Waterloo Taint Analysis Tool (WAINT)
User's manual

by

Dipl.-Inf. Xavier Noumbissi Noundou
xavier.noumbis@gmail.com

Contents

1 Abstract

Businesses increasingly use software. This is even more relevant for companies relying on e-commerce. However, software is error-prone and contain several bugs. Security bugs are one of the major problems faced by companies today. In the worst case, security bugs enable unauthorized users to gain full control of an application.

My PhD thesis introduces the concept of tainted path and describes techniques and algorithms to compute them in any imperative programming language that uses pointers (C, C++, Java, etc.). I implemented these algorithms in WAINT.

WAINT computes *tainted paths* in a C program without running it. WAINT does not require the developer to annotate the program under analysis. WAINT implements a flow-sensitive, interprocedural and context-sensitive analysis that computes tainted paths in C programs at compile-time.

2 Installation Instructions

This section of the manual explains how to install WAINT on a Linux machine. We have not tested WAINT on a Windows machine, but the installation should follow similar steps.

2.1 Required Software

This section enumerates all software that you need to run WAINT.

a) The compiler infrastructure **LLVM,** `version 3.3` (`http://llvm.org`)

b) The precompiled LLVM's tool chain **clang+llvm,** `version 3.3` which include binaries like `clang`, `llvm-link`, etc.

c) The DSA pointer analysis **poolalloc** (`https://github.com/llvm-mirror/poolalloc.git`).

2.2 Environment Variables

Table 1 that shows all environment variables that you have to define and export in order to successfully run WAINT.

Environment variables	Description
WAINT_HOME	`waint` home folder (e.g.: /home/user/waint)
LLVM_HOME	`llvm` home folder (e.g.: /home/user/llvm)
LLVM_LIB	`llvm` compiled library folder (e.g.: $LLVM_HOME/build/Release+Asserts/lib)
LLVM_BIN	`llvm` compiled binaries (e.g.: $LLVM_HOME/build/Release+Asserts/bin)
POOLALLOC	`poolalloc` home folder (e.g.: /home/user/poolalloc)

Table 1: Table with all environment variables required to install and use WAINT

You declare and export an environment variable ENV_VAR by writing the following commands in your "`.bashrc`" file:

```
ENV_VAR = path_to_folder
export ENV_VAR
```

2.3 How to Configure "clang+llvm" for use with WAINT

2.4 How to Configure "LLVM" for use with WAINT

a) Create a folder 'build' in $LLVM_HOME.

b) Copy and customized the script `configure-llvm.sh` from `waint`'s 'script' folder into the newly created 'build' folder. `llvm`'s 'build' folder.

c) Create a symbolic link in the folder "$LLVM_HOME/lib/Analysis". You can achieve this by running: **"ln -s $WAINT_HOME $LLVM_HOME/lib/Analysis/waint"**.

d) Run the script `configure-llvm.sh`.

2.5 How to Configure "poolalloc" for use with WAINT

a) The sources of the DSA pointer analysis **poolalloc** can be gathered using the command **"git clone https://github.com/llvm-mirror/poolalloc.git"**.

b) After getting the sources of `poolalloc`, the user has to checkout the `git` version under commit '181c62f1d29ae9de660bad0a6593130d15803abc' using the command **"git checkout 181c62f1d29ae9de660bad0a6593130d15803abc"**.

c) Copy and customized the script `configure-poolalloc.sh` from `waint` 'script' folder into $POOLALLOC, and run it.

d) Then run **"make"**, and **"make install"**. Make sure to run **"make install"** as `root` (or `administrator` on a `Windows` system).

3 Usage