**pppd命令使用详解70**

======================；关于**pppd**命令使用；======================；Authentication；----------------；PPP协议对称地允许两端都可以要求对端认证自己；**pppd**默认接受对端的认证请求，同时不要求对端认；**pppd**将认证时用到的密语保存在密语文件中，对于；是/etc/ppp/chap-se

======================

关于**pppd**命令使用

======================

Authentication

----------------

PPP协议对称地允许两端都可以要求对端认证自己。这样，两个不同并相互独立的认证交互过程发生，这两个认证交互过程可以使用不同的认证协议。

**pppd**默认接受对端的认证请求，同时不要求对端认证自己本端，但若没有密语，**pppd**将拒绝对端认证自己。

**pppd**将认证时用到的密语保存在密语文件中，对于PAP认证密语文件是/etc/ppp/pap-secrets，对于CHAP、MS-CHAP、MS-CHAPv2和EAP MD5-Challenge认证密语文件

是/etc/ppp/chap-secrets，对于EAP SRP-SHA1认证密语文件是/etc/ppp/srp-secrets。这些密语文件格式相同，内容会像选项文件一样被解析为一个个单词。密语文件既保存了**pppd**认证对端的密语，也保存了提供给对端认证自己本端的密语。每行有三个域保存一个密语：客户端名、服务端名和密语。对于PPP服务端，也可在每行后面跟上一组IP地址表，表示指定IP的客户端才可认证成功。密语以“@”打头表示后续是可以从中读取密语的文件名，客户端名、服务端名若为“\*”表示任意名，IP地址表若为“\*”表示所有IP地址。若客户端名为“-”或者未列出任何IP地址表，表示拒绝所有IP地址客户端的认证。IP地址表以“!”开头表示这些IP地址的客户端会被拒绝，IP地址表后也可跟上“/”加数字表示子网。

当**pppd**认证对端时，在密语文件中搜索匹配行，以对端名字匹配客户端名域，以本端名字匹配服务端域，本端名字默认是本端系统的hostname，若使用domain选项则加上domain name。当**pppd**提供密语给对端认证自己本端时，首先需要确认本端的名字，若有user选项使用该选项，若无此选项默认名字是本端系统的hostname。**pppd**在密语文件中搜索匹配行，以本端的名字匹配客户端名域，以对端名字匹配服务端域。在CHAP或EAP认证中，**pppd**可以从对端发送报文中确定对端名字，但PAP认证中，**pppd**只能从remotename选项中确定对端名字，否则使用空字符串作为对端名字。

当使用PAP认证对端时，对端提供的Password首先与密语文件的密语进行比对，若不匹配，使用crypt()函数对Password进行加密然后再进行比对，所以密语可以是加密形式。若指定了papcrypt选项，第一步的未加密情况下的比对步骤将被跳过。

Routing

---------

当IPCP协商成功完成，**pppd**将设置PPP interface的本地和远端IP地址，这已足够可以创建路由。与别的机器通信通常需要修改路由表和ARP表，大部分情况下，使用defaultroute和proxyarp选项就可以了。若不行，就是用/etc/ppp/ip-up脚本。

Scripts

---------

**pppd**在处理的不同阶段会执行脚步文件，**pppd**并不等待脚本执行完，脚本以root账户来运行，并将脚本的标准输入、标准输出、标准错误重定向到/dev/null，**参数**通过环境变量交换。这些环境变量如下：

DEVICE：使用的串口设备名

IFNAME：使用的网络接口名

IPLOCAL：本端的IP地址，只有IPCP成功协商才设置该值

The IP address for the local end of the link. This is only set when IPCP has come up.

IPREMOTE：远端的IP地址，只有IPCP成功协商才设置该值

PEERNAME：对端认证的名字，这仅在对端认证自己的情况下会设置。

SPEED：串口设备波特率

ORIG\_UID：运行**pppd**进程的用户的实际UID

PPPLOGNAME：运行**pppd**进程的用户的用户名

对于ip-down和auth-down脚本，**pppd**设置下列环境变量提供连接的统计值。

CONNECT\_TIME：从PPP协商开始到连接中断之间的秒数。

BYTES\_SENT：连接期间发送的字节数。

BYTES\_RCVD：连接期间接受的字节数。

LINKNAME：linkname选项设置的链路逻辑名称。

DNS1：第一个DNS server地址。

DNS2：第二个DNS server地址

**pppd**激活下列脚本，即使脚本不退出也不算错。

/etc/ppp/auth-up：当对端认证本端成功后执行该脚本，执行时传递**参数**：interface-name peer-name user-name tty-device speed，若对端不做认证本端该脚本不执行，比如使用了noauth选项。

