com

- 启动 yum-cron

  # systemctl enable yum-cron.service

  # systemctl start yum-cron.service

梁如军（linuxbooks@126.com）

关闭不必要的服务

- 查看已启动的服务

  # systemctl list-unit-files |grep enabled|grep .service
- 使用systemctl disable命令关闭不必要的服务
- 可以编写自己的脚本文件关闭不必要的服务

梁如军（linuxbooks@126.com）

物理安全和登录安全

- 设置计算机BIOS
- 配置GRUB的口令
- 禁用三键重启热键
- 设置屏幕锁定
- 为 BASH 设置超时自动注销

梁如军（linuxbooks@126.com）

设置计算机BIOS

- 禁止附加存储介质启动系统
  - 如：光驱、USB等
- 设置BIOS修改口令

生成 GRUB 口令

- GRUB可以允许用户绕过所有的安全验证而进入调试Shell模式
- 管理员应该设置GRUB口令避免修改启动参数从而提供安全性
- 为GRUB生成口令

grub2-mkpasswd-pbkdf2

Password:<*ENTER-YOUR-PASSWORD*>

Retype password:<*ENTER-YOUR-PASSWORD*>

PBKDF2 hash of your password is

grub.pbkdf2.sha512.10000.016088B35F8C0326EC214BD06C42B50039AE144EA311DFD1011029E598FA87B585B1E12323D86C84CE7CB635254F010D0C1B3BCBF6478AF4C235C826205BA62A.CFA807484702FA7A6FF52A7D3DB66E7D2D2E17D583AA8BDC108DF3BA075079AA4B4140AD5166E3AF276F7F0

启用GRUB配置文件的口令

- 编辑/etc/grub.d/00_header,在其尾部添加如下行

```
cat <<EOF
set superusers="YourName"
password_pbkdf2 YourName
grub.pbkdf2.sha512.10000.016088B35F8C0326EC214BD06C42B500
39AE144EA311DFD1011029E598FA87B585B1E12323D86C84CE7CB
635254F010D0C1B3BCBF6478AF4C235C826205BA62A.CFA8074847
02FA7A6FF52A7D3DB66E7D2D2E17D583AA8BDC108DF3BA075079
AA4B4140AD5166E3AF276F7F0B92D491D6F242F5DB79B3C9FFB3
358A2734B27A86
```

- 更新配置文件grub.cfg

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```


梁如军（linuxbooks@126.com）

禁用重启热键

- 在 RHEL/CentOS 中默认情况下可以通过键盘热键 <Ctrl>+<Alt>+<Delete> 重启系统
- 为了提高安全性禁用重启热键

  # systemctl mask control-alt-delete.servie

梁如军（linuxbooks@126.com）

设置屏幕锁定

- 安装vlock包

  # yum -y install vlock

- 使用vlock命令锁定屏幕
  - 锁定当前屏幕：

    $ vlock

  - 锁定所有已登录的终端会话并禁止虚拟控制台切换：

    $ vlock -a

为 BASH 设置超时自动注销

```
# vi /etc/profile.d/autologout.sh
TMOUT=300  # 5分钟后超时
readonly TMOUT
export TMOUT

# chmod +x /etc/profile.d/autologout.sh
```


梁如军（linuxbooks@126.com）

禁止ROOT账号登录

梁如军（linuxbooks@126.com）

Root 账号

- 尽量减少使用 root 账号
  - 是整个 Linux 系统的关键
  - 如果可能，使用其他帐户
- 除非绝对必要，否则不要以 root 直接登录
  - 使用“*su -*” 替换 root 的直接登录
  - 使用 *sudo*

sudo简介

- sudo 允许授权用户以**超级用户**或**其他用户**身份执行命令
  - sudo 能限制指定用户在指定主机上运行某些命令
  - 默认情况下，只有root用户可以使用sudo命令
- sudo 使用普通用户自己的口令
  - sudo 是设置了SUID权限位的执行文件
- sudo 使用时间戳文件来完成类似“检票”的系统
  - 当用户执行 sudo 时，**5分钟** 内不用再输入口令
- 记录所有登录（包括成功的和不成功的登录）
  - /var/log/secure

与sudo相关的文件

CentOS
The Community ENTerprise Operating System

- /usr/bin/sudo：以其他用户身份执行命令
- /etc/sudoers：sudo的配置文件
  - 列出哪（些）个用户在哪个（些）主机上执行哪个（些）命令
- /usr/sbin/visudo：用于编辑 sudoers 文件
  - 防止两个用户同时进行修改
  - 进行有限的语法检查
- /var/run/sudo/目录：包含用户的时间戳文件

梁如军（linuxbooks@126.com）

sudo 的执行过程

- 在 /var/run/sudo/$USER目录中查找时间戳文件
  - 若时间戳已过期，提示用户输入自己的口令
    - 若口令输入错误则退出 sudo 的执行
    - 若口令输入正确则继续 sudo 的执行过程
  - 若时间戳未过期，继续 sudo 的执行过程
- 读取配置文件 /etc/sudoers，判断用户是否具有执行此 sudo 命令的权限
  - 若有权限执行则执行 sudo 后面的命令
  - 若无权执行则退出 sudo 的执行

快速配置sudo

- 将所有需要使用sudo的普通用户添加到 wheel 组中

  # usermod -G wheel osmond

  # usermod -G wheel jason

- 配置允许wheel组可以执行sudo命令

  # visudo

  // 删除如下行的注释符，之后保存退出

  %wheel ALL=(ALL) ALL

梁如军（linuxbooks@126.com）

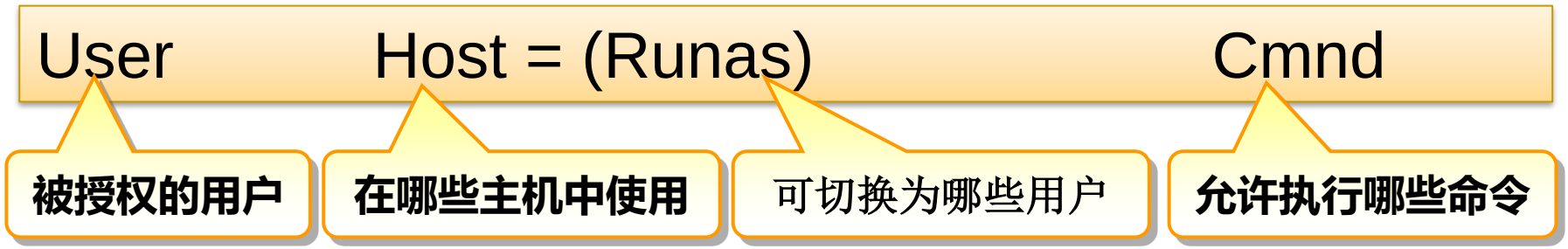
su与sudo比较

CentOS
The Community ENTerprise Operating System

- su
  - 直接切换为超级用户
  - 普通用户要切换为超级用户必须知道超级用户的口令
  - 适用于系统中只有单个系统管理员的情况
- sudo
  - 直接使用 sudo 命令前缀执行系统管理命令
  - 执行系统管理命令时无需知道超级用户的口令
  - 适用于系统中有多个系统管理员的情况
    - 允许 root 为几个用户或组委派权利，使之能运行部分或全部由 root （或另一个）用户执行的命令
    - sudo 设计者的宗旨：给用户尽可能少的权限但仍允许完成他们的工作
    - 当然系统只有单个系统管理员时也可以使用

梁如军（linuxbooks@126.com）

/etc/sudoers 的配置语法



- (Runas)部分可以省略，省略时表示(root)
- 四个部分均可设置**多个项目**，每个项目用**逗号间隔**。
- 在User部分使用“%组名”的形式为组中的所有用户授权。
- 在Cmnd部分
  - 使用**NOPASSWD:**前缀参数，表示不用输入口令即可执行Cmnd。
  - 使用**！前缀**表示逻辑非。使用**Shell通配符** 匹配可以执行的命令。
- **ALL**表示所有。以#开始的行为注释行。行末的\为续行符。

梁如军（linuxbooks@126.com）

/etc/sudoers 的配置语法续

- 定义四种别名
  - User_Alias
  - Host_Alias
  - Runas_Alias
  - Cmnd_Alias
- 语法

User_Alias *USER_ALIAS_NAME* = user1, user2, ...
Host_Alias *HOST_ALIAS_NAME* = host1, host2, ...
Runas_Alias *RUNAS_ALIAS_NAME* = runas1, runas2, ...
Cmnd_Alias *COMMAND_ALIAS_NAME* = cmnd1, cmnd2, ...

- 使用别名简化授权

/etc/sudoers 的配置举例1

- 专职系统管理员(millert,mikef和dowdy)可以在任何主机上以root用户身份执行任何命令而不需要进行身份验证。

```
User_Alias FULLTIMERS = millert, mikef, dowdy
FULLTIMERS ALL = NOPASSWD: ALL
```


梁如军（linuxbooks@126.com）

/etc/sudoers 的配置举例2

- 兼职管理员(jalala, sonar和huge)可以在任何主机上以root用户身份运行别名 BROWSE、PROCESSES、USERS 中定义的命令，同时可以修改除了 root 用户之外的所有用户口令。

```
User_Alias      PARTTIMERS2 = jalala, sonar , huge
Cmnd_Alias      BROWSE = /bin/ls, /bin/cd, /bin/cat
Cmnd_Alias      PROCESSES = /bin/nice, /bin/kill, /usr/bin/killall
Cmnd_Alias      USERS = /usr/sbin/useradd [A-z]*,/usr/sbin/userdel -r [A-z]*

PARTTIMERS2  ALL= USERS,PROCESSES,BROWSE, /usr/bin/passwd
[A-z]*, !/usr/bin/passwd root
```


梁如军（linuxbooks@126.com）

sudo命令

CentOS
The Community ENTerprise Operating System

```
sudo -V | -h | -k | -l | -v
sudo [-Hb] [-u username|#uid] { -i | -s | <command> }
```

- -H：将环境变量中的$HOME指定为要变更身份的使用者的目录（如不加-u参数就是/root）。
- -b：在后台执行指令。
- -u username|#uid :以指定的用户作为新的身份。省略此参数表示以root的身份执行指令。
- -i：模拟一个新用户身份的初始Shell。
- -s：执行环境变量$SHELL所指定的Shell，或是/etc/passwd里所指定的Shell。

梁如军（linuxbooks@126.com）

sudo命令举例

- 查看当前用户可以使用sudo执行的命令列表

  $ sudo -l

- 进入交互式（ interactively）登录，类似“su -”

  $ sudo -i

- 直接执行授权的命令

  $ sudo /usr/sbin/useradd otheruser

  $ sudo /usr/bin/passwd otheruser

  $ sudo sh -c "cd /home ; du -s * | sort -rn > USAGE"

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

可插拔认证模块（PAM）

梁如军（linuxbooks@126.com）

PAM简介

- PAM（Pluggable Authentication Modules）
  - http://www.kernel.org/pub/linux/libs/pam/index.html
  - 本地文档：/usr/share/doc/pam-<*version*>/
- **PAM 是一系列被应用程序共享的可配置的动态加载的用于用户验证和授权的库文件**
  - PAM 将应用程序的验证机制从应用程序中独立出来
  - 应用程序调用PAM来实现用户验证和授权
  - PAM 独立的模块化实现便于功能扩展和维护
  - PAM 为认证模块的编制建立了标准的API，以便各应用程序能方便地使用
  - PAM 验证机制对其上层的应用程序和最终用户都是透明的

梁如军（linuxbooks@126.com）

PAM的组成

- PAM由PAM核心和PAM模块组成
- PAM核心
  - 负责调用PAM模块
  - **/lib/libpam.so**
- PAM模块
  - 用于实现各种验证的动态可加载的库文件
  - 对每个 PAM 模块都会执行 **通过** 或 **失败** 的测试
  - PAM 模块保存在 **/lib/security** 目录中
  - 一些 PAM 模块需要模块的配置文件，模块配置文件通常保存在 **/etc/security** 目录下

梁如军（linuxbooks@126.com）

PAM 客户

- PAM客户：使用PAM验证功能的应用程序
- 判断应用程序是否为PAM客户

  # ldd /bin/login|grep libpam
- 查询系统已安装的PAM客户

  # rpm -q --whatrequires pam
- PAM客户配置文件
  - 每个PAM客户都可以调用不同的PAM模块实现验证
  - 决定了PAM模块在什么时候如何被PAM客户使用
  - 保存在 **/etc/pam.d** 目录下

梁如军（linuxbooks@126.com）

PAM客户调用PAM的过程

- PAM客户（如：login）调用PAM核心 **/lib/libpam.so**
- PAM核心收到来自应用程序的请求后在**/etc/pam.d**目录下查找与应用程序同名的配置文件（如：/etc/pam.d/ login）
- PAM核心根据配置文件的设置执行指定的PAM模块
  - 在执行一些模块（如：pam_access）时，还会读取**/etc/security**目录下相应的模块配置文件（如：/etc/security/access.conf）
- PAM核心接收每一个PAM模块的执行结果（或成功或失败），根据PAM模块的返回结果和配置文件的设置决定验证是否通过
- PAM核心将最终结果返回给调用它的应用程序，应用程序根据返回结果决定验证是否通过

PAM客户的配置文件

- /etc/pam.d/*
  - 每个配置文件均使用同样的语法
- 配置文件中每一行的格式为：

  模块类型　控制标记　模块路径　执行参数
  - **模块类型（module-type）**：指定模块的测试类型
  - **控制标记（control-flag）**：决定PAM如何使用模块调用后的返回值
  - **模块路径（module-path）**：指定相对于/lib/security/目录的模块文件名
  - **执行参数（module-arguments）**：指定模块参数，多个参数之间用空格间隔（可选项）

梁如军（linuxbooks@126.com）

PAM的模块类型

- PAM的四种模块类型
  - **auth** 验证某用户的确是这个用户
  - **account** 批准某账户的使用
  - **password** 控制密码的修改
  - **session** 打开、关闭、并记录会话
- 每一类都会在需要时被调用，并为服务提供独立的结果

梁如军（linuxbooks@126.com）

PAM的控制标志

CentOS
The Community ENTerprise Operating System

- 模块可以被堆叠 (同种类型的模块按先后顺序依次执行)
- 控制标志决定每个类型测试将如何影响同类型的测试以及最终的测试结果
  - **required**：必须通过，若失败则继续测试
  - **requisite**：与 required 类似，不同之处是它在失败后停止测试
  - **sufficient**：如果到此为止一直通过，现在就返回成功；如果失败，忽略测试，继续检查
  - **optional**：测试通过与否都无关紧要
  - **include**：返回在被调用的文件中配置的测试的总体测试值

梁如军（linuxbooks@126.com）

控制标志的复杂语法

[value1=action1 value2=action2 ...]

- valueN可以是：
  - PAM函数库中任何一个函数的返回值，如：
    - new_authtok_reqd、user_unknown
    - success、Ignore、default 等
- actionN可以是：
  - **ignore**：忽略此测试的值，即此值对最终测试结果无关紧要
  - **bad**：这个返回值应该被看作整个层叠模块验证失败
  - **ok**：这个返回值直接作为整个层叠模块的返回值
  - **非负整数**：表示需要忽略后面堆叠的n个同类型的模块

梁如军（linuxbooks@126.com）

控制标志的两种语法的关系

- required 等价于
  - **[success=ok new_authtok_reqd=ok ignore=ignore default=bad]**
- requisite 等价于
  - **[success=ok new_authtok_reqd=ok ignore=ignore default=die]**
- sufficient 等价于
  - **[success=done new_authtok_reqd=done default=ignore]**
- optional 等价于
  - **[success=ok new_authtok_reqd=ok default=ignore]**

梁如军（linuxbooks@126.com）

示例：/etc/pam.d/login

```
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
auth       substack     system-auth
auth       include      postlogin
account    required     pam_nologin.so
account    include      system-auth
password   include      system-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_console.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      system-auth
session    include      postlogin
-session   optional     pam_ck_connector.so
```


梁如军（linuxbooks@126.com）

共享验证文件：system-auth

- system-auth 为许多 PAM客户 提供了一个“通用的全局配置”
- system-auth 包含了若干标准验证测试
- system-auth 由 **include** 控制标志调用
- system-auth 对可共享的标准系统验证实现简单、统一的管理
- 修改 system-auth 配置文件可以同时影响多个PAM客户的验证

