Software Assurance Tips

A product of the Software Assurance Tips Team[1]

Andrea Barnes

Monday 20th December, 2021

1 GCC as a Static Analysis Tool

Updated Wednesday 11th February, 2026

A new static analysis feature has been released with GCC-10 and GCC-11. David Malcom is a Redhat Developer on the GCC project who has implemented `-fanalyzer`: “A static analysis pass to identify various problems at compile-time, rather than at runtime.”[2] This analyzer writes security issues to the console using the `-Wanalyzer` tag (not to be confused with the GCC -W warning flags). The output includes Common Weakness Enumeration (CWE) identifiers as well the ability to print a path of events that trigger the flag.

1.1 History of GCC-10 and GCC-11 Static Analyzers

GCC-10 includes 15 `-fanalyzer` checkers:

- `-Wanalyzer-double-free`
- `-Wanalyzer-use-after-free`
- `-Wanalyzer-free-of-non-heap`
- `-Wanalyzer-malloc-leak`
- `-Wanalyzer-possible-null-argument`
- `-Wanalyzer-possible-null-dereference`
- `-Wanalyzer-null-argument`
- `-Wanalyzer-null-dereference`
- `-Wanalyzer-double-fclose`
- `-Wanalyzer-file-leak`
- `-Wanalyzer-stale-setjmp-buffer`
- `-Wanalyzer-use-of-pointer-in-stale-stack-frame`
- `-Wanalyzer-unsafe-call-within-signal-handler`
- `-Wanalyzer-tainted-array-index`
- `-Wanalyzer-exposure-through-output-file`

The checker works well on small to medium sized, C examples. An issue that Malcom noted includes bugs in the analyzer’s state-tracking component regarding symbolic values and canonicalization to compare different states. As these bugs were fixed, more bugs would be found. This prompted Malcom to rewrite the entire component for the release of GCC-11.

In the release of GCC-11, Malcom fixed the state-tracking component bugs by implementing the symbolic values as singletons, using pointers and reducing large amounts of canonicalization code. Other features that Malcom updated in this release include partial C++ support for `new` and `delete`, rewriting the memory leak detection to generate fewer false positives, and fixing non-determinism logic to ensure that the analyzer’s behavior would not vary from run to run.[3]

Additionally, GCC-11 adds 4 new `-fanalyzer` checkers:

- `-Wanalyzer-write-to-const`
- `-Wanalyzer-write-to-string-literal`
- `-Wanalyzer-shift-count-negative`

- `-Wanalyzer-shift-count-overflow`

The `-fanalyzer` flag can be directly added to the GCC command in terminal or to the CFLAGS variable for Makefiles. Ensure that GCC warnings (-W warnings) are not disabled. Disabling warnings will also cause the fanalyzer warnings to be suppressed. Malcolm is still developing the tool and we can look forward to new updates and a large rewrite when GCC-12 is released.

1.2 Installing GCC-11 on Ubuntu-based Distributions

This install is performed on a fresh install of Linux Mint 20.1 Cinnamon with some extra checks.

1.2.1 Update Apt Repository

```
$ sudo apt update && sudo apt upgrade
```

`update` will update the list of available packages and their versions while `upgrade` will install newer versions of the packages you already have.

1.2.2 Check if GCC-11 is Already Installable

```
$ apt-cache search gcc-11
```

`apt-cache` will display the gcc-11 packages if they are already installable from your current repositories. If you see them, you can skip down to Section 1.2.4 to install and configure.

1.2.3 Add Ubuntu Toolchain Repository

```
$ add-apt-repository -y ppa:ubuntu-toolchain-r/test
```

Newer versions of GCC and other development packages can be found in Ubuntu's Toolchain test branch PPA.

If you receive the error `gpg: keyserver receive failed: general error`, you can manually pull down the key from the error code with the following command:

```
$ sudo apt-key adv –keyserver keyserver.ubuntu.com –recv-key KEY_FROM_ERROR
```

Don't forget to run update again to update the list of available packages:

```
$ sudo apt update
```

1.2.4 Installing and Configuring GCC

```
$ sudo apt install gcc-11 && sudo apt install g++-11
```

This is the general install command for GCC (with G++ support). To run, you can use gcc-11 when compiling, or you can configure gcc to default to gcc-11 with the following command:

```
$ sudo update-alternatives –config gcc
```

This will allow you to make a selection based on the versions of gcc install. If you receive an error that there are no alternatives, you can use the following few commands to add them manually:

```
$ sudo update-alternatives –remove-all gcc #To clean out
$ ls /usr/bin/gcc* #To see what versions you have installed
$ sudo update-alternatives –install /usr/bin/gcc gcc /usr/bin/gcc-11 10
```

The setup for the last command is `–install <link> <name> <path> <priority>`. Repeat the last command for all versions of gcc listed.

References

[1] Jon Hood, ed. SwATips. https://www.SwATips.com/.

[2] David Malcolm. Static Analysis in GCC 10. RedHat Developer. Mar. 26, 2020. URL: https://developers.redhat.com/blog/2020/03/26/static-analysis-in-gcc-10.

[3] David Malcolm. Static Analysis Updates in GCC 11. RedHat Developer. Jan. 28, 2021. URL: https://developers.redhat.com/blog/2021/01/28/static-analysis-updates-in-gcc-11.