

# WAIN: WATERLOO TAINT ANALYSIS TOOL

## USER'S MANUAL

by

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

June 9, 2015

## Contents

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Installation Instructions</b>	<b>2</b>
2.1	Required Software . . . . .	2
2.2	Environment Variables . . . . .	2
2.3	How to Configure "clang+llvm" for use with WAIN . . . . .	3
2.4	How to Configure "LLVM" for use with WAIN . . . . .	3
2.5	How to Configure "poolalloc" for use with WAIN . . . . .	3
<b>3</b>	<b>Compiling and Running WAIN</b>	<b>3</b>
3.1	Folder Structure . . . . .	3
3.2	Important Bash Scripts . . . . .	4

## 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 WAIN.

WAIN computes *tainted paths* in a C program without running it. WAIN does not require the developer to annotate the program under analysis. WAIN 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 **WAIN**T on a Linux machine. We have not tested **WAIN**T 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 **WAIN**T.

- a) **WAIN**T: <https://github.com/xnoumbis/waint.git>
- b) The compiler infrastructure **LLVM**, version 3.3 (<http://llvm.org>)
- c) The precompiled LLVM's tool chain **clang+llvm**, version 3.3 which include binaries like clang, llvm-link, etc.
- d) 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 **WAIN**T.

Environment variables	Description
<b>WAIN_HOME</b>	<b>WAIN</b> T home folder (e.g.: /home/user/waint)
<b>LLVM_HOME</b>	llvm home folder (e.g.: /home/user/llvm)
<b>LLVM_LIB</b>	llvm compiled libraries folder (e.g.: <b>\$LLVM_HOME</b> /build/Release+Asserts/lib)
<b>LLVM_BIN</b>	llvm compiled binaries (e.g.: <b>\$LLVM_HOME</b> /build/Release+Asserts/bin)
<b>POOLALLOC</b>	poolalloc home folder (e.g.: /home/user/poolalloc)
<b>CLANGLLVMBIN</b>	clang+llvm binaries' folder (e.g.: /home/user/clang+llvm/bin)

Table 1: Table with all environment variables required to install and use **WAIN**T

You define 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 WAIT

- Download and unpack **clang+llvm**, version 3.3.
- Add the `bin` folder to your environment variable **PATH**. For instance by adding the following line in your file `".bashrc"`

```
PATH=$PATH:$CLANGLLVM_BIN
export PATH
```

## 2.4 How to Configure "LLVM" for use with WAIT

- Create a folder "build" in **\$LLVM\_HOME**.
- Copy and customized the script `configure-llvm.sh` from WAIT's "script" folder into the newly created "build" folder. `llvm`'s "build" folder.
- Create a symbolic link in the folder "**\$LLVM\_HOME/lib/Analysis**". You can achieve this by running: `ln -s $WAIT_HOME/src $LLVM_HOME/lib/Analysis/wait`.
- Run the script `configure-llvm.sh`.

## 2.5 How to Configure "poolalloc" for use with WAIT

- The sources of the DSA pointer analysis **poolalloc** can be gathered using the command `git clone https://github.com/llvm-mirror/poolalloc.git`.
- After getting the sources of **poolalloc**, the user has to checkout the `git` version under commit '181c62f1d29ae9de660bad0a6593130d15803abc' using the command `git checkout 181c62f1d29ae9de660bad0a6593130d15803abc`.
- Copy and customized the script `configure-poolalloc.sh` from WAIT's "script" folder into **\$POOLALLOC**, and run it.
- Then run `"make"`, and `"make install"`. Make sure to run `"make install"` as `root` (or administrator on a Windows system).

# 3 Compiling and Running WAIT

## 3.1 Folder Structure

The following folders constitute WAIT directory structure:

- "benchmarks": folder with sample scripts to run WAIT.
- "doc": folder with the manual.

- "scripts": folder with installation configuration files for poolalloc and LLVM.
- "src": folder with all C++ source files, and Bash scripts to compile and run **WAIN**T.
- "src/cfg": folder with all configuration files

## 3.2 Important Bash Scripts

Among others, **WAIN**T source folder contains the following two important Bash scripts:

- a) `ctainthelp.sh`: this script is used to generate and merge `llvm` intermediate representation (IR) files.
- b) `runOpt.sh`: this script is used to run the analysis of **WAIN**T on the program under analysis. We encourage users to look at the sample scripts in the folder "benchmarks" to learn how to use `runOpt.sh`.