梁如军（linuxbooks@126.com）

/etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 1000 quiet
account     required      pam_permit.so

password    requisite     pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type=
password    sufficient    pam_unix.so md5 shadow nullok try_first_pass use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
```


梁如军（linuxbooks@126.com）

常用的PAM模块（1）

- **pam_env**：环境变量初始化
- **pam_unix**：标准UNIX的用户认证
- **pam_pwquality**：强制使用强壮的口令
- **pam_tally[2]**：记录用户的失败登录和账户锁定
- **pam_nologin**：若 **/etc/nologin** 存在，则除了root 用户之外的任何用户都不能登录
- **pam_securetty**：root 用户只能使用 **/etc/securetty** 文件中指定的终端设备登录
- **pam_mkhomedir**：当用户主目录不存在时创建其主目录

梁如军（linuxbooks@126.com）

常用的PAM模块（2）

- **pam_access**
  - 用一个文本配置文件完成基于用户、组、以及来源的访问限制
  - 配置文件：**/etc/security/access.conf**
  - **# man pam_access**
- **pam_listfile**
  - 允许用户针对**某一服务**单独建立文件来配置基于用户、组、本地终端、远端主机的限制
  - 配置文件由模块参数指定，配置文件包含被允许或拒绝的账户列表
  - **# man pam_listfile**

梁如军（linuxbooks@126.com）

常用的PAM模块（3）

- **pam_limits**
  - 使用一个文本配置文件，限制授权用户可使用的资源
  - 配置文件： **/etc/security/limits.conf**
  - **# man pam_limits**
- **pam_time**
  - 使用一个文本配置文件，配置基于时间的服务访问限制
  - 配置文件：**/etc/security/time.conf**
  - **# man pam_time**

梁如军（linuxbooks@126.com）

查看PAM模块及其手册

- 查找当前系统已安装的PAM模块手册

  # man -k pam_
- 查看某PAM模块的手册

  # man pam_XXX
- 查找使用了指定模块的PAM配置文件

  # grep -l pam_XXX /etc/pam.d/*

梁如军（linuxbooks@126.com）

PAM 故障排除

- 检查系统日志
  - /var/log/message
  - /var/log/secure
- PAM错误可能会使root用户被拒之门外
  - 在测试 PAM 时打开一个 root shell
  - 用单用户模式绕过 PAM
  - 使用救援光盘引导系统

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

口令策略与口令安全

梁如军（linuxbooks@126.com）

口令策略

- 口令必须保持私有
- 一个口令在使用 60 天后必须更换
- 避免使用最近使用过的5个口令
- 口令必须符合复杂性要求
  - 不包含用户的账户名、超过两个连续字符的用户全名
  - 至少12个字符的长度
  - 至少包含英文大写字符、英文小写字符、十进制数字和非字母字符（例如： !, $, #, %）四类字符中的三类

Linux口令安全

- 散列存储机制
  - CentOS 7中默认使用带加盐的sha512哈希算法生成系统用户的口令
- 影子口令机制
  - CentOS 7默认将用户口令保存在只能被 root 查看的 /etc/shadow 文件中
  - 基于/etc/shadow文件的时间字段实现了口令时效
- 基于 PAM 的账号保护和口令策略实施

梁如军（linuxbooks@126.com）

口令安全与口令策略

- **散列口令**：使口令更难被破译（默认设置）
  - 使用带有 **sha512**（sha256、md5）参数的 **pam_unix.so**
- **影子口令**：实现口令时效，且 /etc/shadow 只能被 root 查看（默认设置）
  - 使用带有 **shadow** 参数的 **pam_unix .so**
- **口令历史**：避免重复使用最近几次设置过的口令
  - 使用带有 **remember=N** 参数的 **pam_unix.so**
- **口令强度**：限制口令中可用的字符及数目
  - 使用带有参数的 **pam_pwquality.so**
- **账户锁定**：记录失败的登录并在N次失败后锁定
  - 使用带有参数的 **pam_tally2.so**

pam_unix.so可用的验证类型

- **auth**：验证用户密码的有效性
  - 可用的模块参数：debug、audit、use_first_pass、try_first_pass、nullok和nodelay
- **account**：检查密码是否过期
  - 可用的模块参数： debug、audit
- **password**：处理本地文件或NIS中的密码修改
  - 可用的模块参数： debug、 audit、 nullok、nis、md5、not_set_pass、 use_authtok、 try_first_pass、use_first_pass、bigcrypt、shadow、remember
- **session**：将登录和注销事件记录到日志中

pam_unix.so的常用参数

- **debug**：将调试信息写入系统日志
- **audit**：提供比debug更多诊断调试信息
- **nullok**：允许口令为空的用户登录系统
- **sha512**：采用sha512哈希算法
- **shadow**：采用shadow口令机制
- **remember=n**：
  - 避免用户重复使用最近n次设置过的口令
  - 会将n个使用过的旧密码，以哈希方式加密后保存到**/etc/security/opasswd** 文件中

pam_unix.so的常用参数续

- try_first_pass：
  - **与auth验证类型一起使用时**，从之前的 auth 类型来取得用户口令，若口令不符合或未输入则要求重新输入一次。
  - **与password验证类型一起使用时**，该选项主要用来防止用户新设定的密码与以前的旧密码相同。
- use_first_pass：
  - **与auth验证类型一起使用时**，从之前的 auth 类型来取得使用者口令，若口令不符合或未输入则认为认证失败。
  - **与password验证类型一起使用时**，该选项主要用来防止用户新设定的密码与前面password提供的口令相同。
- use_authok：
  - **与password验证类型一起使用时**，使用此选项强制用户使用前面堆叠验证模块提供的密码，例如由 pam_cracklib 验证模块提供的新密码。

梁如军（linuxbooks@126.com）

使用 pam_unix.so 模块加强口令安全

- **/etc/pam.d/system-auth**

auth sufficient pam_unix.so nullok try_first_pass

account required pam_unix.so

password sufficient pam_unix.so **sha512 shadow** nullok try_first_pass use_authtok **remember=5**

session required pam_unix.so

梁如军（linuxbooks@126.com）

用户口令的强壮性检查——pam_pwquality.so模块参数

- **minlen=N**：最小口令长度。
- **difok=N**：新口令有几个字符不能和旧口令相同，默认是5或有一半的字符与旧口令不同。
- **dcredit=N**：当N>=0时，N代表新口令最多可以有多少个阿拉伯数字。当N<0时，N代表新口令最少要有多少个阿拉伯数字。
- **ucredit=N**：与dcredit书写规则类似，但此处指大写字母。
- **lcredit=N**：与dcredit书写规则类似，但此处指小写字母。
- **ocredit=N**：与dcredit书写规则类似，但此处指特殊字符。

使用pam_cracklib.so模块设置口令策略

- **/etc/pam.d/system-auth**

password requisite pam_pwquality.so try_first_pass local_users_only retry=3 authtok_type= **minlen=12 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1**

- 或者写入模块配置文件/etc/security/pwquality.conf

```
# echo '
minlen=12
dcredit=-1
ucredit=-1
ocredit=-1
lcredit=-1
' >> /etc/security/pwquality.conf
```


梁如军（linuxbooks@126.com）

pam_tally2.so模块

- 设置在登录失败若干次之后锁定账户
- 将用户失败的登录次数记录于二进制文件/var/log/tallylog
- 管理员可以使用如下命令将某用户的失败登录计数器清零从而解除账户锁定

  # /sbin/pam_tally2 --user <username> --reset
- 常用的模块参数
  - **onerr=fail**：当登录失败时
  - **deny=n**：n次之后禁止登录

梁如军（linuxbooks@126.com）

使用pam_tally2.so模块 设置登录失败后的账户锁定

- **/etc/pam.d/system-auth**

auth required pam_tally2.so **deny=5 onerr=fail even_deny_root unlock_time=1200**

梁如军（linuxbooks@126.com）

基于PAM的访问控制

- 登录访问控制——**pam_access.so**
- 列表访问控制——**pam_listfile.so**
- 时间访问控制——**pam_time.so**
- 资源访问控制——**pam_limits.so**

梁如军（linuxbooks@126.com）

pam_access.so模块简介

- 通过 **用户及其来源** 实现登录访问控制
  - 根据指定的 (user, host) 允许或拒绝用户的**网络登录**
  - 根据指定的 (user, tty)允许或拒绝用户的**本地登录**
- 可用的模块测试类型
  - auth, account, password and session
- 默认的模块配置文件
  - /etc/security/access.conf

  # man 5 access.conf
- 常用模块参数
  - accessfile=<filename>
    - 指定与默认模块配置文件分离的其他配置文件

梁如军（linuxbooks@126.com）

pam_access的模块配置文件

CentOS The Community ENTerprise Operating System

权限：用户：来源

- 权限（permission）
  - **+**代表允许，**-**代表拒绝
- 用户（users）
  - 以空格间隔的用户名或组名，**ALL**表示所有用户
- 来源（origins）
  - **ttyX** 表示非远程登录， **LOCAL** 表示任何本地登录
  - somehostname、FQDN、**.**somedomain.com
  - 192.168.0.1、192.168.0**.**、192.168.0.0**/24**
  - **ALL** 表示所有来源，**EXCEPT** 表示除了…之外

梁如军（linuxbooks@126.com）

pam_access.so举例（1）

禁止 **root** 用户从 **tty2** 上登录，
用户 **jjheng** 可以在除 **tty4** 本地终端之外的所有终端登录。

- 修改login的PAM配置文件 **/etc/pam.d/login**

  account required pam_nologin.so

  account required pam_access.so

  ……

- 修改配置文件 **/etc/security/access.conf**

  - : root : tty2

  - : jjheng : LOCAL EXCEPT tty4

梁如军（linuxbooks@126.com）

pam_access.so举例（2）

限制 root 用户只能从 centos.ls-al.me 主机上使用 ssh 登录本系统，
限制 root 用户只能从 192.168.0.0/24上使用 ssh 登录本系统，
禁止所有除了 osmond 之外的用户使用 ssh 登录本系统。

- 修改sshd的PAM配置文件 **/etc/pam.d/sshd**

```
account    required     pam_nologin.so
account    required     pam_access.so
    accessfile=/etc/ssh/sshd_access.conf
.............
```

- 修改配置文件 **/etc/ssh/sshd_access.conf**

```
- : root : ALL EXCEPT centos.ls-al.me
- : root : ALL EXCEPT 192.168.0.0/24
- : ALL EXCEPT osmond : ALL
```


梁如军（linuxbooks@126.com）

pam_listfile.so模块简介

- 使用允许/禁止列表实现访问控制
  - 基于 **user** 的允许/禁止列表
  - 基于 **group** 的允许/禁止列表
  - 基于 tty|rhost|ruser|shell 的允许/禁止列表
- 可用的模块测试类型
  - auth, account, password and session
- 模块配置文件
  - 存放同类型的项目，如全部为用户名或全部为组名
  - 每一行存放一个项目

梁如军（linuxbooks@126.com）

pam_listfile.so的模块参数

- **item=[tty|user|rhost|ruser|group|shell]**
  - 设置访问控制的对象类型
- **file=/path/to/filename**
  - 指定保存有“item”对象的文件位置
- **sense=allow|deny**
  - 在文件**file**中找到指定的 item 对象时的动作（或允许或拒绝）
  - 如果在文件中找不到相应的item对象，则执行相反的动作
- **onerr=succeed|fail**
  - 指定当某类事件发生时的返回值（或成功或失败）
- **apply=[user|@group]**
  - 指定规则所适用的对象
  - 只有当 item=[tty|rhost|shell] 时才有意思

梁如军（linuxbooks@126.com）

pam_listfile.so举例（1）

Vsftpd 的默认配置：拒绝/etc/vsftpd/ftpusers中列出的用户登录ftp

- /etc/pam.d/vsftpd

  session optional pam_keyinit.so force revoke

  auth required pam_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed

  ……

- /etc/vsftpd/ftpusers

  root

  bin

  ……

梁如军（linuxbooks@126.com）

pam_listfile.so举例（2）

拒绝 /etc/ssh/sshd.deny 中列出的用户 **ssh** 登录

- 编辑 **/etc/pam.d/sshd**

  auth include system-auth

  auth required pam_listfile.so item=user sense=deny onerr=succeed file=/etc/ssh/sshd.deny

  ……

- 编辑 **/etc/ssh/sshd.deny**

  root

  jjheng

  ……

梁如军（linuxbooks@126.com）

pam_listfile.so举例（3）

使用 su 时只能切换为 /etc/security/su.ok 中列出的用户

- 编辑 /etc/pam.d/su

```
auth   sufficient    pam_rootok.so
auth   required    pam_listfile.so   item=user  onerr=fail
    sense=allow   file=/etc/security/su.ok
……
```

- 编辑 /etc/security/su.ok

```
root
tom
……
```


梁如军（linuxbooks@126.com）

pam_time.so模块简介

- 实现基于时间的登录访问控制
  - **通过 login 限制本地登录的访问时间**
  - **通过 sshd 限制网络登录的访问时间**
- 可用的模块测试类型
  - account
- 默认的模块配置文件
  - /etc/security/time.conf

 # man 5 time.conf

梁如军（linuxbooks@126.com）

pam_time的模块配置文件

服务；终端；用户； 时间

- **服务、终端、用户 是逻辑表达式的列表**
  - “**!**”表示非；“**|**”表示或；“**&**”表示与
- 服务（services）—— 即PAM客户
  - 应用PAM功能的服务名称（如：**login**、**sshd**）
- 终端（ttys）—— 应用此规则的终端名
  - “*”表示任何终端
- 用户（users） —— 应用此规则的用户名
  - “*”表示任何用户

梁如军（linuxbooks@126.com）

pam_time的模块配置文件续

- 时间（times） —— 应用此规则的时间范围
  - 用星期几表示日期
    - **Mo、Tu、We、Th、Fr、Sa、Su（周1~周日）**
    - **Wk指每一天，Wd指周末，Al也指每一天**
    - MoTuSa：指星期一星期二和星期六
    - AlFr：指除星期五外的每一天
  - 时间使用24小时制
    - **HHMM- HHMM** 的形式
    - Al1800-0800：指每天下午6点整到第二天的早晨8点整
  - 可用“**!**”表示除此以外的所有日期/时间
  - “**|**”表示或

梁如军（linuxbooks@126.com）

pam_time.so举例（1）

限制用户本地登录的登录时间

- 修改配置文件 **/etc/pam.d/login**

  account include system-auth

  account required pam_time.so

- 编辑配置文件 **/etc/security/time.conf**

  # 允许fanny和david在周1、3、5早9点到晚10点登录本机

  login ; tty* ; fanny|david ; MoWeFr0900-2200

  # 允许以student开始的用户每天早8点到晚6点登录本机

  login ; tty* ; student* ; Wk0800-1800

  # 禁止所有普通用户登录本机

  login ; tty* ; !root ; !Al0000-2400

梁如军（linuxbooks@126.com）

pam_time.so举例（2）

限制用户ssh远程登录的登录时间

- 修改配置文件 **/etc/pam.d/sshd**

  account include system-auth

  account required pam_time.so

- 编辑配置文件 **/etc/security/time.conf**

  # 允许用户osmond每天0点到晚11点登录本机

  sshd ; * ; osmond ; Al0000-2300

  # 允许所有以student开始的用户每天早8点到晚6点登录本机

  sshd ; * ; student* ; Wk0800-1800

  # 允许所有普通用户每天晚6点到次日早8点或周末全天登录

  sshd ; * ; !root ; Wd0000-2400 | Wk1800-0800

梁如军（linuxbooks@126.com）

pam_limits.so模块简介

- 实现资源访问的限制
  - 限制用户或组的同时登录数
  - 限制用户在会话过程中对系统资源的使用
- 可用的模块测试类型
  - session
- 默认的模块配置文件
  - /etc/security/limits.conf

 # man 5 limits.conf

梁如军（linuxbooks@126.com）

pam_limits的模块配置文件1


```
<domain>      <type>  <item>  <value>
```

- Domain：指定被限制的用户名或组名
  - 组名前面加@以示与用户名区别
  - 通配符 * 表示所有用户
- Type：指定限制类型
  - **soft** 指的是当前系统生效的设置值
  - **hard** 表明系统中所能设定的最大值
    - 可以超出软限制（警告），但不能超过硬限制（soft < hard ）
  - 用 **-** 就表明同时设置 soft 和 hard 的值
- Value：指定限制值

梁如军（linuxbooks@126.com）

pam_limits的模块配置文件2


```
<domain>      <type>  <item>  <value>
```

- Item：指定被限制的资源项目
  - CPU
    - **cpu**：最大的CPU占用时间（分钟）
    - **nproc**：最大进程数
  - 文件
    - **fsize**：最大的文件大小(KB)
    - **nofile**：最大可以打开的文件数量
    - **locks**： 最大可锁定文件的数目

梁如军（linuxbooks@126.com）

pam_limits的模块配置文件3


