Friday, May 20, 2011

# libpcap for Android

The Android project includes libpcap as an external library as can be seen from the source here on [kernel.org](http://android.git.kernel.org/?p=platform/external/libpcap.git;a=summary) and[Github](https://github.com/android/platform_external_libpcap)  
  
My project for Google Summer of Code - The mobile network scanner requires the use of libpcap or tcpdump. I was successful in compiling libpcap from its source and loading the library to use the native functions in JNI. But since an Android application does not run with Root privileges, it is a challenge to get the native functions to work in the same process.  
  
The native code and the Java code for an android application runs in the same process. So the native code does not have root privileges. So as of now, it is not possible to get libpcap functions to work using the NDK. But there are alternatives which I will be suggesting in this blog post.  
  
First of all, here are the instructions to successfully compile the libpcap.so library for use in the JNI code.  
  
Create a folder called jni in the application root.

**Android.mk**

LOCAL\_PATH := ./jni

include $(CLEAR\_VARS)

LOCAL\_MODULE := pcaptest

LOCAL\_SRC\_FILES := libpcap-native.c

LOCAL\_C\_INCLUDES := $(NDK\_ROOT)/external/libpcap

LOCAL\_STATIC\_LIBRARIES := libpcap

LOCAL\_LDLIBS := -ldl -llog

include $(BUILD\_SHARED\_LIBRARY)

include $(NDK\_ROOT)/external/libpcap/Android.mk

Libpcap for android is built as a static library and its functions are then used as a shared library.  
A shared library build specifications are defined in the Android.mk make file in the JNI folder of the project. The JNI folder contains the make files for the library. It also contains the native C code with function definitions according to the Java package.  
  
  
This is a sample JNI code for getting the list of available network interfaces by using the pcap\_lookupdevs function

#include <jni.h>

#include <string.h>

#include <android/log.h>

#include <pcap.h>

#define DEBUG\_TAG "Sample\_LIBPCAP\_DEBUGGING"

void Java\_org\_umit\_android\_libpcaptest\_libpcaptest\_testLog(JNIEnv \*env, jclass clazz, jstring message)

{

char errbuf[1024];

errbuf[0] = '\0';

char \*szLogThis;

char \*dev = pcap\_lookupdev(errbuf);

if (dev == NULL) {

szLogThis = "Couldn't find default device";

}

else szLogThis = dev;

\_\_android\_log\_print(ANDROID\_LOG\_DEBUG, DEBUG\_TAG, "Device status: [%s]", szLogThis);

\_\_android\_log\_print(ANDROID\_LOG\_DEBUG, DEBUG\_TAG, "errbuf [%s]", errbuf);

(\*env)->ReleaseStringUTFChars(env, message, szLogThis);

}

The C code can now be called as native functions from inside Java code by loading the shared library

static{

System.loadLibrary("pcaptest");

}

private native void testLog(String logThis);

The native code needs to compiled with ndk-build command and requires the android-ndk to be downloaded from the official website.  
  
For now, I am getting the error

D/Sample\_LIBPCAP\_DEBUGGING( 364): Device status: [Couldn't open device]

D/Sample\_LIBPCAP\_DEBUGGING( 364): errbuf [socket: Operation not permitted]

And when I try and use the pcap\_open\_live function, I get the following error  
 D/Sample\_LIBPCAP\_DEBUGGING(  310): errbuf [socket: Operation not permitted]  
  
The reason for this error is that the native code and the application both lie in the same process. Invoking a "su" here would not work because an "su" just forks a new process that has root privileges but that process would not contain our native code.  
  
So a work around for this is that a binary or a unix library should be compiled before hand and then supplied with the apk. The binary can be extracted from the apk. The binary may require root privileges but it would be as easy as forking another process with "su" command.  
  
Here is a blog post explaining that <http://nerdposts.blogspot.com/2010/11/android-superuser-shared-jni-libs.html>  
  
For now, I am just sticking to tcpdump for the requirements of this project. If we need a specific implementation of packet sniffing from libpcap, the plan is to compile a binary or a unix shared library and supply it with the apk.

