Aircrack-ng

patching

Tutorial: How To Patch Drivers

Version: 1.01 March 15, 2009
By: darkAudax

Introduction

People new to wireless security assessments often do not understand the need for patching drivers or how to do it. You sometimes need to patch the ieee80211 or mac80211 stack as well. This tutorial attempts to provide some background and some basic skills to the reader.

IMPORTANT The existence of this tutorial does mean we will provide basic linux support on the IRC channel or forum. Do not post basic linux or generic patching questions on the forum, they will be ignored. It is intended to assist people in being self-sufficient.

First off, what is a patch? All executable programs are generated from source code. This source code provides all the instructions to achieve the desired functionality the original author wanted. However, there are times that we want the programs to do something slightly differently for the purposes of wireless security assessments. An example being packet injection. So the source code is changed to achieve the new functions. In order to share this code, a “patch” is made with the differences between the original source code and the modified source code.

This “patch” can then be applied by anybody to their own copy of the original source code then compiled to obtain the new functionality. This makes it very transportable and maintainable.

Patches are easy to apply, once you understand a few simple concepts:

- Patches are usually for a specific version of the source code. This means old patches may not work with newer versions of the source code. Sometimes they do, sometimes they don't. As well, a patch may or may not work on older versions of the source code. All things being equal, use a patch specifically written for your version of the source code.
- Patches are generally built from 'clean' unpatched program sources. So, one patch may make a change that causes other patches to fail.
- Patches are not part of the released source code, thus do not be surprised it they don't work. Always keep a backup of your original program source!

Other common questions:

- What patches do I need?
- Where do I get the patch?

The aircrack-ng wiki [http://aircrack-ng.org/doku.php] typically indicates which patches are required for particular drivers. And the appropriate pages contain detailed installation and patching instructions. See the drivers page for links to the various detailed pages.

You can obtain the patch in a variety of ways. The wiki page normally provides a download link. As well, patches are included in the aircrack-ng source package in the “patches” directory. Many times patches are under development and you can find links to them on the Forum [http://forum.aircrack-ng.org/].

Applying a Patch

This section provides generic patching information. Wherever possible, follow the detailed instructions given on the wiki page for your driver. It is strongly recommended you backup your source prior to applying patches in case something goes wrong. This way you can easily recover the original source code.

This whole section is done via a console session.

The first step is to download the patch. The wiki page normally provides the instructions for this. Typically you use “wget” as in:

```
wget "URL to patch"

wget http://patches.aircrack-ng.org/rtl8187_2.6.24v3.patch
```

Then you need to move the patch to the appropriate directory using the “mv” command. The question arises as to which is the “appropriate directory”? There is no correct answer to the question. It depends on what you are patching and how the patch was created. Looking at the directories referenced in the patch itself usually gives you a good indication of where it should go. You might need to try a few locations. Here are some typical locations:

- Same directory as the file(s) to be patched
- One directory above the file(s) to be patched
- /usr/src/linux or similar when patching kernel modules

Once the patch is in place then change to the directory containing the patch with “cd”.

Now is time to run the “patch” command. The generic format is:

```
patch -Np0  -i <name of the patch file>
```

Where:

- -N means don't apply the patch if it has already been installed.
- -p0 means the number of directories to strip from the file names within the patch. You sometimes need to experiment with -p1, -p2, -p3, etc. to strip varying numbers of directories.

If you want to test applying the patch:

```
patch -Np0 --dry-run --verbose -i <name of the patch file>
```

NOTE: There is double dash in front of “dry-run” and “verbose”.

It is always a good idea to perform a test prior to applying it for real. This way you can avoid problems.

Once the patch is installed then you need to recompile the program. This is typically done via:

```
make
make install
```

If you are recompiling a kernel module then see this [links#compiling_kernels|wiki entry]] for instructions on compiling kernels and single modules.

To undo (reverse) a patch:

```
patch -Rp0  -i <name of the patch file>
```

Where:

- -R means reverse the patch if it has already been installed.
- -p0 means the number of directories to strip from the file names within the patch. You sometimes need to experiment with -p1, -p2, -p3, etc. to strip varying numbers of directories.

Troubleshooting Tips

Can't find file to patch at input line

You get an error message similar to the following:

```
can't find file to patch at input line 4
Perhaps you used the wrong -p or --strip option?
The text leading up to this was:
--------------------------
|diff -Naur rtl8187_linux_26.1010.0622.2006_orig/beta-8187/ieee80211_crypt.h rtl8187_linux_26.1010.0622.2006_rawtx/beta-8187/ieee80211_crypt.h
|--- rtl8187_linux_26.1010.0622.2006_orig/beta-8187/ieee80211_crypt.h   2006-06-05 22:58:02.000000000 -0400
|+++ rtl8187_linux_26.1010.0622.2006_rawtx/beta-8187/ieee80211_crypt.h  2008-08-12 13:11:32.000000000 -0400
--------------------------
File to patch:
```

There are a few possible solutions depending on the root cause of the problem:

- Make sure the file you are trying to patch really exists on your system. In the example above, verify that “ieee80211_crypt.h” really exists. If if does not, then install the source code which contains the file. As well, most times you need the kernel headers and/or kernel source installed. The version of the kernel headers/source MUST match the version of the kernel you are running. “uname -r” will show you which kernel version you are running. As well, Fedora requires the kernel-devel rpm to be installed.
- You have the patch located in the wrong directory. Re-read the section in the Introduction section above regarding where to place the patch. Sometimes you can get clues about where to place the patch by looking at the patch itself. The directory paths specified in the patch should give you an indication of where it should be placed.
- Play with the ”-pX” value. This allows you to strip directories off of the referenced files in the patch. Try -p0, -p1, -p2, etc.
- The version of the patch may be wrong for your kernel version. Check to ensure that the patch you are using is known to work properly against the kernel you are running.

Hunk #X FAILED at XXX

You get an error message similar to the following:

```
patching file drivers/net/wireless/iwlwifi/iwl-sta.c
Hunk #1 FAILED at 968.
1 out of 1 hunk FAILED -- saving rejects to file drivers/net/wireless/iwlwifi/iwl-sta.c.rej
```

```
patching file drivers/net/wireless/iwlwifi/iwl-tx.c
Hunk #1 FAILED at 783.
Hunk #2 FAILED at 805.
Hunk #3 FAILED at 819.
3 out of 3 hunks FAILED -- saving rejects to file drivers/net/wireless/iwlwifi/iwl-tx.c.rej
```

This means the patch does not match the program source code on your system in one or more places. As a result, the patch process has failed.

There are a few possible solutions depending on the root cause of the problem:

- In some rare cases, a few failures may still be ignored. You can try compiling the program and see if it works. Not a high probability but worth a try.
- The version of the patch may be wrong for your kernel version. Check to ensure that the patch you are using is known to work properly against the kernel you are running. Most likely, you need an older or new version of the patch.
- If all hunks fail: The patch may be whitespace-damaged. Try adding the -l option to the patch command line.
- Try applying the patch with the “fuzz” option: add ”-F3” to the patch command line. (The number specifies the maximal fuzz allowed - 3 is a value that works well.)
- If all else fails, you can try manually updating the source code by reviewing the patch and applying the changes by hand.

patching.txt · Last modified: 2009/06/01 18:21 by mister_x