```
<domain>      <type>  <item>  <value>
```

- Item：指定被限制的资源项目
  - 内存
    - **data**：最大的数据段大小(KB)
    - **stack**：最大的堆栈段大小(KB)
    - **rss**：最大的可驻留空间(KB)
    - **memlock**：最大锁定内存地址空间大小(KB)
    - **msqqueue**：POSIX信息队列的最大可使用的内存(bytes)
  - 登录数
    - **maxlogins**：该用户可以登录到系统的最多次数
    - **maxsyslogins**：系统能够同时登录的最大用户数

梁如军（linuxbooks@126.com）

pam_limits.so举例（1）

限制用户/组、系统的最大登录数

- cat /etc/pam.d/system-auth

  session required pam_limits.so

- 编辑 /etc/security/limits.conf

  # 限制 osmond 用户的同时登录数为2

  osmond hard maxlogins 2

  # 限制 students 组中的用户同时登录数为20

  @students hard maxlogins 20

  # 限制系统能够同时登录的最大用户数为50

  *** hard maxsyslogins 50**

梁如军（linuxbooks@126.com）

pam_limits.so举例（2）

限制用户/组在一个会话过程中可使用的系统资源

- cat /etc/pam.d/system-auth

session required pam_limits.so

- 编辑 /etc/security/limits.conf

限制用户 jjheng 最多允许创建 20 个进程

jjheng hard nproc 20

限制 student 组最多允许创建 200 个进程

@student hard nproc 200

限制用户 jjheng 的最多打开 50 个文件

jjheng hard nofile 50

限制用户 jjheng 创建文件的最大限制为 100MB

jjheng hard fsize 102400

梁如军（linuxbooks@126.com）

OPENSSL

梁如军（linuxbooks@126.com）

SSL/TLS

- TLS/SSL 协议层
- TLS 握手协议
- SSL/TLS 协议版本
- 基于TCP的TLS增强应用协议
- OpenSSL

梁如军（linuxbooks@126.com）

SSL/TLS的引入动机

- TCP协议的风险
  - 窃听（eavesdropping）：第三方可以获知通信内容。
  - 篡改（tampering）：第三方可以修改通信内容。
  - 冒充（pretending）：第三方可以冒充他人身份参与通信。
- 许多服务都需要相同的支持
  - 身份验证
  - 数据完整性
  - ……

梁如军（linuxbooks@126.com）

SSL/TLS简介

- SSL = Secure Socket Layer（安全套接字层）
- SSL的特点
  - 提供身份验证的客户端和服务器应用程序
  - 在一个公共信道发送之前对数据进行加密
  - 确保数据完整性
  - 它被设计为有效率
  - 在双方协商使用的主要加密算法

SSL的功能实现

- 对称加密
- 非对称加密
- 数字签名
- 数字证书（X509v.3）
- 明确和正式的规范
- 协商参数
- 在连接的时候握手
- 重用先前谈判的参数

梁如军（linuxbooks@126.com）

广泛使用的SSL

- 电子商务
  - 订单：订购的产品表单使用SSL发送
  - 付款：使用SSL发送信用卡号等数据
- 访问安全信息
  - 信息通信只能由“合格的”用户访问
  - 发送密码或其他敏感数据

梁如军（linuxbooks@126.com）

SSL/TLS 协议的实现版本

- SSL – Secure Sockets Layer Version 2.0
  - Initially developed by Netscape
  - SSL 2.0 is sensitive to man-in-the-middle attacks leading e.g. to the negotiation of weak encryption keys
  - SSL 2.0 should not be used anymore
- SSL – Secure Sockets Layer Version 3.0
  - Internet Draft authored by Netscape, November 1996
  - Supported by all browsers
  - Vulnerable to the BEAST Cipher-Block-Chaining (CBC) attack
- TLS – Transport Layer Security Version 1.0 (SSL 3.1)
  - IETF RFC 2246, January 1999
  - TLS 1.0 ist not backwards compatible to SSL 3.0 (differences in MAC computation, PRF function for master_secret and key material)
  - Supported by all browsers
  - Vulnerable to the BEAST Cipher-Block-Chaining (CBC) attack

梁如军（linuxbooks@126.com）

最近的TLS 协议版本

- TLS – Transport Layer Security Version 1.1 (SSL 3.2)
  - IETF RFC 4346, April 2006
  - Protection against CBC attacks (Serge Vaudenay, EPFL, 2004):
  - Implicit Initialization Vector (IV) is replaced with an explicit IV
  - Handling of padding errors is changed to use the bad_record_mac alert rather then decryption_failed.
- TLS – Transport Layer Security Version 1.2 (SSL 3.3)
  - IETF RFC 5246, August 2008, updated by RFC
  - Combined MD5/SHA-1 hash and PRF functions replaced by SHA-256 based default algorithms or cipher-suite specified methods.
  - Support of Authenticated Encryption with Additional Data (AEAD) modes (e.g. AES-GCM accelerated by Intel AES-NI instruction set)
- TLS 1.1 and 1.2 Support
  - Windows 7, Windows Server 2008 R2
  - GnuTLS library, the OpenSSL 1.0.1 snapshot and strongSwan libtls.

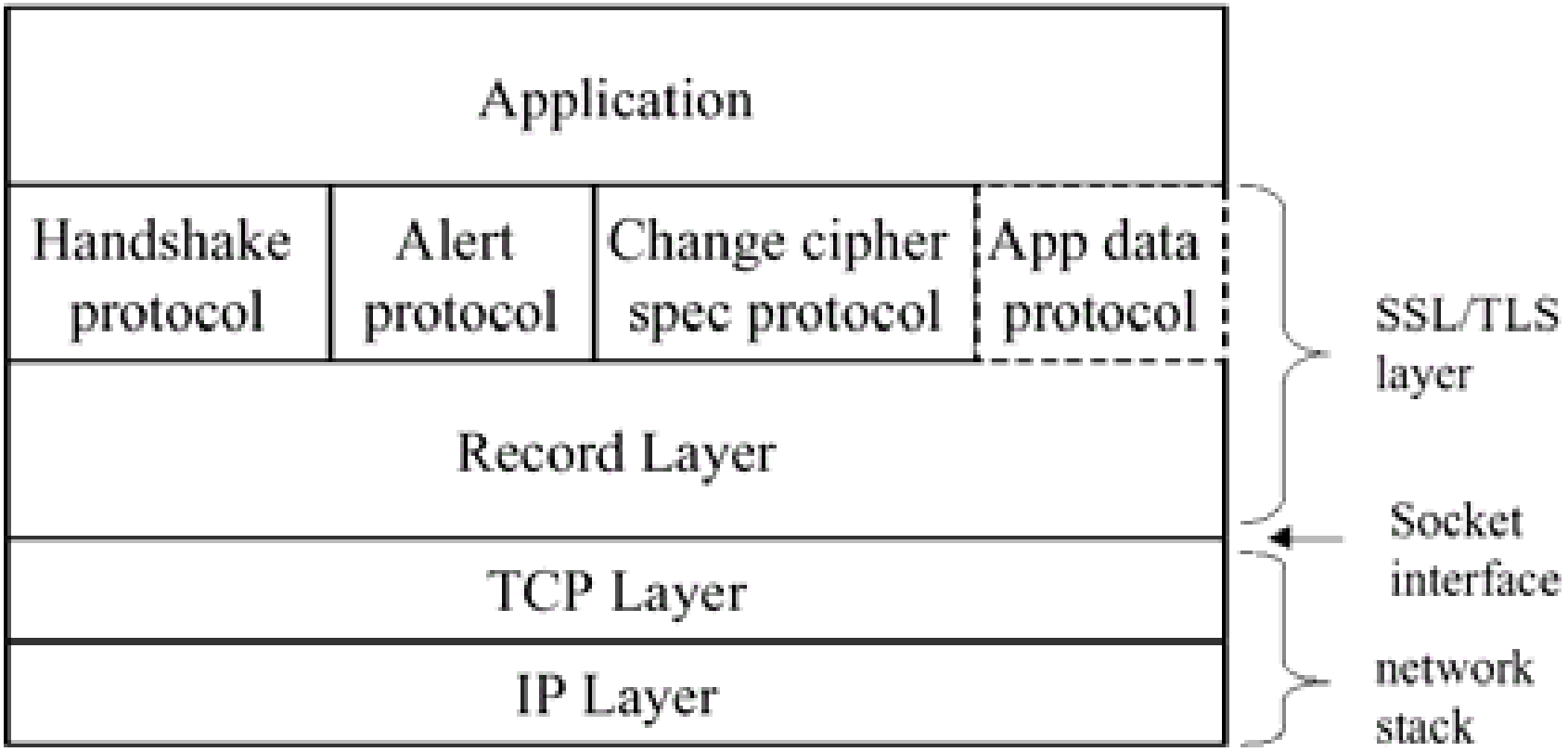
梁如军（linuxbooks@126.com）

TLS增强的基于TCP的应用协议

服务名	端口号	实现的安全服务
https	443/tcp	http protocol over TLS
smtps	465/tcp	smtp protocol over TLS
smtp	25/tcp	STARTTLS keyword (RFC 2487)
imaps	993/tcp	imap4 protocol over TLS
imap4	143/tcp	STARTTLS keyword (RFC 2595)
pop3s	995/tcp	pop3 protocol over TLS
pop3	110/tcp	STLS keyword (RFC 2595)
ldaps	636/tcp	ldap protocol over TLS
nntps	563/tcp	nntp protocol over TLS
FTPS-Data	989/tcp	FTP Data over SSL/TLS
FTPS	990/tcp	FTP Control over SSL/TLS

梁如军（linuxbooks@126.com）

SSL/TLS 与TCP/IP



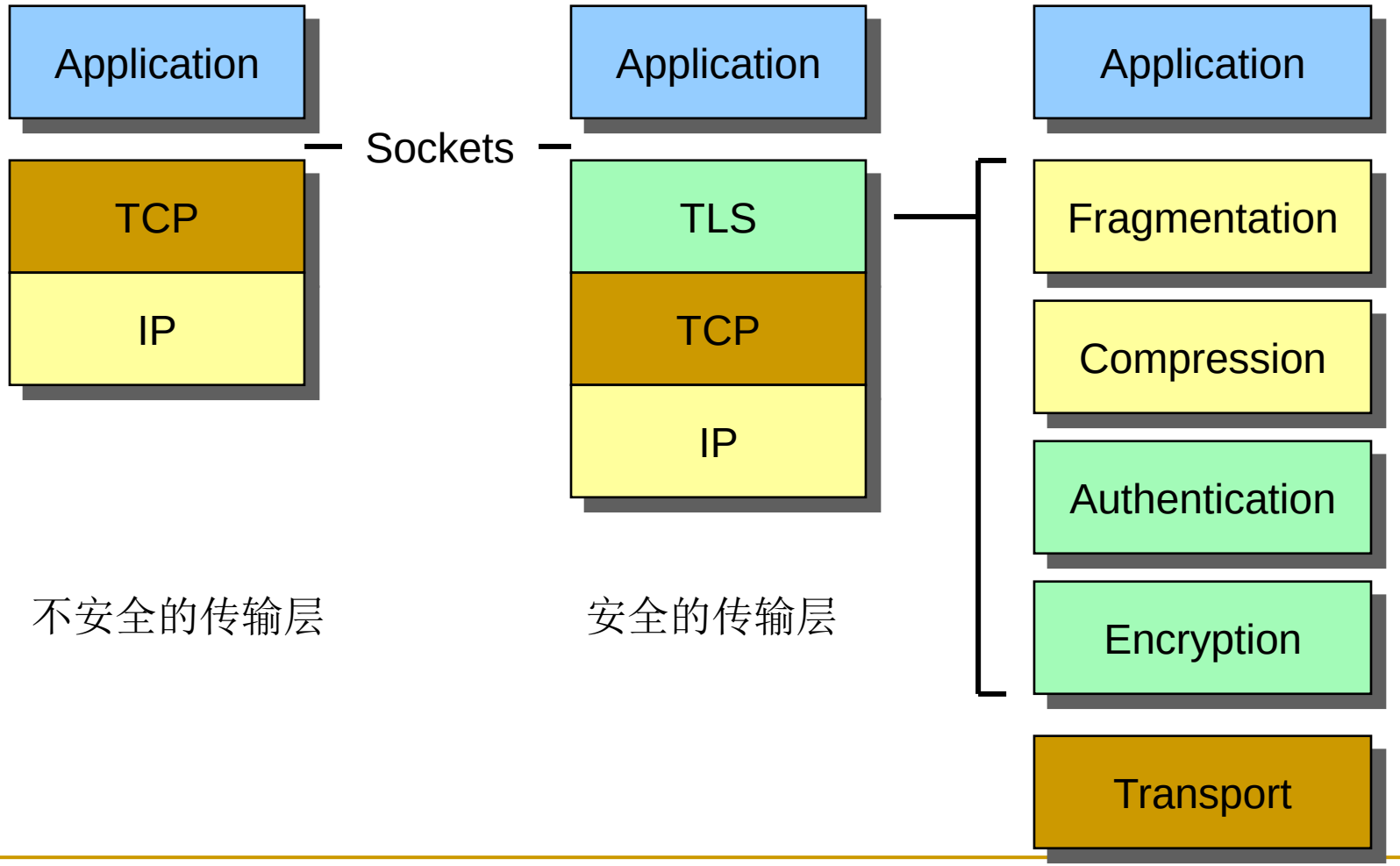
梁如军（linuxbooks@126.com）

SSL协议组件

- 握手协议（Handshake protocol）
  - 允许当事人协商需要交易的安全性的不同算法
  - 允许当事人之间的任何身份验证
- 警报协议（Alert protocol）
  - 通知异常情况或报告问题
- 更改密码说明协议（Change Cipher Spec protocol）
  - 强制一个新的握手的执行重新协商安全参数，并重复认证
- 记录协议（Record protocol）
  - 涉及的压缩，加密和MAC

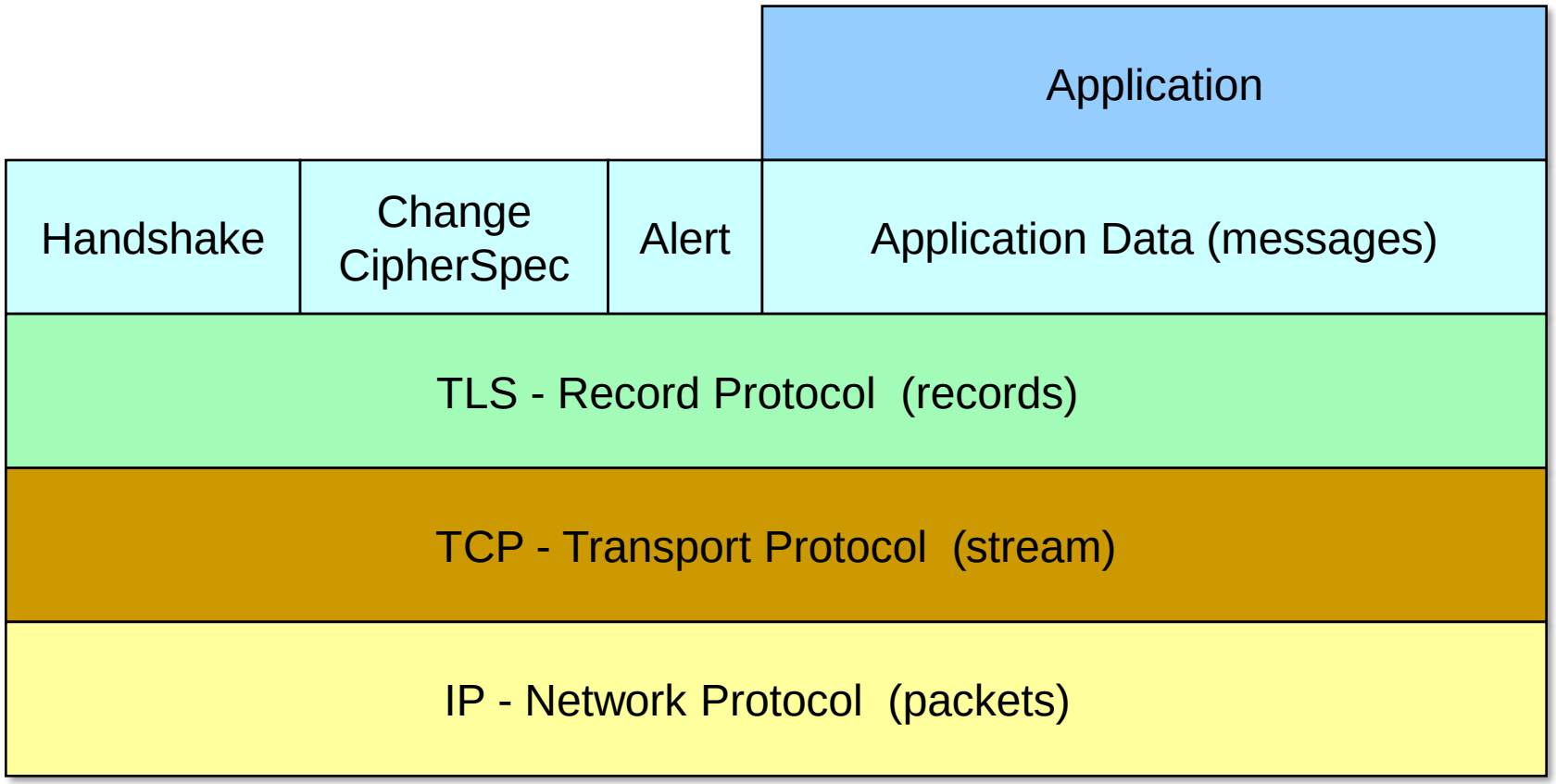
梁如军（linuxbooks@126.com）

TLS/SSL 协议层



梁如军（linuxbooks@126.com）

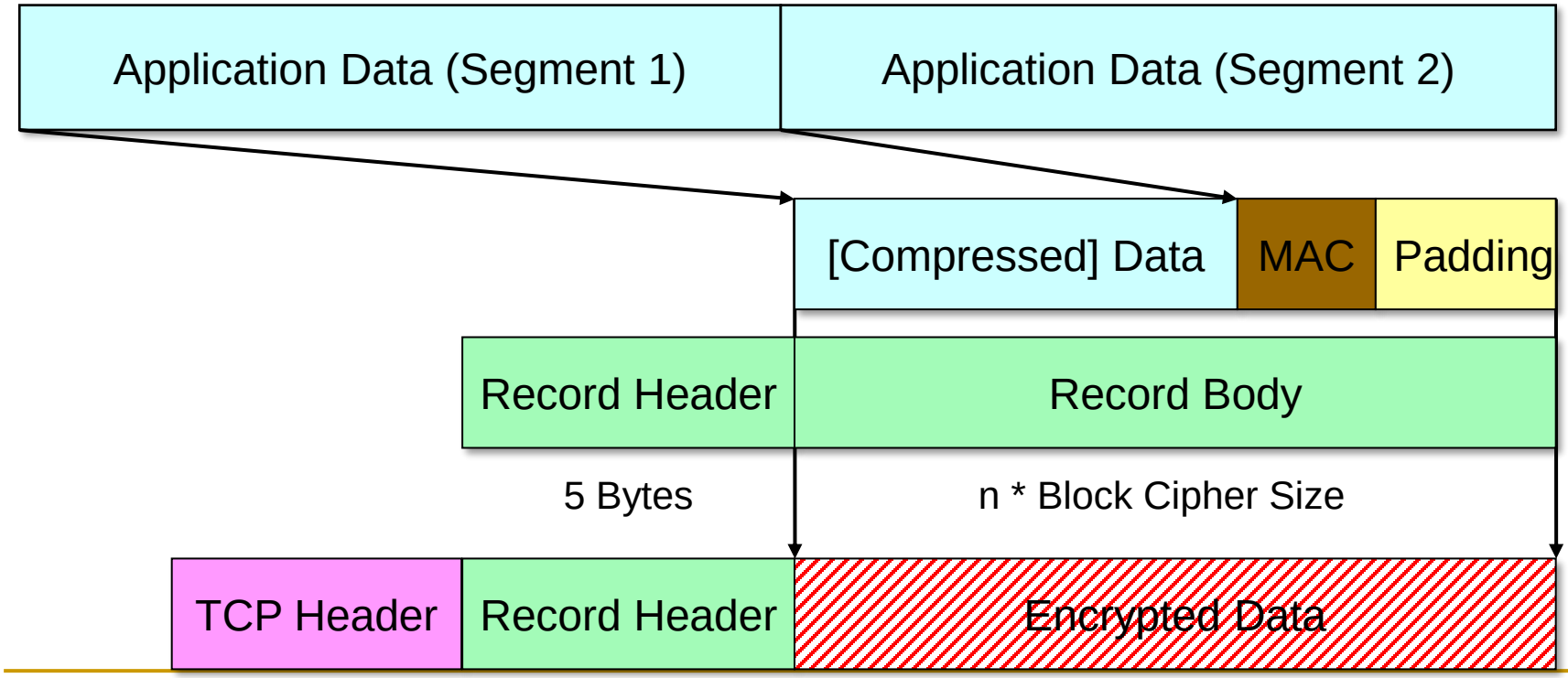
TLS Record Protocol



梁如军（linuxbooks@126.com）

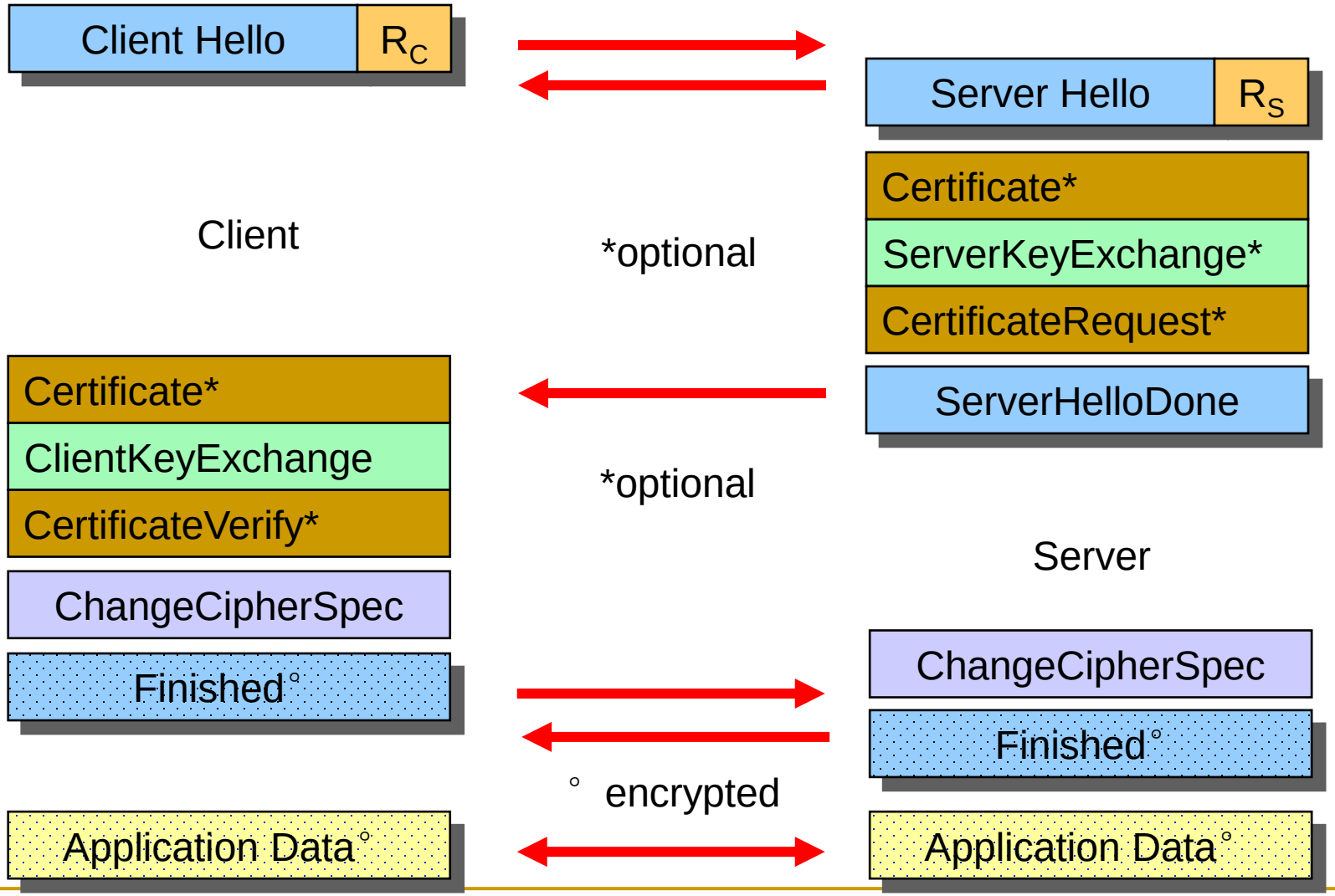
TLS Record Structure

