

The static check needle in the warnings haystack

ELCE 2019
Lyon, France

Frank Rowand, Sony

October 30, 2019
191026_1321

Examples of Static Analysis Tools

- gcc
- dtc
- smatch
- sparse

What is the problem?

Premise

There many warnings reported by the static analysis tools against existing code

It can be difficult to determine what new warnings a code change has created when so many warnings already exist

Thus many developers and maintainers do not check for new warnings

Premise

There many warnings reported by the static analysis tools against existing code

It can be difficult to determine what new warnings a code change has created

Thus many developers and maintainers do not check for new warnings

True or False?

Look at an example

```
$ git checkout v5.4-rc3
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=3 drivers/of/overlay.o >/dev/null 2>&1
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=3 drivers/of/overlay.o 1>/dev/null 2>old_warnings
```

```
$ git checkout 5.4-rc3_test_v3
$ git log -n2 --oneline
d85d7811732f of: drivers/of/overlay.c (v3) add new_func()
4f5cafb5cb84 Linux 5.4-rc3
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=3 drivers/of/overlay.o >/dev/null 2>&1
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=3 drivers/of/overlay.o 1>/dev/null 2>new_warnings
```

(1) Many warnings?

```
$ wc -l old_warnings; wc -l new_warnings  
1318 old_warnings  
1320 new_warnings
```

Yes

(2) Hard to find new warnings?

```
$ wc -l old_warnings; wc -l new_warnings  
1318 old_warnings  
1320 new_warnings
```

```
$ diff -u old_warnings new_warnings | wc -l  
127
```

Naive analysis suggests two new lines of warning messages

But diff of old vs new warnings show 127 lines differ

(2) Hard to find new warnings?

```
$ wc -l old_warnings; wc -l new_warnings  
1324 old_warnings  
1326 new_warnings
```

```
$ diff -u old_warnings new_warnings | wc -l  
128
```

Yes

The problem commit

```
$ git diff 4f5cafb5cb84 d85d7811732f
diff --git a/drivers/of/overlay.c b/drivers/of/overlay.c
index c423e94baf0f..435f2272306a 100644
--- a/drivers/of/overlay.c
+++ b/drivers/of/overlay.c
@@ -23,6 +23,9 @@
```

```
#include "of_private.h"
```

```
+void new_func(phandle phandle) { }
```

```
+EXPORT_SYMBOL(new_func);
```

```
+
```

```
/**
```

```
 * struct target - info about current target node as recursing through overlay
```

```
 * @np: node where current level of overlay will be appl
```

(3) Don't check for new warnings?

Who in this room ever checks their submissions for new warnings?

What percentage of submissions?

Counter example

A different simple commit

Commit that adds a warning

(warning message reformatted for readability)

```
$ git checkout resolver_new_warnings
$ git log -n2 --oneline
8a55f0697865 of: drivers/of/resolver.c new warnings for make W=1, W=2,
4f5cafb5cb84 Linux 5.4-rc3
```

```
$ rm drivers/of/resolver.o; make W=1 drivers/of/resolver.o
...
CC      drivers/of/resolver.o
```

```
drivers/of/resolver.c:350:6: warning: no previous prototype
for 'htsc_ex_02_p_3_new_func'
[-Wmissing-prototypes]
```

New warning is obvious, right?

Many subsystems build with no errors for
“make W=1”

Thus it is easy to see newly created warnings
for W=1

New warning is obvious, right?

(build in source tree)

```
$ env | grep KBUILD_OUTPUT
```

```
$ rm drivers/of/resolver.o; \  
make W=1 drivers/of/resolver.o
```

```
...  
CC      drivers/of/resolver.o  
drivers/of/resolver.c:350:6: warning: no previous prototype for 'ht  
sc_ex_02_p_3_new_func' [-Wmissing-prototypes]
```

New warning is obvious, right?

