This article is to demonstrate how to install libraries and override the existing ones, if needed. Airodump will be used for this example. Many of you who have tried using the aircrack package (or several other apps) have experienced odd errors. In this example, we will be exploring Airodump, a part of the AirCrack package, which would crash with an error relating to fopen64.

Contents

- 1 Why doesn't it work?
- 2 What do I need?
- 3 How do I get aircrack?
- 4 Will Aireplay work?
- 5 Testing
- 6 Getting the Library
- 7 Making Sure Your Library Path Is Set
    - 7.1 It's not there! HELP!
- 8 The Workarounds
    - 8.1 Method A: LD_PRELOAD
    - 8.2 Method B: LD_LIBRARY_PATH
- 9 How to use Airodump with dd-wrt
- 10 Closing Notes

Why doesn't it work?

The problem here is that many math functions and several other functions have been removed from the libraries to conserve space. I was informed, however, that dd-wrt will soon have a new version which will have a writable root filesystem. This will only be for the mini version. However, BrainSlayer mentioned adding a lot back to the libc to fix the errors. So if you have installed airodump on a version higher than the current (v23), it may work without using the workaround.

What do I need?

- jffs needs to be enabled
-Basic handle on using the command line
-Using ipkg
-Using Aircrack
-A halfway decent knowledge of what you're doing

How do I get aircrack?

You can obtain and install the aircrack version for OpenWrt by doing this:

```
wget http://downloads.openwrt.org/people/nbd/ar7/packages/aircrack_2.41-1_mipsel.ipk
ipkg install aircrack_2.41-1_mipsel.ipk
```

Note: Check out the wiki for ipkg. Also the OpenWrt wiki for Aircrack.

Note: URL for aircrack_2.4r1-1_mipsel.ipk updated 7 May 2006.

Will Aireplay work?

Unfortunately, I've not been able to get it to inject packets. I'm told it's not possible with the broadcom drivers. I believe a patch to the driver or something similar would fix this. I'm currently exploring this option, and have several interesting leads so far, but I'll have to dig into it when I get more time on my hands. If I get this, I will definitely post it! If anyone has any info or wishes to help, please let me know!

Testing

If you've gotten this far, try to run airodump. If you don't get an error, then you're good to go! However, many people will get an error such as this:

```
/jffs # airodump prism0 cap 6 1
airodump: can't resolve symbol 'fopen64'
```

If that's you, read on..

Getting the Library

The solution to this particular problem and a number of others you may encounter with applications you try to run on your dd-wrt is that you do not have the correct library. For this problem, we need the uclibc library. We obtain this by using the wonderful ipkg tool.

```
/jffs # ipkg list|grep uclibc
uclibc   - A standard c   library for embedded systems
uclibc - Standard C library for embedded Linux systems
```

Fortunately, in this case, the library we need is on the main list! If you're not so lucky, you'll need to search out a location of a library package.

(Please note that you should only get a library which is definitely compatible with this os/hardware. It's best to search for an OpenWrt package, and try, as we have with this one).

To install the uclib library run:

```
ipkg install uclibc
```

Making Sure Your Library Path Is Set

Unfortunately, we can't simply stop here. The problem we encounter is that the library must be loaded. Your jffs library paths should be included in the LD_LIBRARY_PATH environment variable. To make sure, type

'env' and verify that the library variable contains jffs locations. It should look something like this:

```
/jffs # env | grep LIBRARY
LD_LIBRARY_PATH=/lib:/usr/lib:/jffs/lib:/jffs/usr/lib:/jffs/usr/local/lib
```

It's not there! HELP!

If the jffs paths are not in this for some reason, you add them by changing this variable. You'd want to copy your current value for this variable and add the new paths, separated by colons. So for instance, if I have **LD_LIBRARY_PATH=/lib:/usr/lib** I would run

```
export LD_LIBRARY_PATH=/lib:/usr/lib:/jffs/usr/lib:/jffs/usr/local/lib
```

Note: This will need to be set each time your router reboots. You can create a quick script which runs this.

The Workarounds

Now that you have the library paths set, you have two methods which you can use to make this work. As I already mentioned, there is a pre-existing libc library which was stripped down, so this is being loaded first, and thus the new library is not effective.

Method A: LD_PRELOAD

This is the method I advise that you use. This simply sets the library you've downloaded (uClibc) to load before any others, and thus, it is loaded instead of the standard libc:

```
export LD_PRELOAD=/jffs/lib/libuClibc-0.9.27.so
env
```

Note: Obviously, your library may be named differently, as versions will change. You need to find your uClibc so file. You can do this using:

```
find /jffs -name "*libuClibc*.so"
```

You can also check the status to make sure it installed by using

```
ipkg status uclibc
```

You can test by running 'env' to make sure this variable got set. When you're done using the app that requires this variable, you can simply unset the variable by the following command:

```
unset LD_PRELOAD
env
```

Method B: LD_LIBRARY_PATH

This method involved changing the order in which the library paths are loaded, and works as well. This may be useful for some who want to override several libraries. I don't reccommend this, but if you want to use this way, I assume you know what you're doing and why you want it.

For this, we simply change the LD_LIBRARY_PATH, as mentioned above, placing JFFS library paths first. So, if, for example, LD_LIBRARY_PATH=/lib:/usr/lib:**/jffs/usr/lib:/jffs/usr/local/lib** you would put the jffs first, like so:

```
export LD_LIBRARY_PATH=/jffs/usr/lib:/jffs/usr/local/lib:/lib:/usr/lib
env
```

How to use Airodump with dd-wrt

One other thing I thought I would mention is a good way to make airodump effective by writing output to a computer on the lan. Here's a quick example of mounting a windows share "admp" on 192.168.1.100, with its username: "NaHeMiA":

```
mkdir /tmp/win
smbmount //192.168.1.100/admp /tmp/win -o rw,username=NaHeMiA
df -h
```

It should ask for your password, you type it in, and df -h will show you if it is now mounted as /tmp/win

You can now run airodump, outputting to this file. This is an example of writing to "capturefile" on the admp share, capturing IV packets on channel 6:

```
wl monitor 1
ifup prism0
airodump prism0 /tmp/win/capturefile 6 1
```

Closing Notes

That's it! You should be set. You may wish to make a startup script to set up the variables automatically, but I would recommend just making some quick shell scripts for setting/unsetting the variables. I set the variable, run airodump, then unset the variable when I'm done, to be sure I do not interfere with anything else.

-NaHeMiA-