```
⊞ Frame 9 (603 bytes on wire, 603 bytes captured)
⊞ Ethernet II, Src: Dell_92:44:9f (00:1a:a0:92:44:9f), Dst: SitecomE_5a:74:5e (00:0c:f6:5a:74:5e)
⊞ Internet Protocol, Src: 192.168.1.198 (192.168.1.198), Dst: hsr.ch (152.96.37.60)
⊞ Transmission Control Protocol, Src Port: 49825 (49825), Dst Port: https (443), Seq: 996, Ack: 1519, Len: 549
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 544
      Encrypted Application Data: C72EA7E35EE5501648FB77E216FCEA6CF62899BBD1ADDDAD...
```



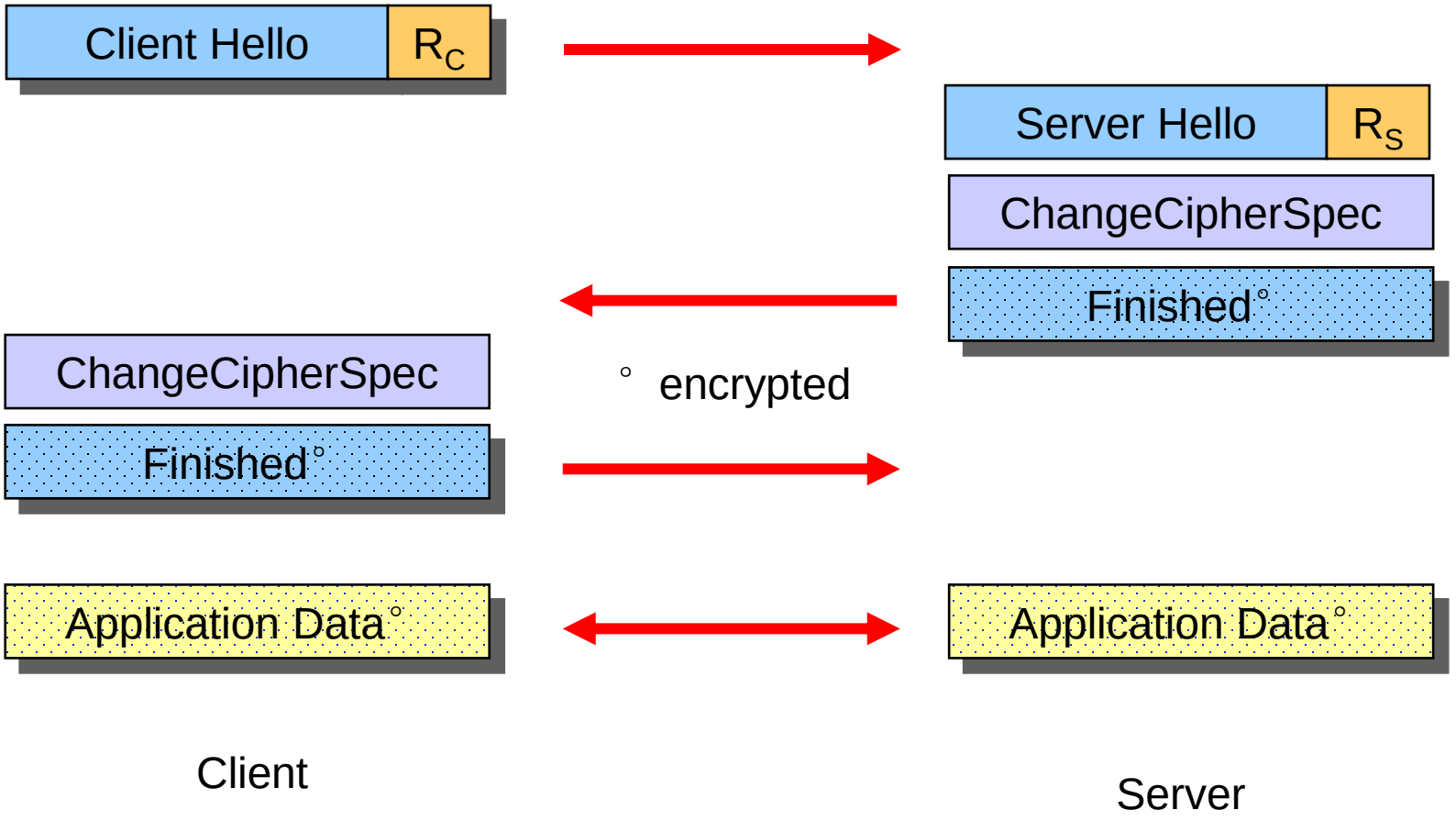
梁如军（linuxbooks@126.com）

TLS Handshake Protocol



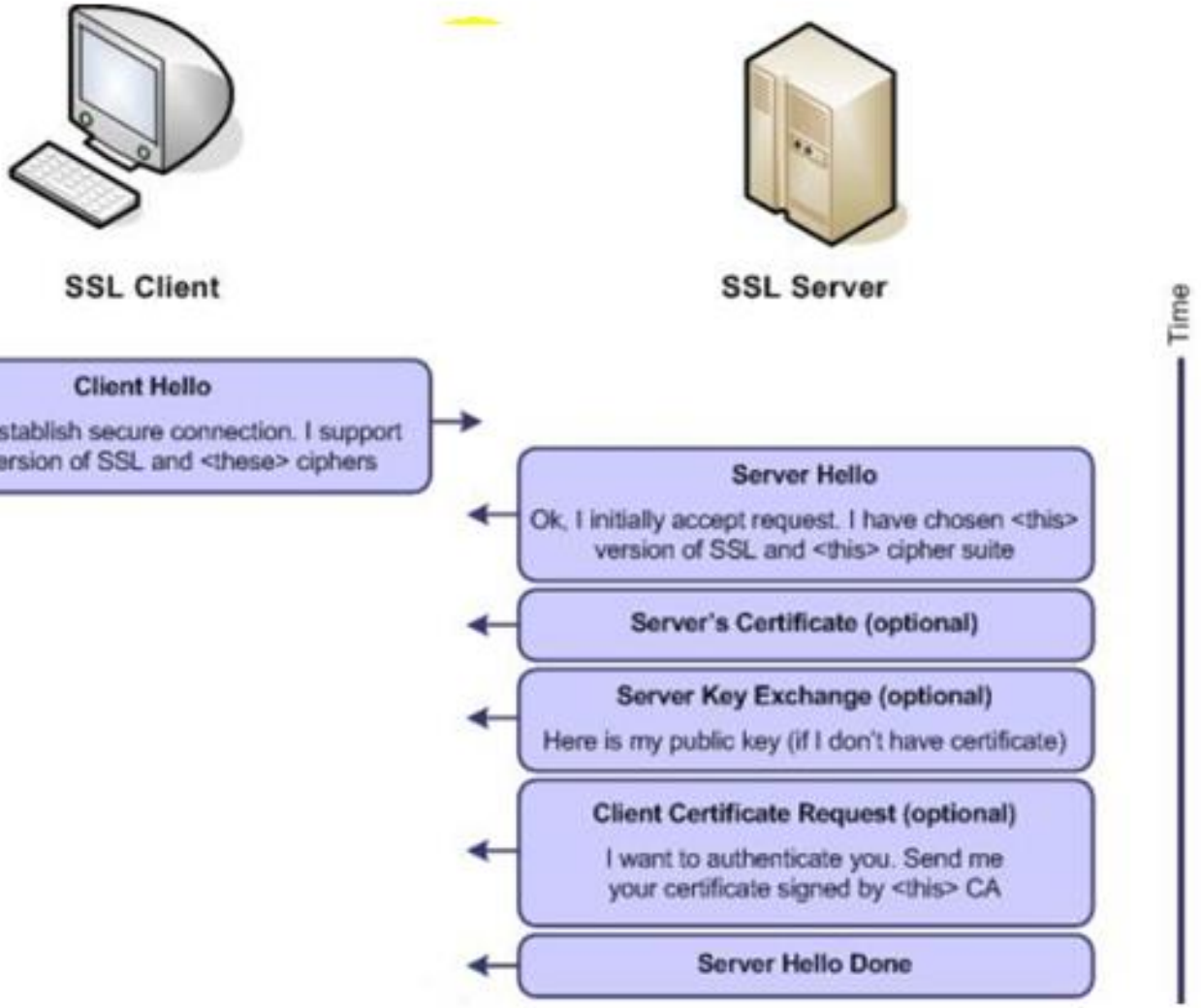
梁如军（linuxbooks@126.com）

Resuming a TLS Session



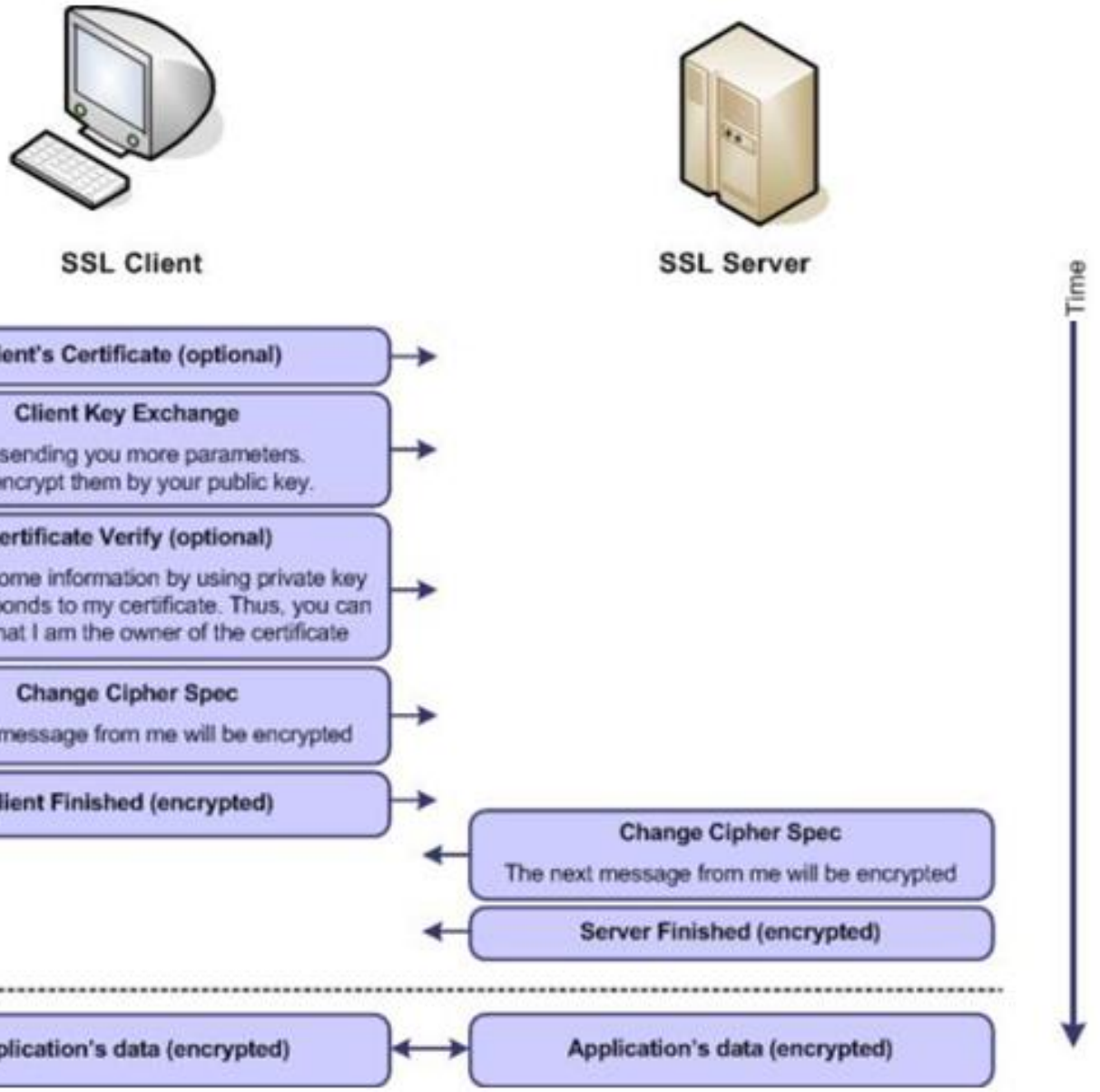
梁如军（linuxbooks@126.com）

SSL如何建立连接（1）



梁如军（linuxbooks@126.com）

SSL如何建立连接（2）


SSL Client
SSL Server
Time
Client's Certificate (optional)
Client Key Exchange
I am sending you more parameters.
I will encrypt them by your public key.
Certificate Verify (optional)
I will sign some information by using private key that corresponds to my certificate. Thus, you can be sure that I am the owner of the certificate
Change Cipher Spec
The next message from me will be encrypted
Client Finished (encrypted)
Change Cipher Spec
The next message from me will be encrypted
Server Finished (encrypted)
Application's data (encrypted)
Application's data (encrypted)

OpenSSL

- 什么是 OpenSSL
- OpenSSL 的特性
- 命令行界面

- 使用 OpenSSL构建安全的 FTP

梁如军（linuxbooks@126.com）

什么是 OpenSSL

- OpenSSL项目是一个协作开发的，健壮的，商业级的，功能齐全的，实现 SSL_v2/v3和TLS_v1协议的开源工具包，包括一个全强度的通用加密库。
- OpenSSL 基于由 Eric A. Young 和 Tim J. Hudson. 开发的优秀的 SSLeay 库
- 当前版本 on CentOS6/Debian7
  - OpenSSL 1.0.1e ----- 11 Feb 2013

OpenSSL 特性

- 开源，基于一个Apache风格的许可证发布
- 提供了SSL v2/v3和TLS v1.0的全功能实现
- 用C语言开发，具有优秀的跨平台性能
- 基于PKI标准，支持 X509 证书标准
- 提供众多的加密和摘要算法库
- 提供了命令行界面（openssl命令）
- 提供了应用程序编程接口

梁如军（linuxbooks@126.com）

OpenSSL应用程序接口

- libssl.a or libssl.so
  - Implementation of SSL_v2/3 & TLS_v1
- libcrypto.a or libcrypto.so
  - Ciphers (AES, DES, RC2/4, Blowfish, IDEA)
  - Digests (MD5, SHA-1, MDC2)
  - Public Keys (RSA, DSA, DH)
  - X509s (ASN.1 DER & PEM)
  - Others (BIO, BASE64)

梁如军（linuxbooks@126.com）

OpenSSL应用程序接口（续）

- OpenSSL’s libraries are also used by other tools, such as OpenCA, OpenSSH, to implement secure transmission of data
- Using SSL Proxy, arbitrary socket connections can be secured by SSL

命令行接口

- 功能
  - 创建 RSA, DSA & DH 密钥对
  - 公共密钥加密操作
  - 创建 X509 证书, CSRs & CRLs
  - 生成消息摘要
  - 使用加密算法加密&解密
  - SSL/TLS 服务器端/客户端测试
  - 处理 S/MIME 签名或加密邮件
  - 时间戳记的请求，生成和验证
  - 创建和管理CA

梁如军（linuxbooks@126.com）

openssl的命令和算法

- $ openssl list-standard-commands
- $ openssl list-cipher-commands
- $ openssl list-message-digest-commands

- $ openssl list-cipher-algorithms
- $ openssl list-public-key-algorithms

梁如军（linuxbooks@126.com）

名词解释（1）

- 密码学(cryptography)：目的是通过将信息编码使其不可读，从而达到安全性。
- 明文(plain text)：发送人、接受人和任何访问消息的人都能理解的消息。
- 加密(encryption)：将明文消息变成密文消息。
- 密文(cipher text)：明文消息经过某种编码后，得到密文消息。
- 解密(decryption)：将密文消息变成明文消息。

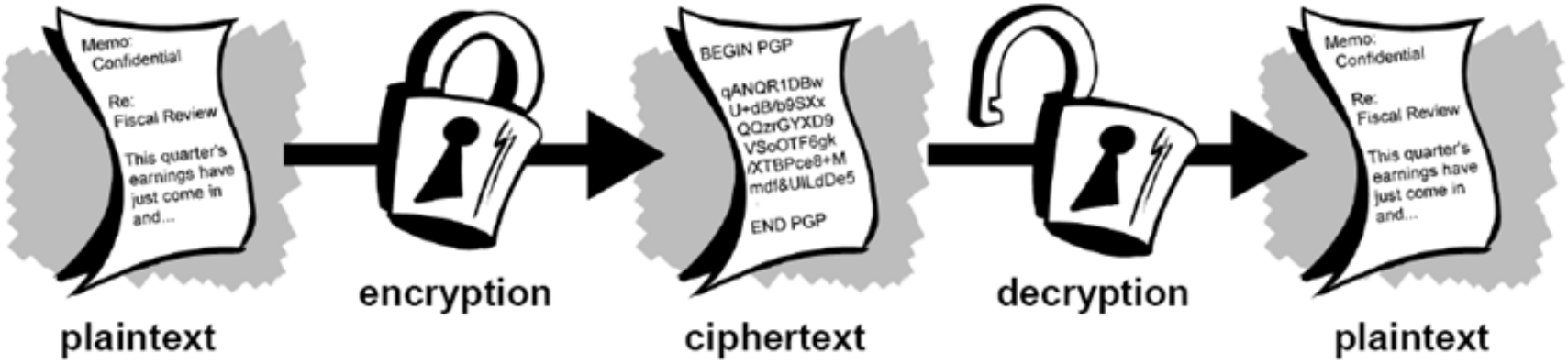
梁如军（linuxbooks@126.com）

名词解释（2）

- 算法：取一个输入文本，产生一个输出文本。
  - 加密算法：发送方进行加密的算法。
  - 解密算法：接收方进行解密的算法。
- 密钥(key)：只有发送方和接收方理解的消息
  - 对称密钥加密(Symmetric Key Cryptography)：
    - 加密与解密使用相同密钥。
  - 非对称密钥加密(Asymmetric Key Cryptography)：
    - 加密与解密使用不同密钥。

梁如军（linuxbooks@126.com）

加密和解密

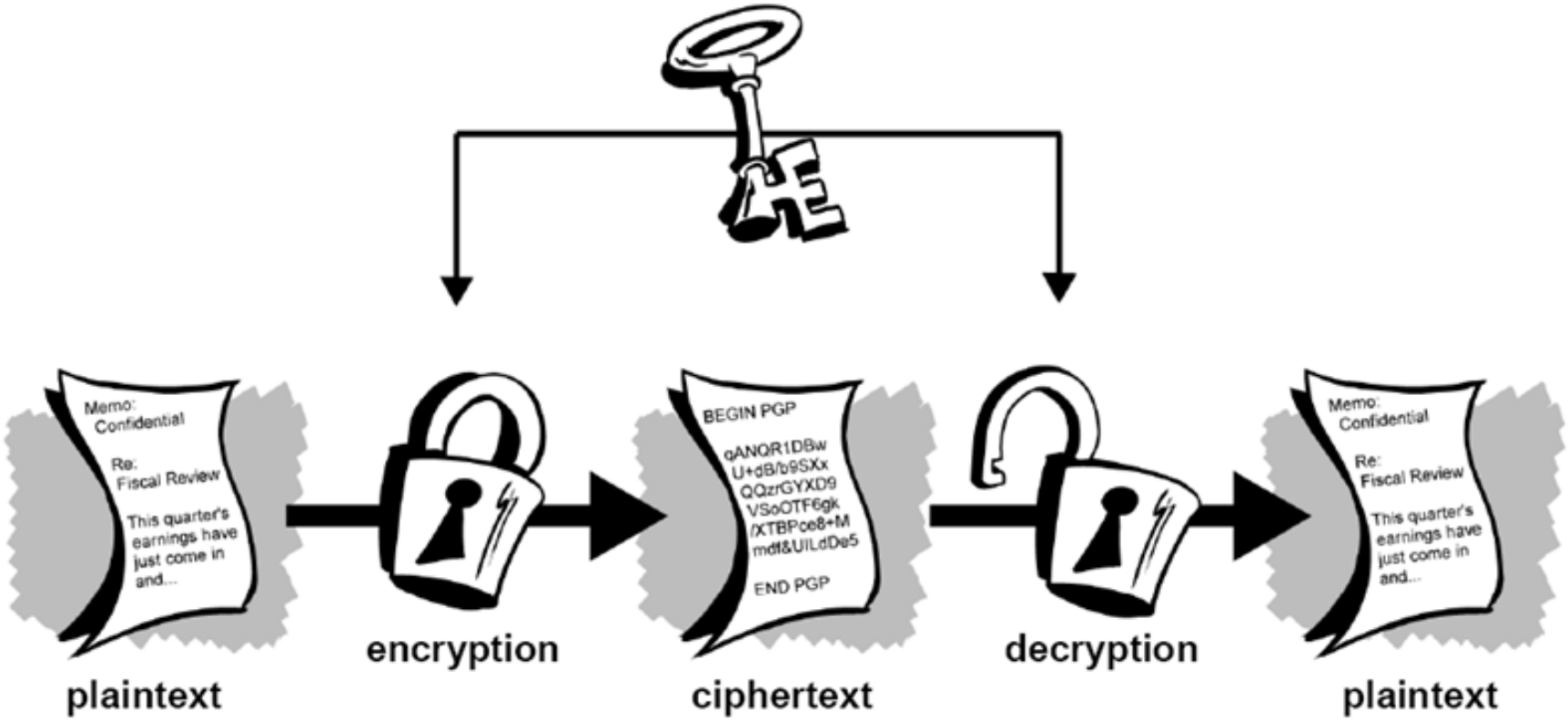


$$CT = f_1(PT, key)$$

$$PT = f_2(CT, key)$$

梁如军（linuxbooks@126.com）

对称加密和解密

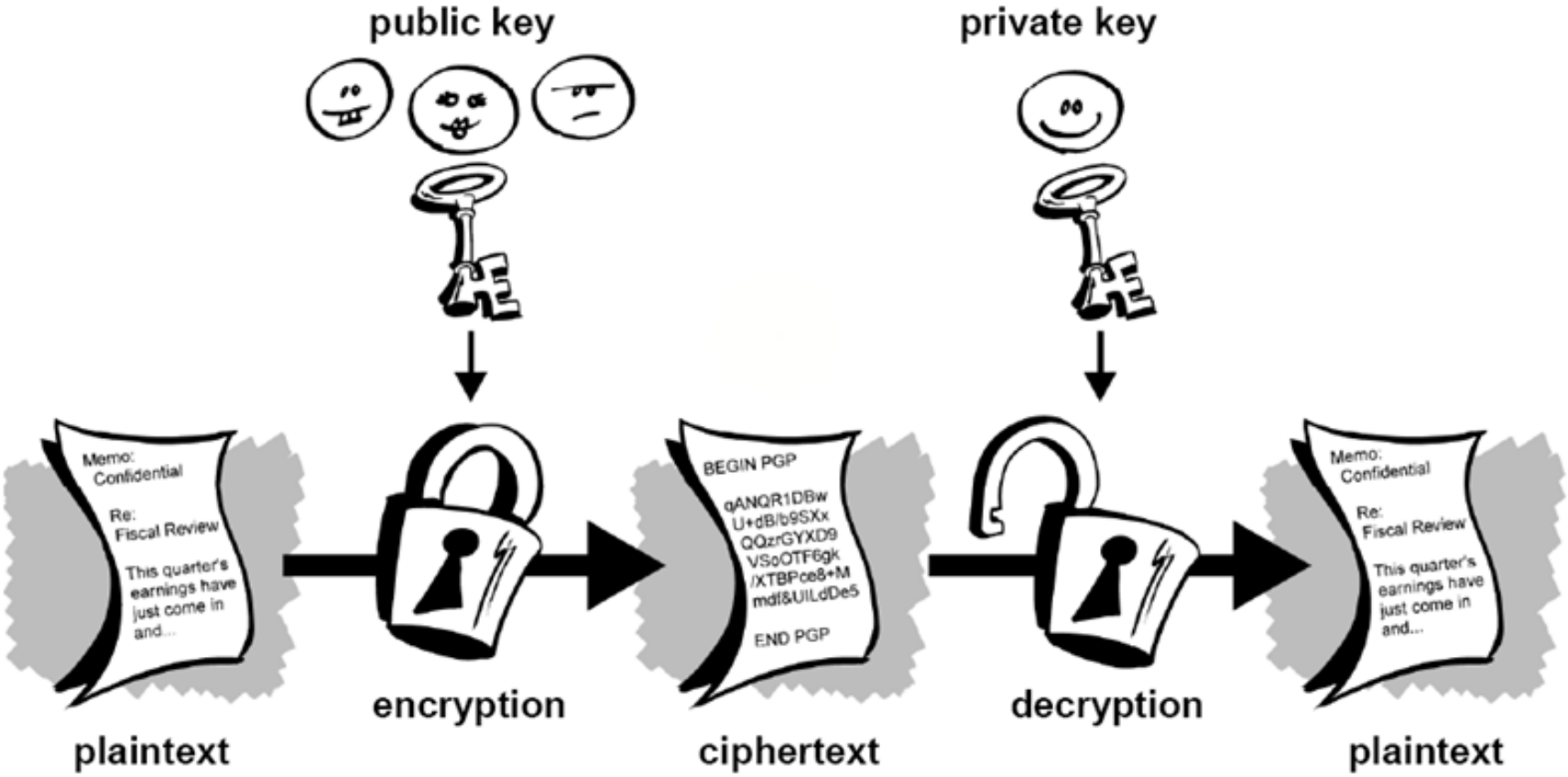


梁如军（linuxbooks@126.com）

加密/解密（对称）

- 加密
  - $ openssl enc –ciphername –k password -e \
    -in inputfile -out outputfile
- 解密
  - $ openssl enc –ciphername –k password -d \
    -in inputfile -out outputfile
- Ciphers name
  - aes256, aes128,des, des3, rc2, rc5, ……

非对称加密和解密

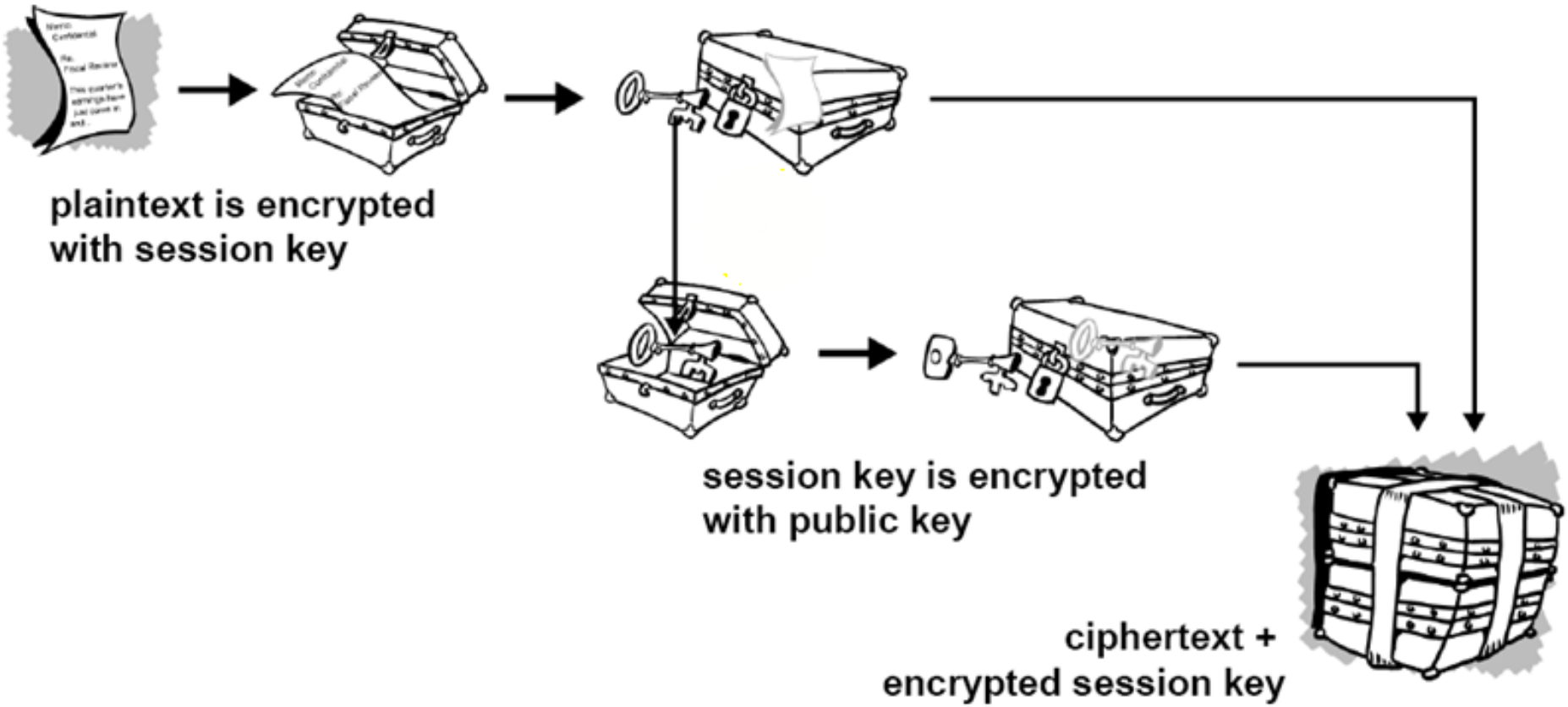


梁如军（linuxbooks@126.com）

加密/解密（非对称）

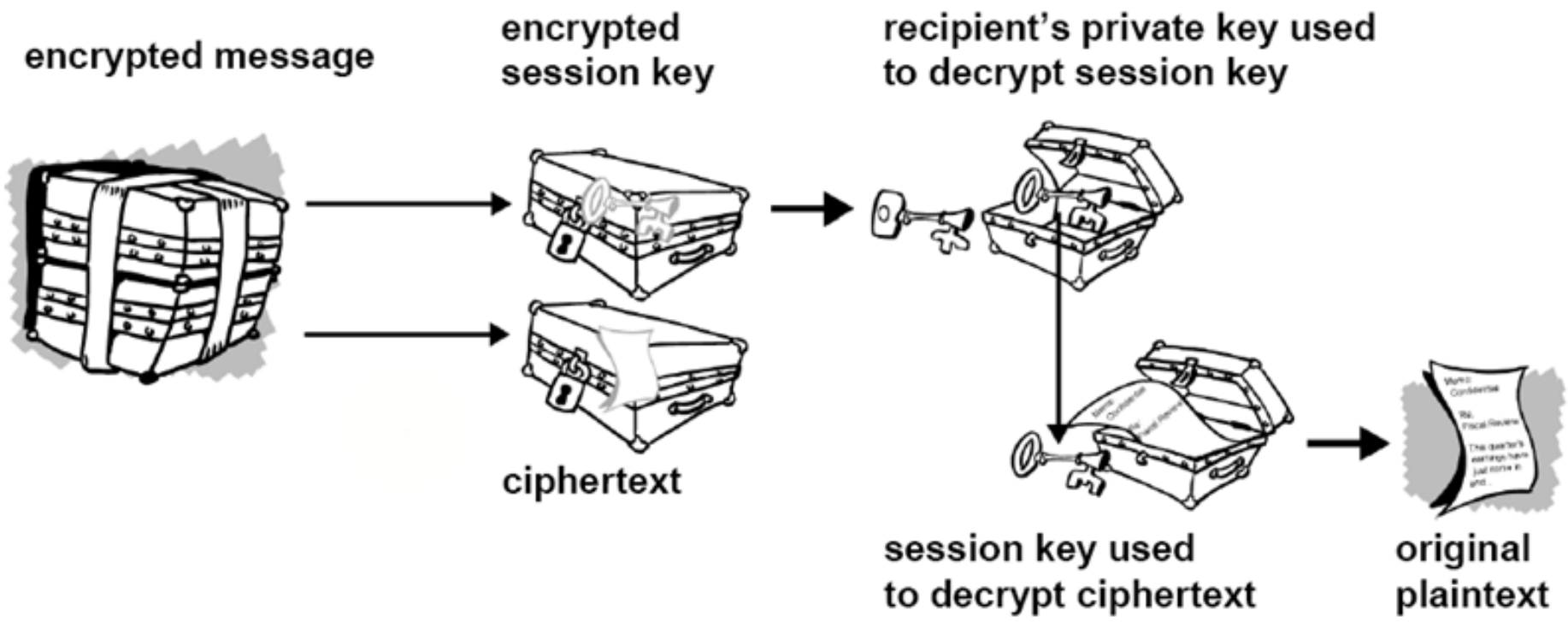
- 生成密钥对
  - $ openssl genrsa -out priv.keyfile 2048
  - $ openssl rsa -in priv.keyfile -pubout > pub.keyfile
- 用公钥加密
  - $ openssl rsautl -in inputfile -out outputfile \
    -pubin -inkey pub.keyfile -encrypt
- 用私钥解密
  - $ openssl rsautl -in inputfile -out outputfile \
    -inkey priv.keyfile -decrypt

混合使用两种加密体系



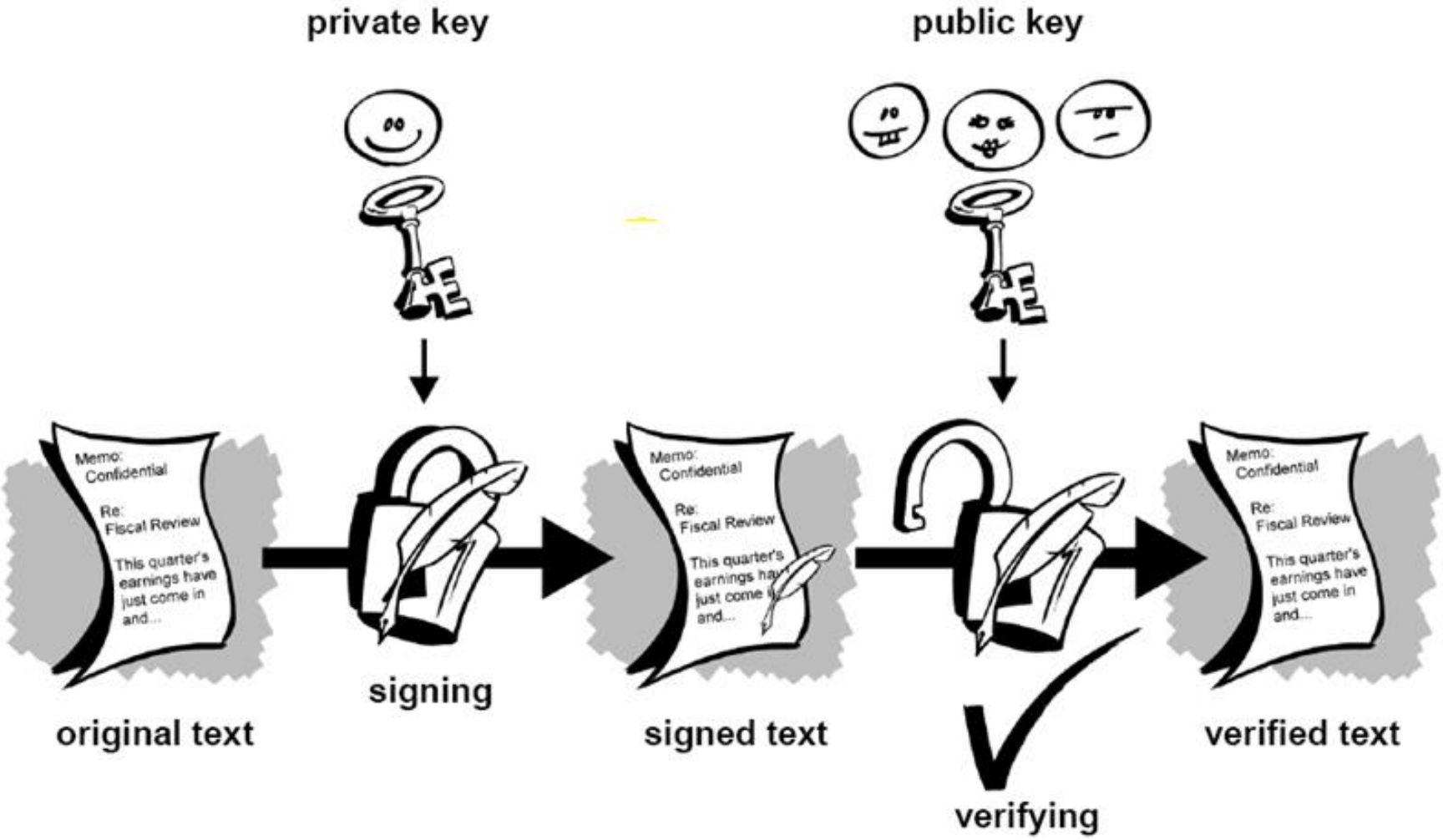
梁如军（linuxbooks@126.com）

混合使用两种加密体系（续）



梁如军（linuxbooks@126.com）

简单的数字签名

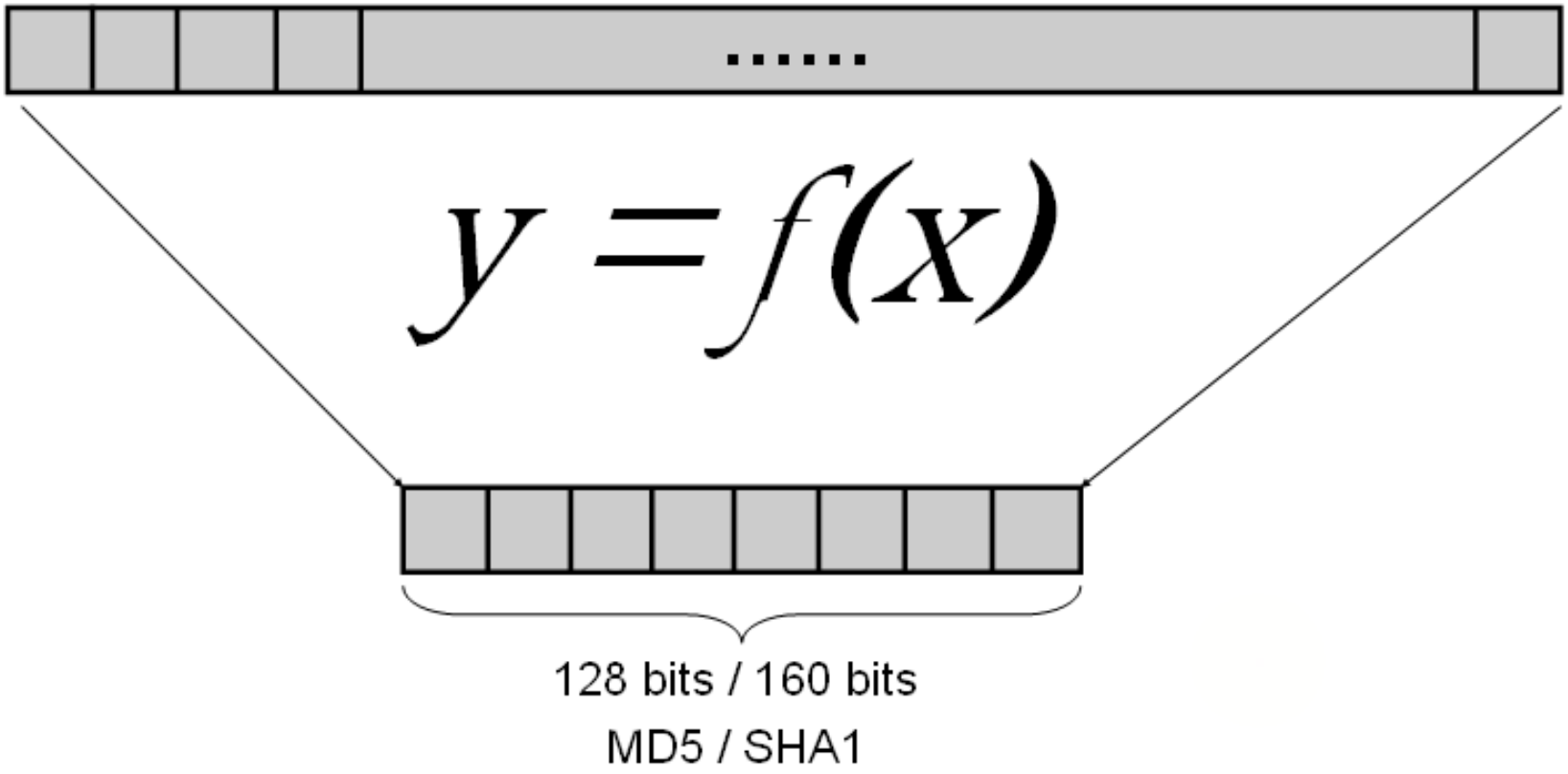


梁如军（linuxbooks@126.com）

签名/验证（非对称）

- 生成密钥对
  - $ openssl genrsa -out priv.keyfile 2048
  - $ openssl rsa -in priv.keyfile -pubout > pub.keyfile
- 用私钥签名
  - $ openssl rsautl -in inputfile -out outputfile \
    -inkey priv.keyfile -sign
- 用公钥验证
  - $ openssl rsautl -in inputfile -out outputfile \
    -pubin -inkey pub.keyfile -verify

哈希（hash）函数



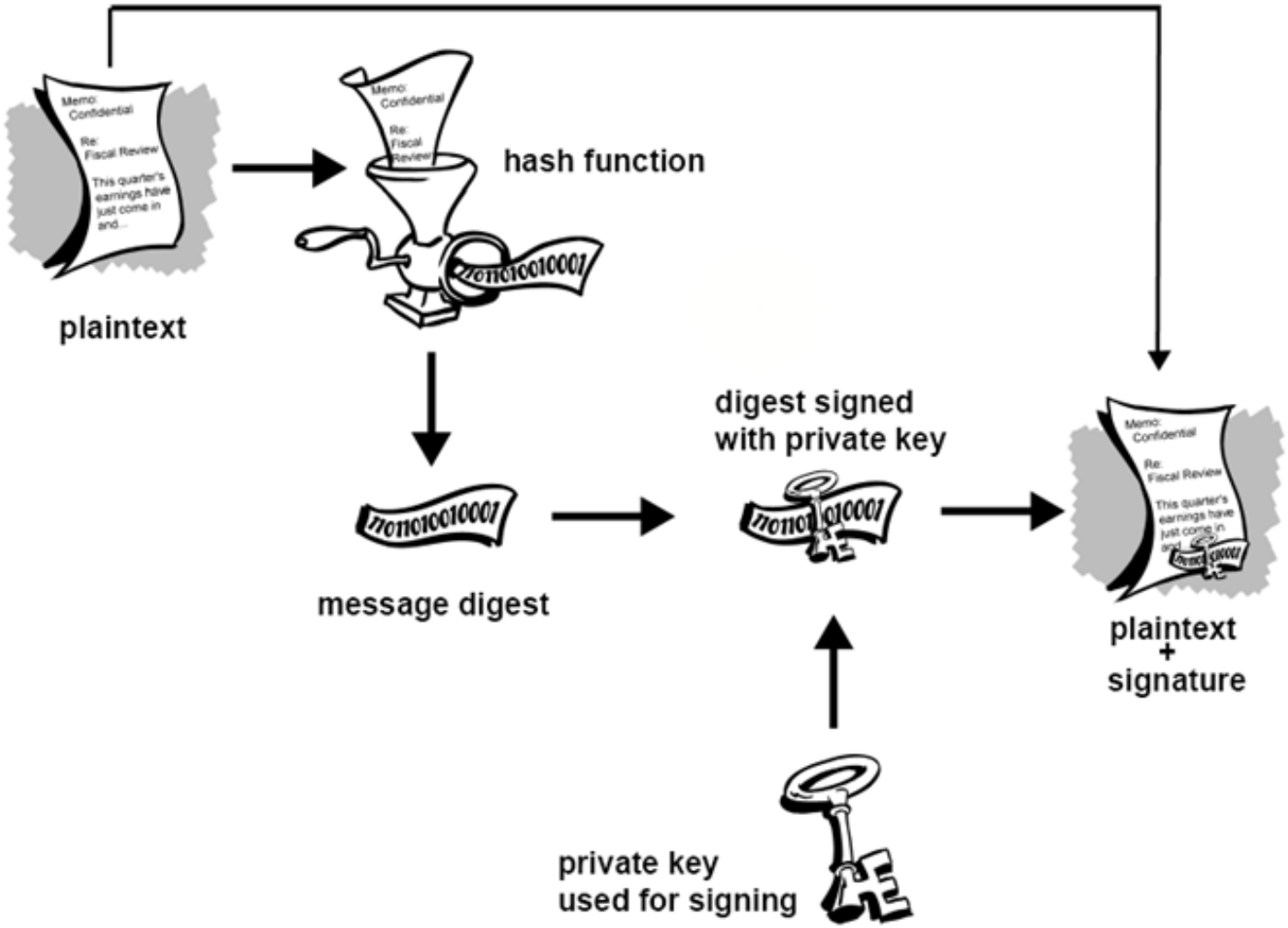
梁如军（linuxbooks@126.com）

Hash/Digest 函数

- 生成 MD5 Hash
  - $ openssl dgst -md5 file
  - $ md5sum file
- 生成 SHA1 Hash
  - $ openssl dgst -sha1 file
  - $ sha1sum file
- 生成 SHA256 Hash
  - $ openssl dgst -sha256 file
  - $ sha256sum file

梁如军（linuxbooks@126.com）

安全的数字签名



梁如军（linuxbooks@126.com）

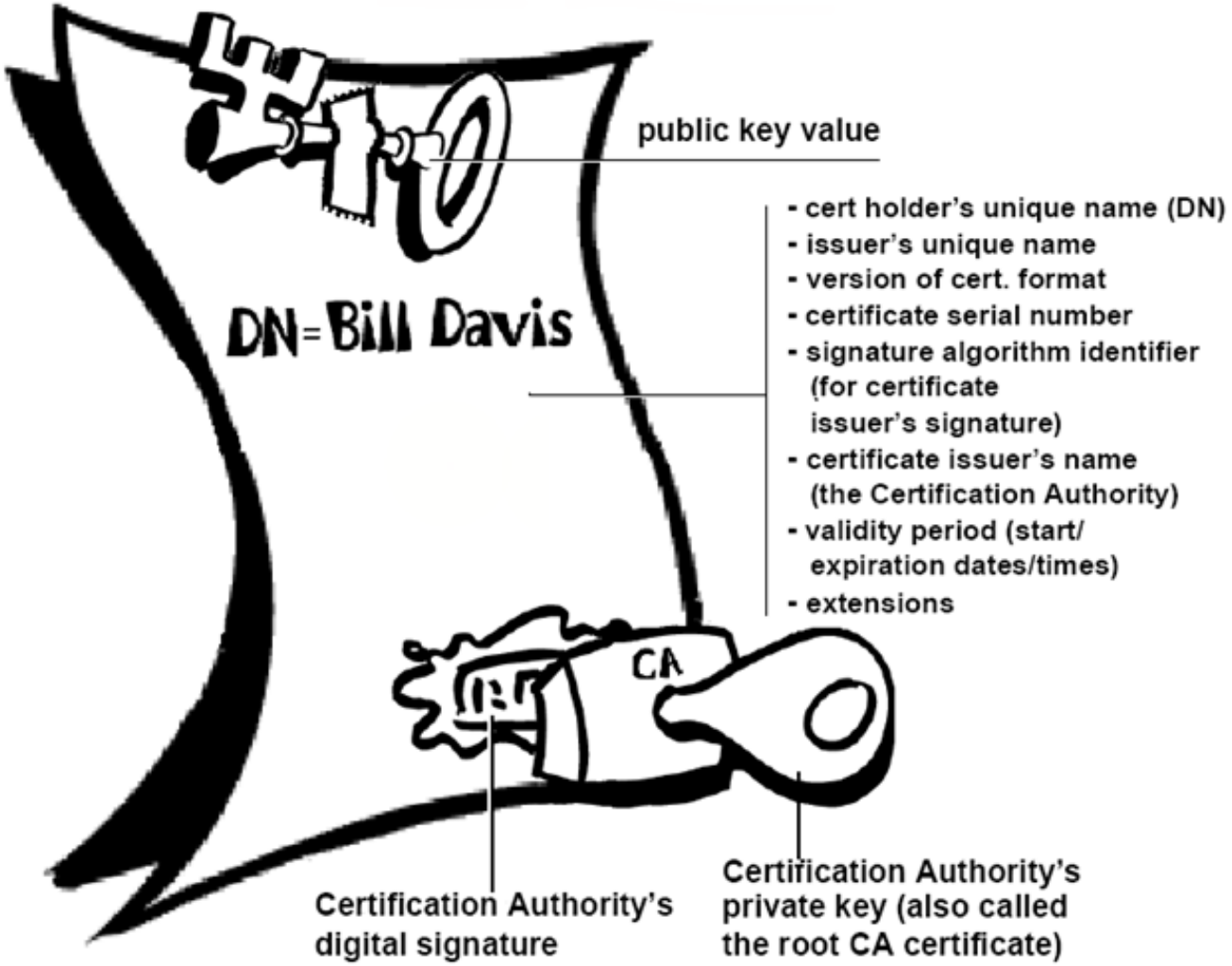
名词解释（3）

- PKI
  - 公钥基础设施（Public Key Infrastructure）
  - 是一个基于非对称加密技术实现并提供安全服务的具有通用性的安全基础设施。
  - 通过一组组件和规程，支持利用数字证书管理密钥并建立信任关系。
  - 同时融合了Hash算法以及对称加密技术。
  - 支持PKI的应用标准可以保护信息的完整性、保密性和不可否认性等安全特征。

名词解释（4）

- 证书（数字证书）
  - 将证书持有者的身份信息和其所拥有的公钥进行绑定的文件。
  - 证书文件还包含颁发证书的权威机构（CA）对该证书的签名。通过签名保障了证书的合法性和有效性。
  - 证书（和相关的私钥）可以提供诸如身份认证、完整性、机密性和不可否认性等安全保护。
  - 当前，通常使用的证书是 X.509 v3 标准的

数字证书（Digital Certificate）



梁如军（linuxbooks@126.com）

数字证书的组成

- 服务器公钥
- 支持的加密算法
- DN（Distinguish Name）：
  - CN（Common Name）：通常是服务器的FQDN
  - 其他的可选属性：Country（C）、State（S）、Location（L）……
- 证书的有效期（起始日期，截止日期）
- 证书的序列号（serial number）
- 被信任的 CA的名字和签名
- X.509 的其他扩展属性等

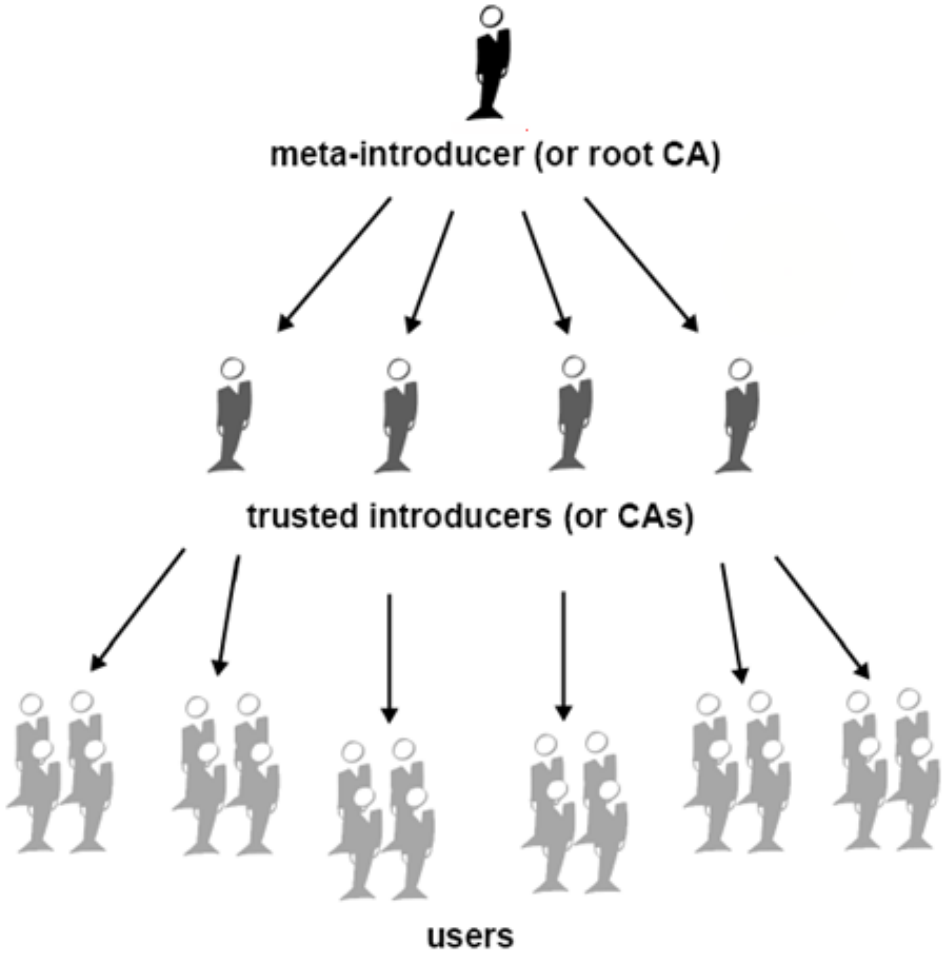
梁如军（linuxbooks@126.com）

名词解释（5）

- CA（Certificate Authority，证书权威机构）
  - CA是PKI中受信任的第三方实体。
    - CA是信任的起点，各个实体必须对CA高度信任，因为他们要通过CA的担保来认证其他实体。
  - CA是PKI的核心，任务包括
    - 证书管理：证书颁发、吊销、更新和续订等
    - CRL 和/或 OCSP发布
    - 证书存储以及事件日志记录等
  - CA的主要任务是颁发证书。当一个主体向CA申请证书时，CA在对其进行必须的验证之后，为其颁发证书。

梁如军（linuxbooks@126.com）

CA的层次信任关系



梁如军（linuxbooks@126.com）

OpenSSL支持的证书和密钥——格式标准

- **DER**（Distinguished Encoding Rules）
  - 是一种使用 DER ASN.1 编码的二进制数据版本
- **PEM**（Privacy-Enhanced Mail）
  - 是一种 base64 编码的 DER 格式数据版本，并添加了头部起始行和尾部结束行以便通过邮件传输
- **PKCS#X**（Public Key Cryptography Standards，公钥加密标准）
  - 是 RSA安全实验室开发的一组证书管理标准，用于定义安全信息交换的方法。

梁如军（linuxbooks@126.com）

OpenSSL支持的证书和密钥——常见的文件后缀

后缀	说明
.key	PEM 格式的 RSA 或 DSA 私钥文件
.csr	PEM 格式的 证书签名请求文件
.crt	PEM 格式的 X.509 证书文件
.pem	PEM 格式的文件（通常用于将证书和密钥保存在一个文件的情况）
.der	DER 格式的证书文件
.p7b .p7c	PKCS#7 格式的证书文件
.pfx .p12	PKCS#12 格式的密钥及证书文件

梁如军（linuxbooks@126.com）

OpenSSL支持的证书和密钥——格式转换

转换	命令
PEM Certificate to DER Certificate	openssl x509 -outform der -in www.example.com.crt -out www.example.com.der
DER Certificate to PEM Certificate	openssl x509 -inform der -in www.example.com.der -out www.example.com.crt
PEM RSA Key to DER RSA Key	openssl rsa -in www.example.com.key -outform DER -out www.example.com.der.key
DER RSA Key to PEM RSA Key	openssl rsa -inform der -in www.example.com.key -out www.example.com.pem.key
PEM Certificate to PKCS#7 Certificate	openssl crl2pkcs7 -nocrl -certfile www.example.com.crt -out www.example.com.p7b
PCKS#7 Certificate to PEM Certificate	openssl pkcs7 -print_certs -in www.example.com.p7b -out www.example.com.crt
PEM Certificate and Key to PKCS#12 Certificate and Key	openssl pkcs12 -export -out www.example.com.pfx -inkey www.example.com.key -in ww.example.com.crt
PKCS#12 Certificate and Key to PEM Certificate and Key	openssl pkcs12 -in www.example.com.pfx -out www.example.com.pem -nodes

梁如军（linuxbooks@126.com）

OpenSSL的密钥和证书管理

- 密钥和证书
  - 使用**openssl**命令
  - 使用crypto-utils软件包提供的 TUI工具 **genkey**
  - 在/etc/pki/tls/certs/目录下执行**make**
- CA管理
  - 使用**openssl**命令
  - 使用基于openssl的前端GUI工具 **xca**（http://xca.sf.net）
  - 安装部署 **OpenCA**（https://pki.openca.org/ ）
  - 使用操作更简单的 openssl 包裹脚本
    - 使用EPEL仓库提供的easy-rsa（用于openvpn和web的密钥管理）
    - 使用 http://www.openssl.org/contrib/ssl.ca-0.1.tar.gz
    - 使用 https://github.com/didier13150/manageCA

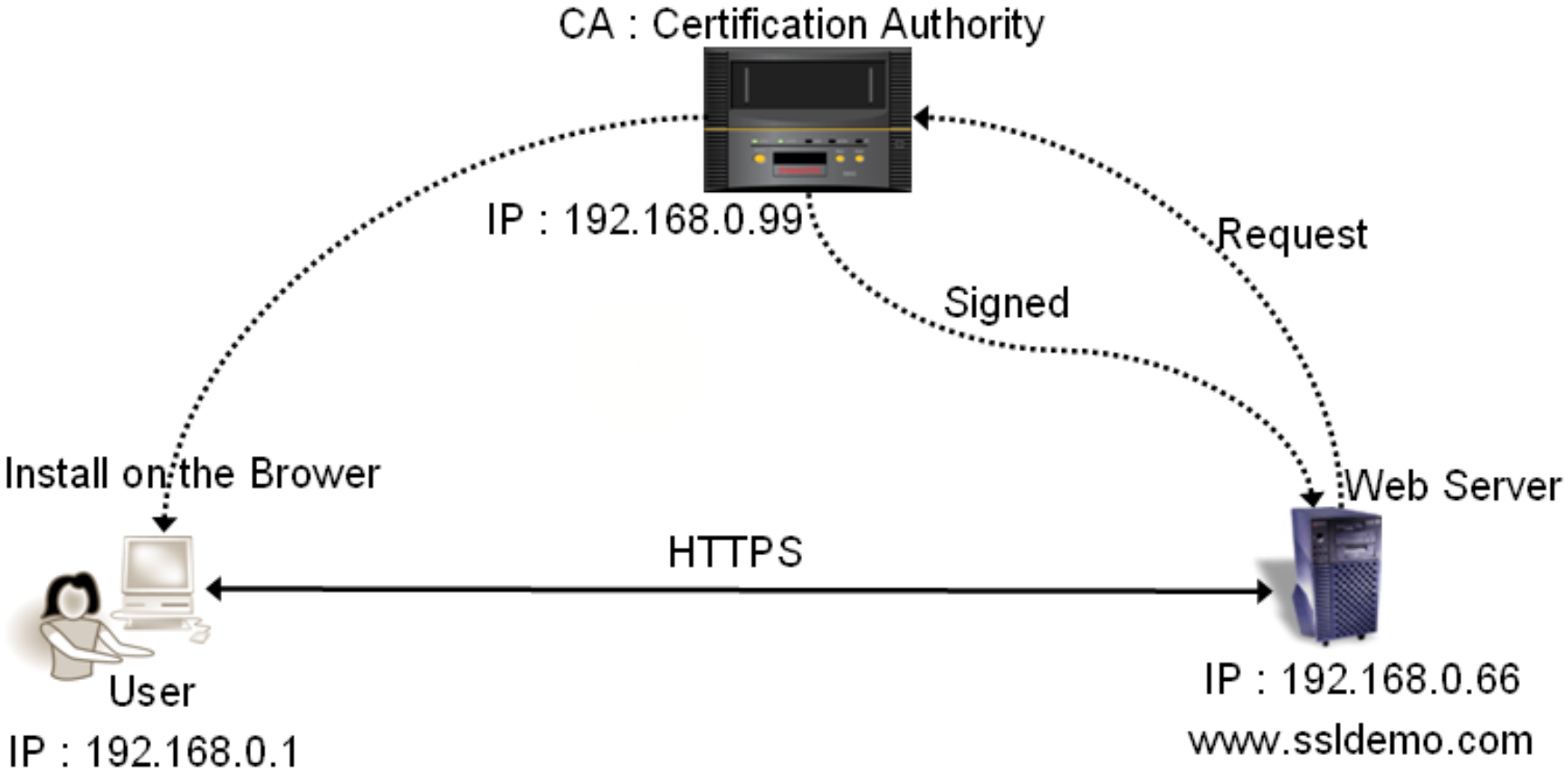
梁如军（linuxbooks@126.com）

生成服务器的数字证书

- 三种类型的证书
  - 自签名证书（通常用于实验环境）
  - 由本地CA签署的证书（用于 Intranet 环境）
  - 由可信任的CA签署的证书（用于 Internet 环境）
- 最终获得两个文件
  - server.key——服务器的私钥
  - server.crt——包含服务器公钥的PEM格式证书

梁如军（linuxbooks@126.com）

由CA签署证书的过程



梁如军（linuxbooks@126.com）

本地CA签署的证书

- 创建本地 root CA
  - 创建符合openssl标准的CA目录结构
  - 创建或编辑CA使用的配置文件 openssl.cnf
  - 创建CA的自签名证书及其私钥
- 本地CA的安全
  - 使用单独的服务器创建CA
  - 确保创建CA服务器的物理安全
  - 仅允许授权用户访问CA服务器
  - 保护好CA的私钥

梁如军（linuxbooks@126.com）

证书的签发过程

- 1.主体（服务器、客户端、用户等）
  - 生成密钥对
  - 生成包含其公钥的证书签名请求（CSR）
  - 将CSR提交给CA，等待批准
- 2. CA
  - 核实申请者的身份
  - 用CA的私钥对申请进行签名生成一个有效的证书
  - 颁发证书，以便申请者可使用该证书
- 3. 主体（服务器、客户端、用户等）
  - 获得由CA签名的有效证书以便使用

梁如军（linuxbooks@126.com）

创建自签名证书

