

The  GNOME™ Conference  
GUADEC

What's Coverity static analysis ever done for us?

Philip Withnall  
Endless Mobile

`philip@tecnocode.co.uk`

# What is static analysis?

Compile-time testing of all possible code paths.

# What is Coverity Scan?

- 🐾 Proprietary
- 🐾 Free to use for open source projects
- 🐾 A locally run tool and paired web service

# What is Coverity Scan?

The screenshot shows the Coverity Scan dashboard for a user named philip@tecnocode.co.uk. The dashboard displays analysis results for four projects: cairo, dbus, evolution-data-server, and flatpak. Each project entry includes an 'Admin' button, a status indicator (e.g., 'Not fully configured'), and a table of metrics: Last Analyzed, Defect Density, and Outstanding Defects. Links for 'Project Overview', 'Project Settings', and 'View Defects' are provided for each project. On the right side, there are sections for 'Travis-CI for GitHub projects and run your builds automatically' with 'Do-it-Yourself' and 'Travis CI' buttons, and 'Additional Resources' with links to 'Review the Quick Start Guide', 'Frequently Asked Questions', and 'Join the conversation'. A 'Questions?' section provides contact information: 'Contact us at: scan-admin@coverity.com'.

Project	Last Analyzed	Defect Density	Outstanding Defects
cairo	Jul 28, 2017	6.66	1,434
dbus	Jul 28, 2017	0.06	9
evolution-data-server	Mar 15, 2016	0.01	4
flatpak	Feb 27, 2017	0.33	45

# What is Coverity Scan?

The screenshot displays the Coverity Scan web interface. At the top, the browser address bar shows the URL: `https://scan7.coverity.com/reports.htm#v15948/p12688/fileInstanceId=30135214&defectInstanceId=6722638&mergedDefectId=1388526`. The page header includes navigation links like "Return to Dashboard", "Guided Tour", and "Help", along with a user profile for "phillip@tecnocode.co.uk".

The main content area features a table of issues. The table has columns for CID, Type, Impact, Status, First Detected, Owner, Classification, Severity, Action, Component, and Category. The following table represents the data shown in the screenshot:

CID	Type	Impact	Status	First Detected	Owner	Classification	Severity	Action	Component	Category
1388521	Bad bit shift operation	Medium	Triaged	12/12/16	muelli@cryptol	Bug	Moderate	Fix Submi	Other	Intege
1388526	Division or modulo by f	Medium	Triaged	12/12/16	phillip@tecnoc	Bug	Insignifican	Fix Submi	Other	Incorr
1388527	Division or modulo by f	Medium	Triaged	12/12/16	phillip@tecnoc	Bug	Insignifican	Fix Submi	Other	Incorr
1388541	Various	Medium	Triaged	12/12/16	phillip@tecnoc	Bug	Insignifican	Fix Submi	Other	Insecu
1388549	Various	Medium	Triaged	12/12/16	muelli@cryptol	Bug	Moderate	Fix Submi	Other	Insecu

Below the table, there is a "1 of 8 issues selected" indicator and a pagination control showing "Page 1 of 1".

The lower portion of the screenshot shows a code editor for the file `timescale.c`. The code includes comments and C code snippets. A red box highlights a specific issue:

```
24. divide_by_zero: In expression (double)dest_width / src_width, division by expression src_width which may be zero has undefined behavior.
```

The code snippet below the error shows the context of the division:

```
(double)dest_width / src_width,  
(double)dest_height / src_height,  
filter level. 255. 0. 0. 16.
```

On the right side of the interface, there is a "Triage" panel. It contains several dropdown menus for "Classification" (set to "Bug"), "Severity" (set to "Insignificant"), and "Action" (set to "Fix Submitted"). Below these are fields for "Ext. Reference" (set to `https://bugzilla.gnome.org/show_bug.cgi?id=77`) and "Owner" (set to `phillip@tecnocode.co.uk ()`). There is also a text area for "Enter comments" and two buttons: "Apply + Next" and "Apply".

At the bottom of the triage panel, there are sections for "Projects & Streams", "Detection History", "Triage History", and "Occurrences". The "Occurrences" section shows a dropdown for "gdk-pixbuf" and a list of "Events contributing to issue":

- 2 zero\_return timescale.c:129
- 3 assignment timescale.c:129
- 24 divide\_by\_zero timescale.c:234

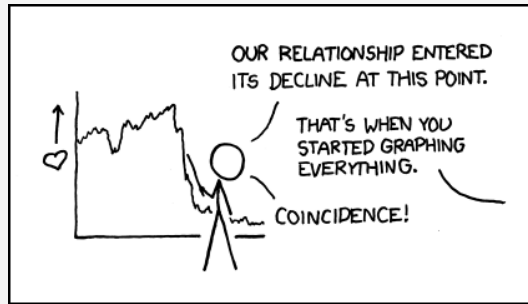
## Is it the best tool for the job?

- 👉 Mature support for triaging and dismissing false positives
- 👉 Wide use over many projects and active development
- 👉 Free to use
- 👉 Proprietary
- 👉 Submission rate limiting
- 👉 Should be used as one tool out of many

## How have we been using Coverity?

- 👉 Jenkins + JHBuild
- 👉 Manually created Jenkins jobs
- 👉 Limited set of hand-picked 'security critical' modules
- 👉 E-mail notification of scan results
- 👉 Partial ownership by module maintainers
- 👉 No real comaintainership of the project

## What impact has this had?



Randall Munroe, <https://xkcd.com/523/>, CC-BY-NC 2.5



## How is this useful?

- 🐾 Find bugs in error paths
- 🐾 Complements unit testing
- 🐾 Find bugs in parsers and file loaders
- 🐾 Find bugs before they are hit at runtime
- 🐾 Jenkins won't forget to run analyses like maintainers do

## How is this not useful?

- 👉 Not reasonable to use as a try-server
- 👉 Initial dump of false positives when adding a project
- 👉 Problems with handling idiomatic C
- 👉 Jenkins + JHBuild is not the most reliable

## How do I get involved?

- 👉 Talk to me; propose modules for inclusion into Jenkins
- 👉 Or go with Coverity yourself
- 👉 Or try other static analysis tools (`clang-analyzer?`) and let me know!

## Miscellany

[Jenkins jobs](https://jenkins.freedesktop.org/view/GNOME%20Coverity/) `https://jenkins.freedesktop.org/view/GNOME%20Coverity/`

[Coverity](http://scan.coverity.com/) `http://scan.coverity.com/`

[Wikipedia on static analysis](https://en.wikipedia.org/wiki/Static_program_analysis)

`https://en.wikipedia.org/wiki/Static_program_analysis`



*Creative Commons Attribution-ShareAlike 4.0 International License*

Beamer theme: `https://git.gnome.org/browse/presentation-templates/tree/GUADEC/2017`