Reimplementation of Carburizer

Louisa Bessad

22 juillet 2014

1. Abstract

Carburizer was created to insure input checking from the driver, a failure detection before corruption appears and a fix of device failures. In this way the reliability of systems can be improved. But it does not handle all issues and few platforms or drivers had been tested. The goal of this document is to reimplement some funtions of Carburizer with Coccinelle. In this way, we have to look for different types of code.

2. Looking for different types of code

First, we will look for loop without timeout. We will consider as a timeout a variable which is incremented or decremented in the condition of the loop or another allocation (such as $\ll=$ *or* $\gg=$).

3. Solutions

To find loop without timeout we will apply two patches in the Linux3.2.59 Kernel. There is a patch for each type of loop : for and while. The loop which match will be modified to avoid infinite looping. The inserted code is the same that Carburizer used. For instance this code :

will become :

4. Results

We have counted the number of each type of loop on the Linux3.2.59 Kernel. Then we have counted the loop found by our patch. During the execution of patches some files have been skipped by Coccinelle to avoid infinite time of execution. It appeared that whatever the computer used to execute the patches without the timeout option of Coccinelle, the execution could never end. For those files the analysis must be made by a programer. The following array shows the result of these executions :

	infinite loop	EXN loop	total loop
while			
for			