```
# cd /etc/pki/tls
# openssl req -new -x509 -days 365 -sha1 \
    -nodes -newkey rsa:2048 \
    -keyout private/ftp.olabs.lan.key \
    -out certs/ftp.olabs.lan.crt \
    -subj '/O=olabs/C=CN/CN=ftp.olabs.lan'
# chmod 600 private/ftp.olabs.lan.key
# openssl rsa -text -in private/ftp.olabs.lan.key
# openssl x509 -text -in certs/ftp.olabs.lan.crt
```


梁如军（linuxbooks@126.com）

创建多域名证书（SAN 证书）

- 多域名证书（SAN 证书）
- 通常也称 UCC（Unified Communications Certificates，统一通信证书）
- 即在一张证书里同时签署多个域名
- 它是 X.509 扩展实现（RFC 2459）
- 使用 subjectAltName指定多个域名，既可以分别指定多个域名，也可以使用通配符的泛域名或两者混用。

梁如军（linuxbooks@126.com）

TCP WRAPPERS

梁如军（linuxbooks@126.com）

TCP Wrappers简介

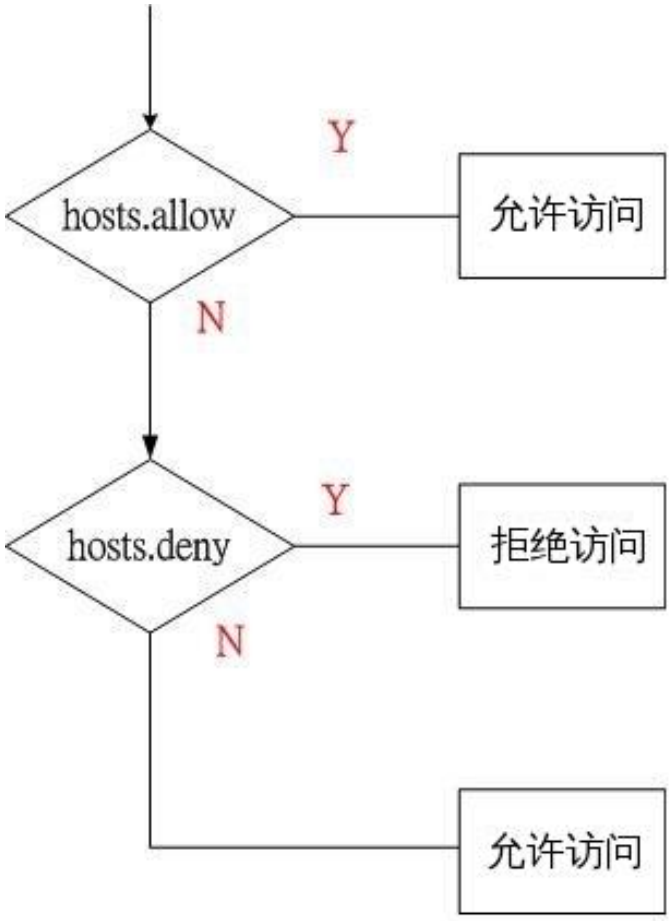
- Transmission Control Protocol Wrappers
- 是一种应用层的网络访问控制程序。
- 由 Wietse Venema 开发的一个免费软件。
- 使用访问控制列表 (ACL) 实现主机访问控制
  - /etc/hosts.allow 是一个许可表
  - /etc/hosts.deny 是一个拒绝表
- 随着广泛的应用逐渐成为一种标准的安全工具
- 在 RHEL/CentOS 中是默认安装的