/etc/ppp/auth-down：之前若执行过/etc/ppp/auth-up脚本，则链路断开时执行该脚本，执行时传递**参数**和/etc/ppp/auth-up相同。

/etc/ppp/ip-up：IPCP协商完成可以收发IP报文时执行该脚本，执行时传递**参数**：interface-name tty-device speed local-IP-address remote-IP-address ipparam

/etc/ppp/ip-down：链路无法再收发IP报文时执行该脚本，该脚本用于撤销执行/etc/ppp/ip-up脚本所带来效果，执行时传递**参数**和/etc/ppp/ip-up相同。

/etc/ppp/ipv6-up：与/etc/ppp/ip-up类似，IPv6CP协商完成可以收发IPv6报文时执行该脚本，执行时传递**参数**：interface-name tty-device speed local-link-local-address remote-link-local-address ipparam

/etc/ppp/ipv6-down：与/etc/ppp/ip-down类似，链路无法再收发IPv6报文时执行该脚本，该脚本用于撤销执行/etc/ppp/ipv6-up脚本所带来效果，执行时传递**参数**和/etc/ppp/ipv6-up相同。

Files

---------

/var/run/pppn.pid：保存在PPP interface unit <n>上运行的**pppd**进程ID。

/var/run/ppp-name.pid：若设置linkname选项，该文件保存逻辑链路名为<name>的**pppd**进程ID。

/etc/ppp/chap-secrets：CHAP/MS-CHAP/MS-CHAPv2认证的安全凭证文件。

/etc/ppp/srp-secrets：EAP认证的安全凭证文件。

/etc/ppp/options：**pppd**默认选项文件。

~/.ppprc：**pppd**选项文件。

/etc/ppp/options.ttyname：**pppd**选项文件。

需要进行鉴权时，通过在options脚本中指定鉴权方式为PAP或CHAP，PPP模块就会从pap-

secrets和chap-secrets脚本中读出用户名和密码，附加到PPP的鉴权包中，发送到服务器端进行身份认证。

chap和pap认证的帐号文件：

/apps/etc/ppp >: cat pap-secrets

# Secrets for authentication using PAP

#User Server Secret IP Address

na\_admin \* 123 \*

/apps/etc/ppp >: cat chap-secrets

# Secrets for authentication using CHAP

#User Server Secret IP Address

na\_admin \* 123 \*

**pppd**命令使用

-------------

命令格式：**pppd** [tty\_name] [speed] [options]

**pppd**默认从选项文件/etc/ppp/options中读取所有配置的选项，若运行**pppd**时通过命令行也指定了选项，对于同时出现在选项文件中的情况，优先使用选项文件中的设置。

关于选项文件：

**pppd**选项取自文件和命令行。**pppd**按照如下顺序读取所有选项：/etc/ppp/options，~/.ppprc，/etc/ppp/options.ttyname，命令行。事实上，**pppd**在读

取/etc/ppp/options.ttyname文件前先在命令行中扫描ttyname。选项文件中若单词包含空格，可以前后使用双引号（"）括起来。一行写不下可以使用反斜杠，以“#”字符打头的行是注释行。选项文件中也可使用file或call这两个选项读取更多选项。

所有可用的选项说明如下：

# tty options

[tty\_name] // 该选项指定使用串口设备ttyname与对端通信。设备名称ttyname可以在前面加上“/dev/”，若未指定设备名称**pppd**就默认使用控制终端并且不会将自己切换到后台执行。

[speed] // 指定串口波特率

lock // 创建一个锁定文件，其他程序发现存在这个文件后，就能得知相应的串口已经被使用。

nolock // 与lock选项功能相反。

modem // 使用调制解调器控制串口线路，此为默认值。**pppd**在打开串口设备时，会等待调制解调器传来的CD（Carrier Detect)信号。

local // 不使用调制解调器的控制信号线，设置该选项，**pppd**会忽略调制解调器的CD（Carrier Detect)信号的状态。该选项与modem选项功能相反。

crtscts // 使用硬件流控

nocrtscts // 关闭硬件流控

xonxoff // 使用软件流控（XON/XOFF）控制串口的数据流量

cdtrcts // 使用非标准的硬件流控

nocdtrcts // 关闭硬件流控

asyncmap <map> // 设置ACCM（Async-Control-Character-Map）为**参数**<map>，**参数**<map>是一个以“0x”打头的16进制数。ACCM是一组32位数字，每一个bit位表示一个ASCII控制符，置为1，对应的ASCII控制符在PPP帧中不能直接用ASCII码表示，**pppd**要求对端将这些字符以escape sequence发送。若未指定该选项，默认ACCM数为0，即**pppd**不要求对端以escape sequence发送任何控制字符。对于接收的escape sequence发送的控制字符还原，还需指定escape选项。

