Wifidog编译

Wifidog编译需要radius-client的动态连结库

先说明radius-client编译动态连结库方法：

在wifidog1.4版本中，主要使用的是freeradius-client下src中的radacct.c和radlogin.c，分别对其进行修改，去掉他们其中的main函数（radacct.c和radlogin.c都是独立程序），为了使用源文件内的函数模块，将其编译成动态连结库。

在未修改时，它们编译成可执行程序的命令如下：（以radacct.c举例）

libtool: link: gcc -g -O2 -o .libs/radstatus radstatus.o ../lib/.libs/libfreeradius-client.so -lcrypt -lnsl

gcc -DHAVE\_CONFIG\_H -DRC\_CONFIG\_FILE=\"/usr/local/etc/radiusclient/radiusclient.conf\" -DRC\_SECURE\_PATH=\"/bin:/usr/bin:/usr/local/bin\" -I. -I.. -I. -I../include -I.. -g -O2 -MT radacct.o -MD -MP -MF .deps/radacct.Tpo -c -o radacct.o radacct.c

mv -f .deps/radacct.Tpo .deps/radacct.Po

/bin/bash ../libtool --tag=CC --mode=link gcc -g -O2 -o radacct radacct.o ../lib/libfreeradius-client.la -lcrypt -lnsl

libtool: link: gcc -g -O2 -o .libs/radacct radacct.o ../lib/.libs/libfreeradius-client.so -lcrypt -lnsl

实际上就是

1. 编译目标文件：

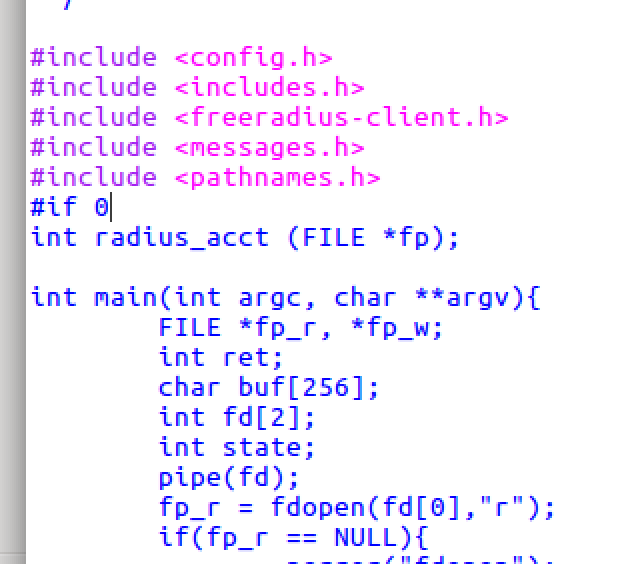
gcc -g -O2 -o .libs/radstatus radstatus.o ../lib/.libs/libfreeradius-client.so -lcrypt -lnsl

gcc -DHAVE\_CONFIG\_H -DRC\_CONFIG\_FILE=\"/usr/local/etc/radiusclient/radiusclient.conf\" -DRC\_SECURE\_PATH=\"/bin:/usr/bin:/usr/local/bin\" -I. -I.. -I. -I../include -I.. -g -O2 -MT radacct.o -MD -MP -MF .deps/radacct.Tpo -c -o radacct.o radacct.c

1. 编译出可执行程序：

gcc -g -O2 -o .libs/radacct radacct.o ../lib/.libs/libfreeradius-client.so -lcrypt –lnsl

当修改radacct.c后，修改如下：



我使用了#if 0 做开关，将main屏蔽了。再进行编译：

在src目录下编译

make clean

make

/bin/bash ../libtool --tag=CC --mode=link gcc -g -O2 -o radacct radacct.o ../lib/libfreeradius-client.la -lcrypt -lnsl

libtool: link: gcc -g -O2 -o .libs/radacct radacct.o ../lib/.libs/libfreeradius-client.so -lcrypt -lnsl

/usr/lib/gcc/x86\_64-linux-gnu/5/../../../x86\_64-linux-gnu/crt1.o: In function `\_start':