```
#  yum -y install tcp_wrappers
```


梁如军（linuxbooks@126.com）

TCP Wrappers实现访问控制

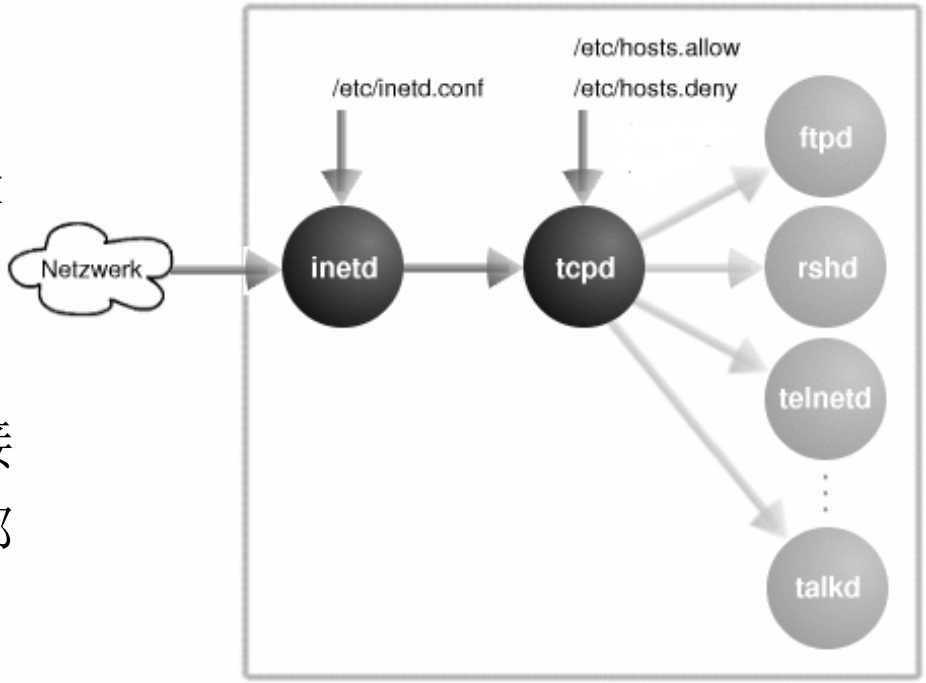
- 读取/etc/hosts.allow文件，如果明确允许访问，则提供访问且不再检查/etc/hosts.deny
- 读取/etc/hosts.deny文件，如果明确拒绝访问，则指定计算机将被拒绝访问。
- 如果两个文件中都没有访问者的计算机名称或IP地址，则自动提供访问权。
- 如果不存在这两个文件或文件内容为空，则 TCP_Wrappers 的访问控制功能被禁用。



梁如军（linuxbooks@126.com）

TCP Wrappers 保护机制的实现方式（1）

- **TCP Wrappers 扩展了 inetd 的功能**
  - 为受inetd控制的服务实施集中式的控制
    - 提供日志支持
    - 返回消息给接入的连接
    - 使服务程序只接受内部连接等
  - **通过 tcpd 守护进程对其他服务程序进行包装**
  - FreeBSD的实现方式



inetd是Xinetd的前身，也是一种超级服务器

梁如军（linuxbooks@126.com）

TCP Wrappers 保护机制的实现方式（2）

- 由网络服务程序调用libwrap.so.*链接库
  - **内置libwrap.so.*库支持的网络服务程序都能使用TCP Wrappers来实现访问控制**
  - 检查TCP Wrappers的支持
    - 显示所有支持TCP Wrappers的软件包

      $ rpm -q --whatrequires libwrap.so.0
    - 查看某网络服务程序对TCP Wrappers的支持

      # ldd $(which sshd) | grep libwrap 或

      # strings -f /usr/sbin/sshd |grep hosts_access
- **RHEL/CentOS中使用了这种实现方式**

梁如军（linuxbooks@126.com）

RHEL/CentOS中支持TCP Wrappers的服务程序