Posted by [Angad](https://www.blogger.com/profile/15375358083423396516)at [4:38 PM](http://blog.umitproject.org/2011/05/libpcap-for-android.html) [[http://img1.blogblog.com/img/icon18_email.gif](https://www.blogger.com/email-post.g?blogID=19489352&postID=1663815035828931188)](https://www.blogger.com/email-post.g?blogID=19489352&postID=1663815035828931188)

[Email This](https://www.blogger.com/share-post.g?blogID=19489352&postID=1663815035828931188&target=email)[BlogThis!](https://www.blogger.com/share-post.g?blogID=19489352&postID=1663815035828931188&target=blog)[Share to Twitter](https://www.blogger.com/share-post.g?blogID=19489352&postID=1663815035828931188&target=twitter)[Share to Facebook](https://www.blogger.com/share-post.g?blogID=19489352&postID=1663815035828931188&target=facebook)[Share to Pinterest](https://www.blogger.com/share-post.g?blogID=19489352&postID=1663815035828931188&target=pinterest)

|  |  |
| --- | --- |
| Reactions: |  |

# 8 comments:

1. https://img2.blogblog.com/img/b36-rounded.png

[**przemoo**](https://www.blogger.com/profile/14411167295715598011)[Sunday, November 06, 2011 6:17:00 AM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1320567437418#c4786880979439751105)

Did you try to implement libcap on android? I have similar needs and I'd like to know if somebody succeded in it before me.

[Reply](javascript:;)

1. https://img2.blogblog.com/img/b36-rounded.png

[**przemoo**](https://www.blogger.com/profile/14411167295715598011)[Sunday, November 06, 2011 6:19:00 AM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1320567556406#c5821976152322593036)

Did you try to implement libcap on android did you find any other way to make a packet sniffer? I have similar project and I'd like to know if someone succeded in this.

[Reply](javascript:;)

1. https://img2.blogblog.com/img/b36-rounded.png

[**Urizev**](https://www.blogger.com/profile/04490741450899678329)[Thursday, December 29, 2011 3:37:00 PM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1325180270664#c1062055000553344208)

Thanks for the post and the work behind. It is really helpful.  
  
I have some troubles compiling the libpcap with NDK. You can found the trouble here:  
  
http://stackoverflow.com/questions/5952878/android-ndk-trying-to-port-jnetpcap

[Reply](javascript:;)

1. http://lh4.googleusercontent.com/-Ljnesfzq4t0/AAAAAAAAAAI/AAAAAAAAAAA/3D9-kFmEPcI/s35-c/photo.jpg

[**kambala balasubrahmanyam**](https://www.blogger.com/profile/13831130905826494768)[Monday, April 02, 2012 2:59:00 PM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1333389546320#c7878558126388970908)

do any one is able implement to packet sniffing using above said process?

[Reply](javascript:;)

1. https://img2.blogblog.com/img/b36-rounded.png

[**Luis Bastiao Silva**](https://www.blogger.com/profile/00275964013369182892)[Sunday, April 22, 2012 8:53:00 AM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1335095593721#c4300477099064425247)

This two projects should be useful:   
  
https://github.com/umitproject/pm-mobile  
https://github.com/umitproject/ns-mobile

[Reply](javascript:;)

1. http://lh4.googleusercontent.com/-hODXQS5xrlM/AAAAAAAAAAI/AAAAAAAAAJU/ux0gTw0lREA/s35-c/photo.jpg

[**Nils Kuhnhenn**](https://www.blogger.com/profile/02246682523121471067)[Thursday, June 14, 2012 6:08:00 AM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1339664917832#c980725671736830067)

I'm currently developing a packet sniffer for android.  
My executable sends the captured packets via loopback to my application.  
The application also contains an ARP-spoofer and is able to display to packets wireshark like (extracting and displaying the packet headers in a human-readable format).  
TCP-Stream reassembly/injection and the extraction of passwords/sessions from some protocols is yet to come.  
Screenshots and information can be found on my blog (German):  
http://4meme.de  
  
I'll release the source-code soon enough on github.

[Reply](javascript:;)

1. https://img2.blogblog.com/img/b36-rounded.png

[**Luis Bastiao Silva**](https://www.blogger.com/profile/00275964013369182892)[Monday, October 22, 2012 1:37:00 AM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1350877024375#c9155715741674978100)

That's awesome! Have you forked https://github.com/umitproject/pm-mobile?  
  
By the way, is it open source? :)   
  
But it seems quite nice. Congrats.

[Reply](javascript:;)

1. http://lh5.googleusercontent.com/-MFHOpDaGwBc/AAAAAAAAAAI/AAAAAAAAASo/NczjKuCIwzo/s35-c/photo.jpg

[**andrea cappellotto**](https://www.blogger.com/profile/12089365257359084320)[Monday, March 18, 2013 1:10:00 PM](http://blog.umitproject.org/2011/05/libpcap-for-android.html?showComment=1363623024680#c1697442205655456211)

Hi!! i'm working on a same thing, i'm be able to run libpcap with root user, finally did you found any solution?  
thanks  
Andrea

[Reply](javascript:;)