\*\* DRAFT \*\*

SCA Wrapper Installation

HP Fortify

3 November 2016

Version 1.2

# Overview

The SCA Wrapper addresses the situation where a build system is making direct (including full path) compiler calls in its build, preventing SCA from intercepting those commands so they can be routed to the SCA app for vulnerability analysis.

The general architecture is as follows. The wrapper app replaces the compile command in place. The real compiler is moved or renamed. SCA Wrapper then receives the “compile” command, wraps it in the sourceanalyzer (SCA) command, and executes it. The executed command includes a call to the real compiler. The build command is executed as if it were a normal build effort, as the wrapper takes care of all SCA commands and interactions.

This has been tested on Windows only.

# Installation

## Installing the SCA Wrapper Application

1. Identify the compiler file used by the build system.
2. Rename it by appending “\_real” or similar to it (i.e. gcc\_real.exe) or alternatively move it to a different folder
3. Copy the scawrapper.exe app to the folder and rename it to the original compiler’s name
4. Place fortify.ini in same the folder where you will execute the build command

# Configuration

## Fortify SCA Configuration

1. Put the <sca install>\bin folder in the path (it should be something like C:\Program Files\HP\_Fortify\HP\_Fortify\_SCA\_and\_Apps\_4.10\bin)
2. If you changed the name of the real compiler above (step 2), then you will need to map the renamed compiler in fortify-sca.properties.
   1. Open <sca install>\Core\config\fortify-sca.properties in a text editor.
   2. Scroll to the compiler mappings and add a line mapping the new compiler (left side maps to compiler file name, right side to associated compiler standard) Example:

com.fortify.sca.compilers.gcc\_real = com.fortify.sca.util.compilers.GccCompiler

## Fortify.ini Configuration

The fortify.ini file should be placed in the directory where the build occurs. You will have to do a few changes to its configuration:

[INI]

version : Leave as is

debug: Set to 1 to turn on logging, 0 for off. Logs are simply written to std out (the command window)

[compiler]

name : Exact name and path of the “real” compiler

[sca]

prog : Leave as is, unless sourceanalyzer is not in the path. Then put full name and path here

buildid : Build ID of the app. This is typically the app name, sometimes with version appended

flags = extra sourceanalyzer flags can be placed here (i.e. adjust memory, debug/verbose logging, etc.)

# Generic Application Development Process Flow

The SCA Wrapper application only handles SCA translation, not the analysis stage. Use the build ID established above to handle analysis (-scan).

The SCA Wrapper application can only handle one compiler type at a time. So if your build system calls 2+ different compilers that need to be wrapped, as of now, they will have to be done separately. In this case they can be translated one at a time but analyzed/scanned all together.

# Known Issues

Spaces in the file/directory names will not work. Using double quotes was causing inconsistent behavior, so until I get that working properly, paths and file names will need to have no spaces in them. Note if sourceanalyzer and the compiler are in the PATH, there is typically no need to fully qualified directories.

The SCA Wrapper application only handles SCA translation, not the analysis stage.

# Updates

1. First version.

1.1 Added simple logging and debug. Added version flag. Fixed some issues with piecing together the compiler command.

1.2 Added multiple location attempts for fortify.ini. (Hopefully) made paths work for Windows and Linux.

# HPE Fortify Contact

Andy Earle

HP Fortify Solution Architect, South Central

512.815.2536

[andy.earle@hp.com](mailto:andy.earle@hp.com)