```
# rpm -q --whatrequires libwrap.so.0
sendmail-8.13.8-8.el5
quota-3.13-1.2.5.el5
xinetd-2.3.14-10.el5
tftp-server-0.49-2.el5.centos
openssh-server-4.3p2-41.el5_5.1
vsftpd-2.0.5-16.el5_5.1
nfs-utils-1.0.9-47.el5_5
.....
```

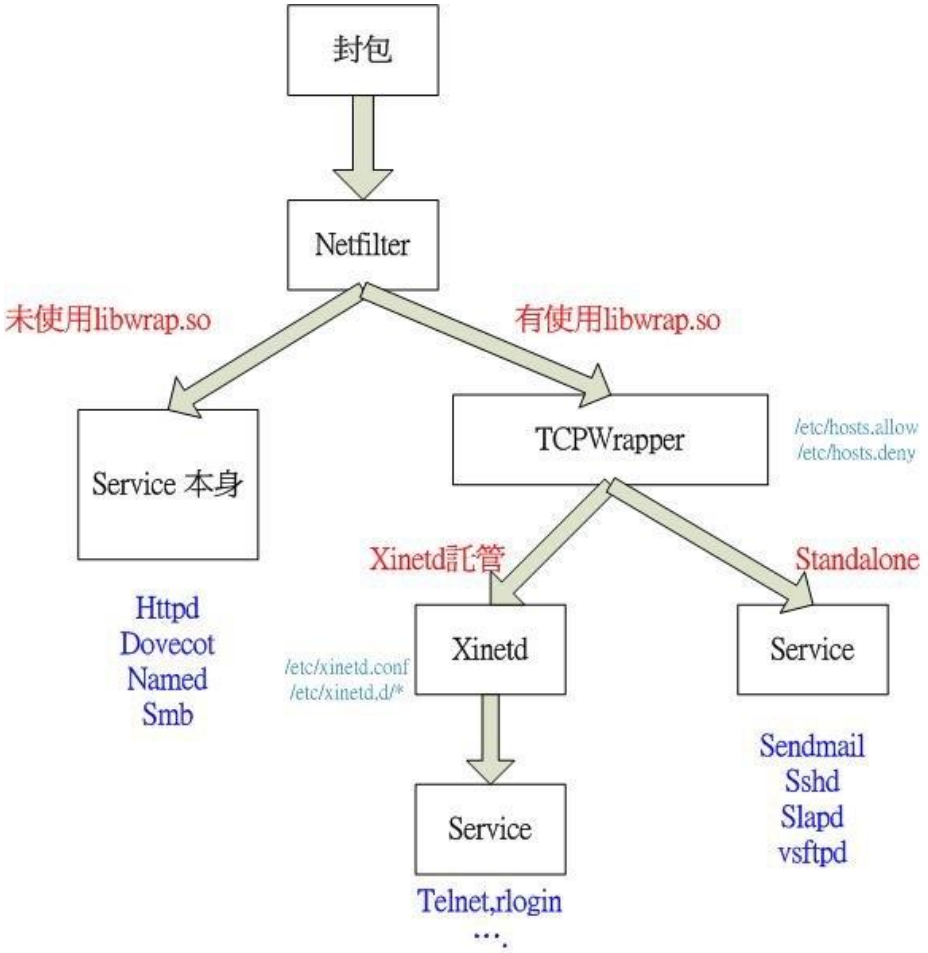
- 对多个服务程序实现集中式管理（**Centralized Management**）

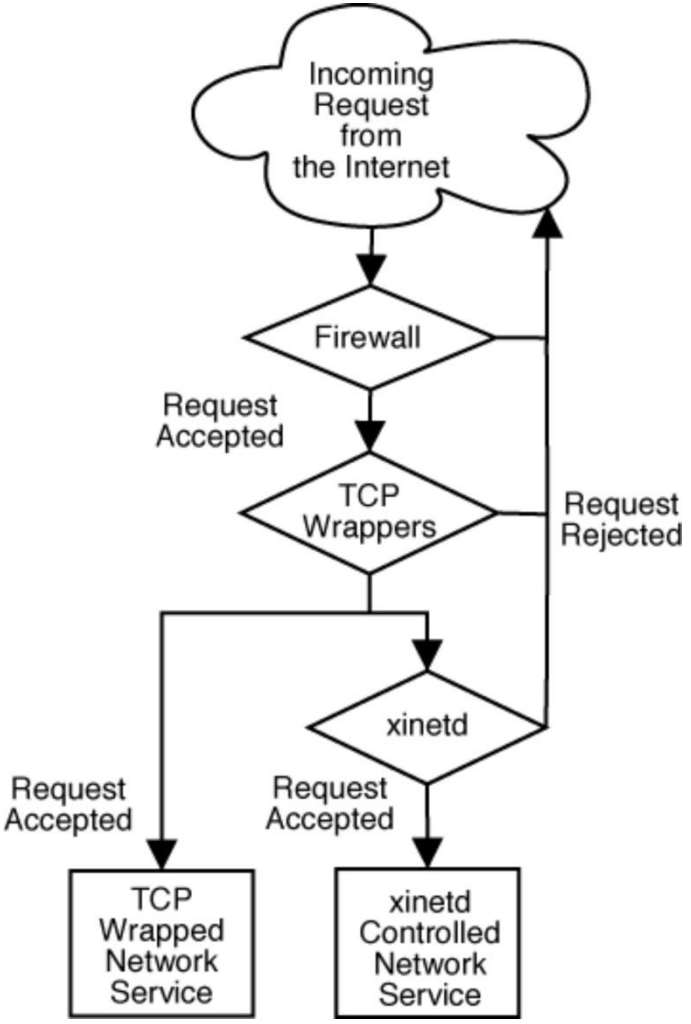
TCP Wrappers与防火墙

- 通常做法是在 Linux 操作系统上安装 Netfilter/iptables 防火墙来处理网络连接。
- 虽然防火墙有非常广泛的用途，但他却不是万能的。
  - 例如它无法处理类似的向连接发起者发送一些文本这样的任务。TCP Wrappers 能够完成它以及更多的其他事情。
- TCP Wrappers 能提供的一些额外的安全功能，但不应被视为好的防火墙的替代品。
- TCP Wrappers 应结合防火墙或其他安全加强设施一并使用，为系统多提供一层安全防护。

梁如军（linuxbooks@126.com）

TCP Wrappers与防火墙图示





梁如军（linuxbooks@126.com）

TCP Wrappers 的配置文件语法

- **/etc/hosts.allow** 和 **/etc/hosts.deny** 语法
  - “#”开始的行为注释；“\”为续行符

任何修改都立即生效，不需要重新启动服务

梁如军（linuxbooks@126.com）

TCP Wrappers 配置文件——宏定义

- 主机名宏定义
  - LOCAL：本地主机
  - KNOWN：可解析域名的主机
  - UNKNOWN：不可解析域名的主机
  - PARANOID：IP与其主机名不符的客户
- 主机和服务宏定义
  - ALL
  - EXCEPT
    - 可嵌套

梁如军（linuxbooks@126.com）

TCP Wrappers 配置文件——主机列表的语法

sshd: centos.example.com 192.168.0.254

sshd: .example.com

sshd: .cracker.org **EXCEPT** trusted.cracker.org

sshd: 123.113.103. 123.113.13.207 **LOCAL**

sshd: 123.113.103. **EXCEPT** 123.113.103.207

sshd: 192.168.0.0**/255.255.254.0**

sshd: 192.168.0.0**/23**

sshd: **ALL**

sshd: **ALL EXCEPT** 192.168.1.

sshd: **ALL EXCEPT** 192.168.1. **PARANOID**

sshd: **/**etc/acl/mylists.hosts

梁如军（linuxbooks@126.com）

TCP Wrappers 配置举例

- 要求仅允许本地主机、192.168.0 网段和 mynet.com 域访问系统中的 telnet 和独立启动的 vsftpd 服务
- 配置过程如下
  - 先编辑 /etc/hosts.deny 拒绝所有主机访问，为此在 /etc/hosts.deny 添加如下行：