default-asyncmap // 关闭asyncmap协商，强制将所有的ASCII控制符在收发方向上进行escape。

sync // 使用同步HDLC串行编码取代异步编码。

# global options

nodetach // **pppd**成功建立连接（即拨号成功）继续在前台执行updetach // **pppd**成功建立连接（即拨号成功）后切换到后台执行file <file> // 指定从<file>文件中读取选项

call <name> // 从/etc/ppp/peers/name文件中读取更多的**pppd**命令选项init <script> // 执行<script>脚本初始化串口，通常这个脚本中执行chat程序配置modem开启自动应答

unit <n> // 设置PPP unit number，创建的PPP网络接口对应名字为ppp<n>。

debug // 开启调试信息

kdebug <n> // 开启内核中的PPP协议模块代码的调试。**参数**<n>可以是下列值的组合：1 - 开启一般调试信息；2 - 要求打印出所接收到的PPP帧数据内容；4 - 要求打印出发送的PPP帧数据内容。

logfd <n> // 将**pppd**执行中的log信息输出到文件描述符<n>中。**pppd**只能将log信息输出到一个文件中，所以该选项与logfile选项只能取其一。默认**pppd**将log信息输出到stdout（即文件描述符为1），除非串口终端已经被占用。

logfile <file> // 将**pppd**执行中的log信息以追加的方式输出到<file>中

nolog // 关闭将**pppd**执行中的log信息输出到文件或文件描述符。该选项将取消logfd和logfile选项的功能。

linkname <name> // 设置链路的逻辑名称为<name>。**pppd**将创建/var/run/ppp-<name>.pid文件保存自己的进程ID，这个可以用于确定哪个**pppd**进程负责处理特定对端的链路。

notty // 通常，**pppd**需要一个终端设备，由tty\_name给出。若设置该选项，则tty\_name不用指定。设置该选项，**pppd**会给自己分配一个伪终端设备，并使用伪终端从设备（slave）。**pppd**还会创建一个子进程作为“character shunt”通过标准输入输出和伪终端主设备之间传输字符。这个选项会造成在PPP interface上收发数据增加延时和CPU负载，因为所有数据收发都必须穿过这个子进程。

pty <script> // 指定**pppd**与<script>命令脚本通信而不是通常的串口设备通信，**pppd**将创建伪终端设备并使用从设备（slave），脚本以子进程形式运行并使用伪终端主设备（master）作为其标准输入（stdin）和标准输出（stdout）。若设置该选项，则无需再设置ttyname。若record选项与该选项联用，脚本子进程将使用管道（pipe）作为标准输入和标准输出。

record <filename> // 设置**pppd**记录所有收发的字符到<filename>文件中，该文件以追加模式打开。该选项在使用伪终端设备时有效。字符以带有时间戳标签的形式保存，可以使用pppdump工具显示出来。

show-password // 该选项将使**pppd**输出PAP报文内容的log信息时，显示password字符串。

hide-password // 该选项将使**pppd**输出PAP报文内容的log信息时，不显示password字符串。默认设置该选项。

plugin <filename> // 装载共享库文件作为插件。若<filename>未指定路径，则默认从/usr/lib/**pppd**/<version>/目录下搜索共享库文件。

child-timeout <n> // **pppd**进程退出前等待<n>秒时间等待子进程退出，若时间超出，**pppd**进程将给所有剩下的子进程发送SIGTERM信号并退出。若设置为0，表示**pppd**永远等待直到所有子进程退出。

dryrun // **pppd**打印出所有已设置的选项值然后退出。

dump // **pppd**打印出所有已设置的选项值然后继续正常运行。ipparam <string> // 给ip-up、ip-pre-up和ip-down脚本提供附加的**参数**，<string>字符串作为这些脚本的第六个**参数**。

ipx // 开启IPXCP和IPX协议，内核需要配置支持IPX。

noipx // 关闭IPXCP和IPX协议。

proxyarp // 将对端IP和MAC地址添加进系统ARP表中。

noproxyarp // 关闭proxyarp选项。

ktune // 允许**pppd**修改一些内核设置，在Linux下，若设置了proxyarp选项，**pppd**设置/proc/sys/net/ipv4/ip\_forward为1用于开启IP转发，若设置了demand选项，**pppd**设置/proc/sys/net/ipv4/ip\_dynaddr为1用于开启动态IP地址。

noktune // 不许**pppd**修改内核设置。

ms-dns <addr> // 该选项一般用于PPP服务端。若对端是运行Windows操作系统的设备，这个选项允许**pppd**给对端提供一到两个DNS地址，第一个该选项指定主DNS地址，第二个选项若有则指定次DNS地址。