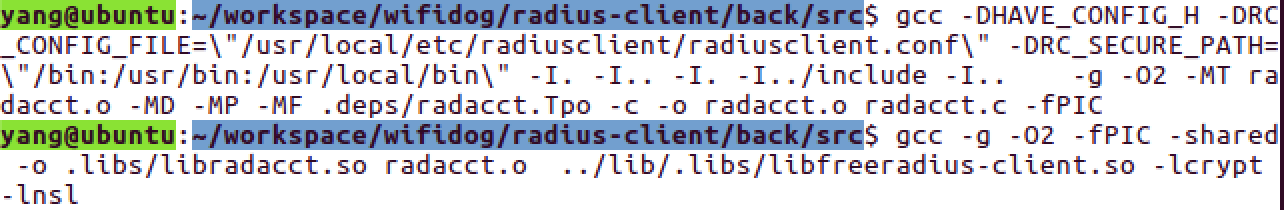
(.text+0x20): undefined reference to `main'

collect2: error: ld returned 1 exit status

Makefile:431: recipe for target 'radacct' failed

make: \*\*\* [radacct] Error 1

必然的错误，没有main函数，无法生成可执行程序，我利用给出的命令制作我需要的动态连结库编译命令。命令如下：



gcc -DHAVE\_CONFIG\_H -DRC\_CONFIG\_FILE=\"/usr/local/etc/radiusclient/radiusclient.conf\" -DRC\_SECURE\_PATH=\"/bin:/usr/bin:/usr/local/bin\" -I. -I.. -I. -I../include -I.. -g -O2 -MT radacct.o -MD -MP -MF .deps/radacct.Tpo -c -o radacct.o radacct.c -fPIC

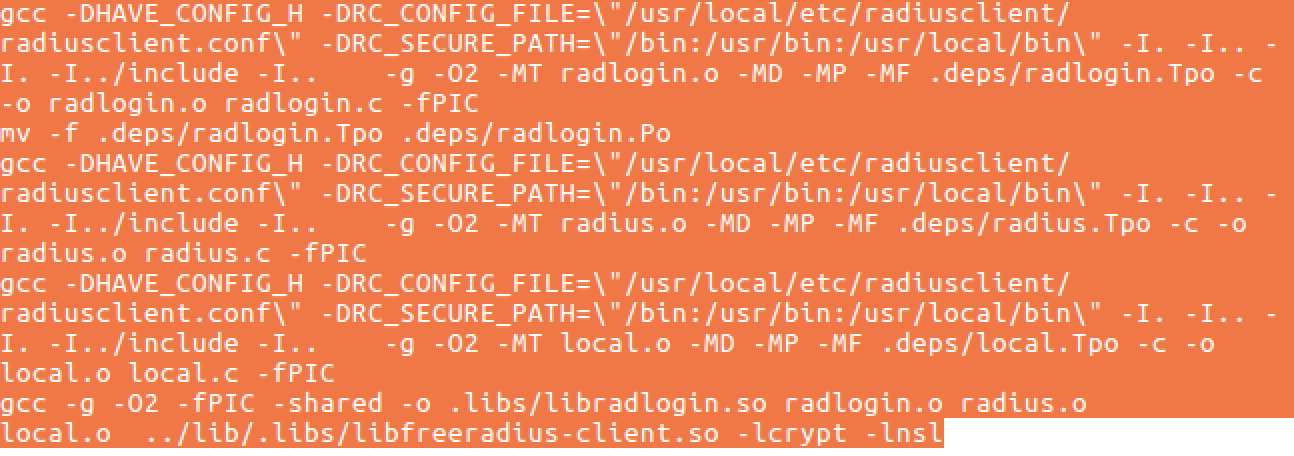
gcc -g -O2 -fPIC -shared -o .libs/libradacct.so radacct.o ../lib/.libs/libfreeradius-client.so -lcrypt -lnsl

第一条命令在上述【编译目标文件】的命令后面加了 –fPIC，为了后面编译动态连结库。

第二条命令在上述【编译出可执行程序】的命令中添加了 –fPIC –shared 并重命名了生成的文件名即我需要的动态连结库。

至此羊氏动态连结库编译大法原理讲解完毕，radlogin.c编译动态连结库同理，只不过是需要讲radacct.c的#if 0 再次改为#if 1，然后讲radlogin.c的#if 1 改为#if 0，然后再make clean ，make，这为了只让radlogin.c编译出错。

需要注意的是：radlogin依赖三个目标文件所以命令如下



这两个动态连结库都在src／.lib下面

wifidog的编译也是类似的