```
in.telnetd,vsftpd: ALL
```

  - 再编辑 /etc/hosts.allow 开放允许访问的主机，为此在 /etc/hosts.allow 添加如下行：

```
in.telnetd,vsftpd: LOCAL, 192.168.0., .mynet.com
```


梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——扩展选项的语法

- deny 或 allow
  - 必须作为一条规则的最后一个选项出现
- spawn <SHELL CMD>：在子shell中执行指定命令
  - 标准设备（stdin, stdout 和 stderr）均连接到空设备，不与客户端对话
- twist <SHELL CMD>：使用指定的Shell命令应答服务请求，且执行完后立即终止该次连接请求
  - 标准设备（stdin, stdout 和 stderr）均连接到客户端进程
- severity：设置 syslogd 的 log facility 和 priority level
- setenv：用于设置服务程序的环境参数

梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——扩展选项spawn和twist的宏

- 在 spawn 和 twist 扩展选项中可以使用宏
  - %a（%A） — The client（server）'s IP address.
  - %h（%H） — The client （server）'s hostname.
  - %c（%s） — The client （server） information: user@host, user@address, a host name, or just an address, depending on how much information is available.
  - %d — The daemon process name.
  - %p — The daemon process ID.

梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——扩展选项 spawn举例

CentOS The Community ENTerprise Operating System

```
sshd: <client list> \
: spawn /bin/echo $(/bin/date) from %h >>
   /var/log/sshd.log \
: deny

vsftpd : <client list> \
: spawn /bin/echo "login attempt from %c to
   %s" | mail -s “%d warning" root
```


梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——扩展选项twist举例

CentOS The Community ENTerprise Operating System

```
vsftpd : <client list> \
: twist /bin/echo -e "\n\nWARNING
   connectiong not allowed.\n\n"

in.telnet: <client list> \
: twist /bin/echo "421 Bad hacker, go away!"
```


梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——扩展选项spawn和twist举例

```
in.telnet : ALL \
: spawn (/bin/echo "security notice from host:  %H " ;\
        /bin/echo ; /usr/sbin/safe_finger @%h) | \
        /bin/mail -s "%d-%h security" root & \
: twist /bin/echo -e "\n\nWARNING connectiong not
    allowed.\n\n"
```


梁如军（linuxbooks@126.com）

TCP Wrappers配置文件——注意事项

- 尽量使用 **IP地址**，而不用主机名或域名
- 使用 allow 或 deny 扩展作为规则的**最后一个**选项，可以将所有访问规则集中设置在一个配置文件中
- <daemon list> 为 **process name**，而非 service name。如：in.telnetd，而非 telnet。
- twist 扩展必须用在每一条规则的**最后一个**选项
- /etc/hosts.allow 和 /etc/hosts.deny 的详细语法
  - **man 5 hosts_access**
  - **man 5 hosts_options**

梁如军（linuxbooks@126.com）

TCP Wrappers 的应用

- 对配置文件 **/etc/hosts.allow** 和 **/etc/hosts.deny** 的修改立即生效
- 可以对网络服务实现实时地动态地保护
  - DenyHosts （基于 TCP Wrappers 实现）
    - http://denyhosts.sourceforge.net/
    - 在 RPMForge和EPEL仓库中均有提供
  - Fail2ban （基于 Netfilter/IPTables 实现）
    - http://fail2ban.sourceforge.net/
    - 在 RPMForge和EPEL仓库中均有提供

梁如军（linuxbooks@126.com）

本章思考题

- 简述Linux服务器的基本安全配置？
- 简述PAM的作用及其配置方法。
- 与口令安全相关的PAM模块有哪些？
- 简述SSL协议的握手过程。
- 什么是PKI？什么是CA？证书的组成？

梁如军（linuxbooks@126.com）

本章实验

- 配置服务器的基本安全
- 配置sudo并禁止root直接登录
- 配置基于PAM的账号的口令的安全保护
- 创建自签名证书，创建自签名SAN证书
- 配置 TCP_Wappers实现主机访问控制

梁如军（linuxbooks@126.com）

进一步学习

- 学习chkrootkit（http://www.chkrootkit.org/）的安装、配置和使用（EPEL仓库提供了其RPM包）。
- 学习aide（http://aide.sf.net/）的安装、配置和使用（EPEL仓库提供了其RPM包）。
- 学习基于PAM的各种访问控制（登录/列表/时间/资源）的配置。
- 学习配置xinetd 实现由其管理服务的访问控制。

进一步学习（续）

- 学习配置vsftpd支持SSL/TLS协议的双向认证。
- 学习使用Windows环境下的xca（http://xca.sf.net）实现CA证书的管理。
- 学习使用 https://letsencrypt.org 提供的免费证书
- 学习使用 http://www.startssl.com/ 提供的免费SSL 证书。
  - 请参考 https://github.com/ioerror/duraconf/tree/master/startssl。

信息安全概念

梁如军（linuxbooks@126.com）

安全概述

CentOS
The Community ENTerprise Operating System

- 安全定义
- 安全元素
- 保安措施
- 安全生命周期
- 安全攻击
- 安全服务
- 安全机制
- 网络安全协议

信息安全的定义

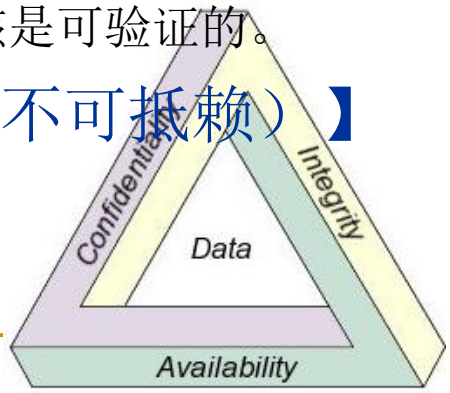
CentOS
The Community ENTerprise Operating System

- Information Security (ISO/IEC 27001:2005)
  - 保持信息的保密性，完整性和可用性。此外，还涉及其他属性：如真实性，问责制，非抵赖性和可靠性。
- Information Security (Wikipedia) = IT Security
  - 信息安全是指保护信息和信息系统免受未经授权的访问和使用，泄露，中断，修改或销毁。
- IT Security
  - IT安全是信息安全的一个子集，关注的是保护计算机 和/或保护计算机中的信息。
- Internet Security (Wikipedia)
  - 互联网安全是计算机安全的一个分支，具体涉及到互联网。它的目标是建立一系列规则和措施来对付互联网上的攻击。

安全元素：The CIA Triad + Extensions

- Confidentiality 【保密性】
  有价值的信息或敏感数据必须受到保护以免遭未经授权的访问
- Integrity 【完整性】
  数据必须被保护，以免在本地存储或在传输过程中意外遭遇恶作剧地更改
- Availability 【可用性】
  在全球商业环境中的服务器和通信基础设施必须保证基于 24/7 的可用。

- Authenticity 【真实性】
  在任何电子交易通信合作伙伴的真实身份（主机/用户）应该是可验证的。
- Accountability (Non-Repudiation)【问责制（不可抵赖）】
  在电子交易和发起实体之间应该有一个可证实的的协会。

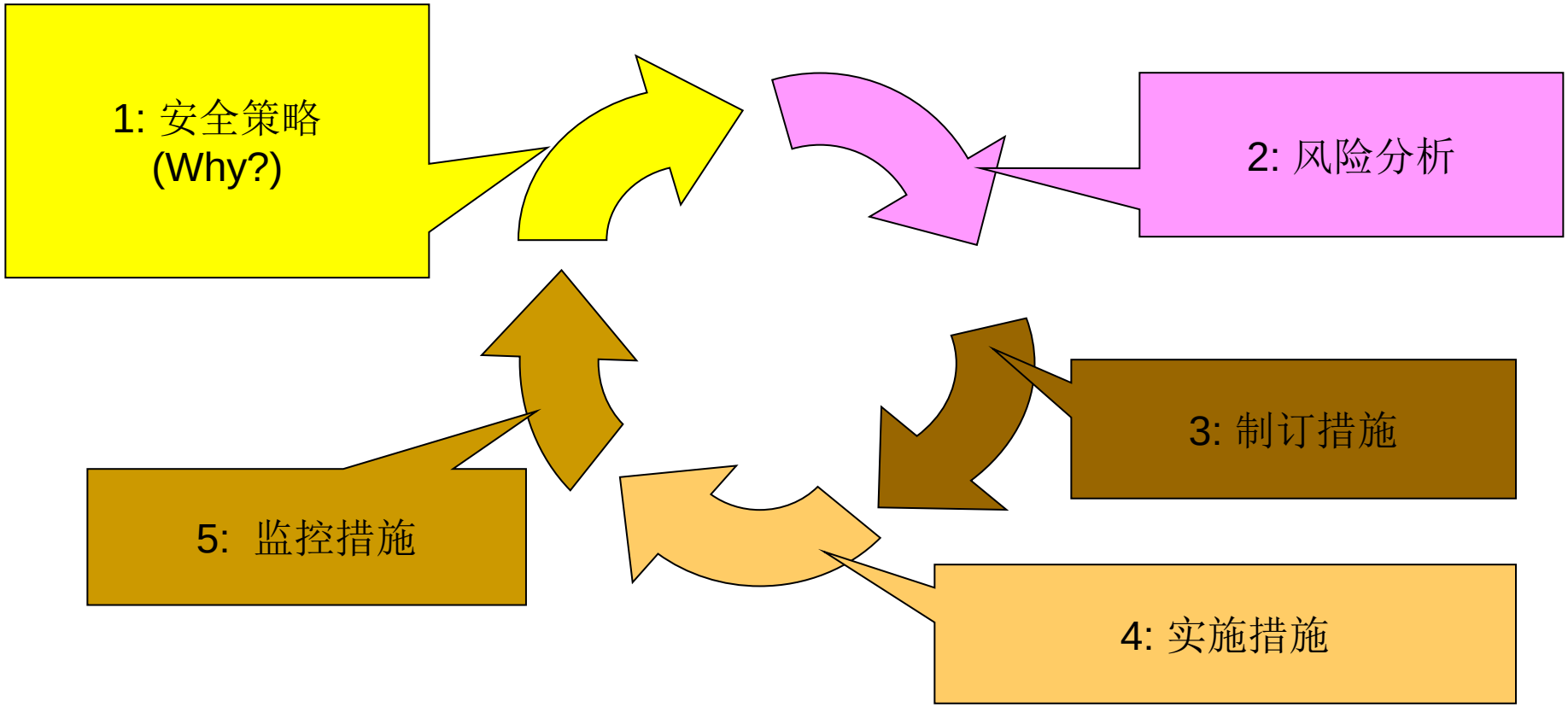


安全措施

- 组织 (计划)
  建立一个安全政策，提高对安全隐患的认识，分析和分类，决定并实施保安措施，明确责任，定期培训员工。
- 保护 (实施)
  加密存储数据和传输的信息，使用认证以确保数据的完整性，安装补丁，并定期检查和使用数据备份机制。
- 过滤器 (实施)
  为用户和主机使用强大的身份验证限制物理系统访问和数据。使用防火墙和病毒扫描过滤流量。
- 结合 (实施)
  结合使用多重安全保护措施（多层/深入的安全）
- 监控 (行动)
  检测攻击（入侵检测系统，蜜罐），运行定期的安全检查（老虎队），及时反应和纠正。

安全的生命周期

CentOS
The Community ENTerprise Operating System



安全环节

- 信息安全的三个环节:
  - 安全攻击
  - 安全机制
  - 安全服务

安全攻击

CentOS
The Community ENTerprise Operating System

- 任何危及某个组织拥有的信息安全的行动
- 通常威胁＆攻击意味着同样的事情
- 有广泛的攻击手段
- 重点关注的通用类型的攻击
  - 被动的
  - 主动的

安全服务

CentOS
The Community ENTerprise Operating System

- 增强了安全性的数据处理系统以及一个组织的信息传输。
- 这些服务是为了对付安全性攻击。
- 使用一种或多种安全机制来提供服务。

安全服务(X.800)

- **验证（Authentication）**
  - 保证每一个通信实体都被验证
- **访问控制（Access Control）**
  - 防止未经授权使用资源
- **数据保密性（Data Confidentiality）**
  - 保护数据免受未经授权的泄露
- **数据完整性（Data Integrity）**
  - 保证接收的数据是被发送方验证的
- **不可抵赖性（Non-Repudiation）**
  - 确保能防止通信的一方当事人否认

安全机制

CentOS
The Community ENTerprise Operating System

- 具体的安全机制：
  - 加密
  - 数字签名
  - 访问控制
  - 数据的完整性
  - 验证交换
  - 流量填充
  - 路由控制
  - 公证

OSI 协议栈的网络安全协议

Communication layers	Security protocols
Application layer	ssh, S/MIME, PGP, Kerberos
Transport layer	SSL, TLS
Network layer	IPsec
Data Link layer	CHAP, PPTP, L2TP, WPA (WLAN), A5 (GSM),
Physical layer	Quantum Cryptography