ms-wins <addr> // 该选项一般用于PPP服务端。若对端是运行Windows操作系统或“Samba”的设备，这个选项允许**pppd**给对端提供一到两个WINS（Windows Internet Name Services）服务器地址，第一个该选项指定主WINS地址，第二个选项若有则指定次WINS地址。<local\_IP\_address>:<remote\_IP\_address> // 设置本地和远端的IP地址（本地IP地址即本端的IP地址，远程IP地址即对端的IP地址），两者都可忽略不写，IP地址可以指定为hostname或者点分十进制格式。默认的本地IP地址是系统的第一个IP地址（除非指定了

noipdfault选项）。若未在该选项中指定远端IP地址，则从对端中获取。所以，一般无需指定

该选项。如果使用该选项指定了本地和远端IP地址，**pppd**将不再接受通过IPCP协商获取的不同的IP地址，除非又指定了ipcp-accept-local和ipcp-accept-remote选项。

ipv6 <local\_interface\_identifier>,<remote\_interface\_identifier> // 设置本地和远端的IPv6地址中的最后64位InterfaceID字段，两者都可忽略不写，InterfaceID必须是IPv6地址的标准形式给出（比如，::dead:beef）。若指定了ipv6cp-use-ipaddr选项，本端的本地IPv6地址的InterfaceID字段就使用该选项中指定的<local\_interface\_identifier>。若指定了ipv6cp-use-persistent选项，则使用EUI-64根据MAC地址自动生成，此时该选项设置值将不再有效。

# link options

connect <script> // 在PPP协商之前，先执行<script>脚本

connect-delay <n> // 执行完connect <script>选项的脚本后等待<n>毫秒，再进行PPP协商。默认为1000毫秒。

disconnect <script> // **pppd**断开链路后，执行<script>脚本

welcome <script> // 在PPP协商之前，执行完connect选项的<script>脚本之后，执行该选项<script>脚本。

defaultroute // IPCP协商完成之后，在系统路由表中添加一个默认路由表项，将对端作为网关

nodefaultroute // 关闭defaultroute选项。

mru <n> // 设置MRU为<n>字节，<n>大小在128到16384之间，默认是1500，对于IPv6协议，MRU必须大于1280。**pppd**告诉对端发送报文长度勿超过<n>字节。

mtu <n> // 设置PPP Interface（PPP网络接口，如ppp0）的MTU为<n>字节，对于IPv6协议，MTU必须大于1280。

default-mru // 关闭MRU协商。设置该选项，**pppd**将使用默认MRU值1500。noaccomp // 关闭收和发方向上的Address/Control压缩。

noccp // 关闭CCP（压缩控制协议）的协商。

nopcomp // 关闭收和发方向上PPP帧协议字段压缩协商。

novj // 关闭收和发方向上VJ（Van Jacobson）方式的TCP/IP头压缩。

novjccomp // 在VJ（Van Jacobson）方式的TCP/IP头压缩里，关闭connection-ID压缩选项。

noipdefault // 当本地IP地址未指定时关闭使用hostname生成本地IP地址的行为。使用该选项，对端在IPCP协商时必须指定本端的本地IP地址。

usepeerdns // 向对端请求两个DNS server地址。对端提供的两个地址通过环境变量DNS1和DNS2传给脚本/etc/ppp/ip-up，同时**pppd**创建/var/run/ppp/resolv.conf文件保存nameserver行。

deflate <nr>,<nt> // 数据包传输时使用Deflate压缩方式，若未指定<nt>，则<nt>等于<nr>，<nt>和<nr>的值最好在9到15范围内。要求对端发送数据压缩时最大窗口时2的<nr>次方，本端发送数据压缩时最大窗口是2的<nt>次方。

nodeflate // 关闭数据包传输时使用Deflate压缩。

multilink // 开启使用PPP multilink。

mp // 等同于multilink选项。

nomultilink // 关闭使用PPP multilink。

nomp // 等同于nomultilink选项。

nomagic // 关闭magic number协商。

persist // 永久连接（即自动重拨）。**pppd**进程在连接中断后不退出并重新进行连接。maxfail选项即使对于永久性连接也有效。

nopersist // 建立连接后若断开则**pppd**退出，这是默认值。

demand // 按需连接（即有数据流量时再重拨）。**pppd**驻留在后台，监视网络流量，一旦有数据流量就启动连接，超时后就断开连接，但**pppd**仍然驻留在后台。使用该选项，必须确保使用<local\_IP\_address>:<remote\_IP\_address>选项设定好远端IP地址。一开始**pppd**还没有完成和对端的连接，**pppd**直接配置好并启用网络接口，当有流量时，**pppd**进行PPP协商、认证从而连接到对端，这些完成之后**pppd**将在该链路上传输网络层数据。demand选