(build external to source tree)

```
$ env | grep KBUILD_OUTPUT
```

```
KBUILD_OUTPUT=../build/dragon_linus_5.4-rc
```

```
$ rm ${KBUILD_OUTPUT}/drivers/of/resolver.o; \  
make W=1 drivers/of/resolver.o
```

```
...
```

```
CC      drivers/of/resolver.o
```

```
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/drivers/of/reso  
lver.c:350:6: warning: no previous prototype for 'htsc_ex_02_p_3_ne  
w_func' [-Wmissing-prototypes]
```

build internal vs. external

(build in source tree)

```
CC      drivers/of/resolver.o
drivers/of/resolver.c:350:6: warning: no previous prototype for 'htsc_ex_02_p_3_new_func' [-Wmissing-prototypes]
```

(build external to source tree)

```
CC      drivers/of/resolver.o
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/drivers/of/resolver.c:350:6: warning: no previous prototype for 'htsc_ex_02_p_3_new_func' [-Wmissing-prototypes]
```

So what?

You are probably thinking:

If Frank is complaining about that, he isn't smart enough to be a kernel developer.

He probably isn't even smart enough to tie his own shoes.

So what?

There are many sources of noise that distract from being able to focus on the messages from the static analysis tools

- Some noise is minor

- Some noise is major

But when all sources of noise add up, the result is that it can be hard to isolate new warnings

Some solutions seem easy

original:

```
$ rm ${KBUILD_OUTPUT}/drivers/of/resolver.o;          \  
> make W=1 drivers/of/resolver.o  
make[1]: Entering directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'  
  GEN      Makefile  
  CALL     /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/checksyscalls.sh  
  CALL     /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/atomic/check-atomics.sh  
  CC       drivers/of/resolver.o  
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/drivers/of/resolver.c:350:6: warning: no previous prototype for 'htsc_ex_  
make[1]: Leaving directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'
```

remove noise:

```
$ rm ${KBUILD_OUTPUT}/drivers/of/resolver.o;          \  
> make W=1 drivers/of/resolver.o 2>&1 1>/dev/null \  
> | sed "s|\`get_CURDIR\`/||g"  
drivers/of/resolver.c:350:6: warning: no previous prototype for 'htsc_ex_02_p_3_new_func' [-Wmissing-prototypes]
```

Some solutions seem easy

remove noise:

```
$ rm ${KBUILD_OUTPUT}/drivers/of/resolver.o; \
> make W=1 drivers/of/resolver.o 2>&1 1>/dev/null \
> | sed "s|\`get_CURDIR`/||g"
drivers/of/resolver.c:350:6: warning: no previous prototype for 'htsc_
```

Why ``get_CURDIR`` instead of ``pwd``?

`pwd` usually works, but there are some corner cases where `make $(CURDIR) != `pwd``

My `get_CURDIR` script invokes `make` to get `$(CURDIR)`

Examples of noise

Changing the value of “W=” can result in other files being rebuilt

Compiling those other files may generate warnings

Examples of noise - other files

```
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=1 drivers/of/overlay.o
make[1]: Entering directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'
  GEN      Makefile
  YACC     scripts/genksyms/parse.tab.[ch]
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/genksyms/parse.y: warning: 9 shift/reduce conflicts [-Wconflicts-
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/genksyms/parse.y: warning: 5 reduce/reduce conflicts [-Wconflicts
  HOSTCC  scripts/genksyms/parse.tab.o
  HOSTCC  scripts/genksyms/lex.lex.o
  HOSTLD  scripts/genksyms/genksyms
  CC      scripts/mod/empty.o
  MKELF   scripts/mod/elfconfig.h
  HOSTCC  scripts/mod/modpost.o
  CC      scripts/mod/devicetable-offsets.s
  HOSTCC  scripts/mod/file2alias.o
  HOSTCC  scripts/mod/sumversion.o
  HOSTLD  scripts/mod/modpost
  CC      kernel/bounds.s
  CC      arch/arm/kernel/asm-offsets.s
  CALL    /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/checksyscalls.sh
  CALL    /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/atomic/check-atomics.sh
  CC      drivers/of/overlay.o
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/drivers/of/overlay.c:26:6: warning: no previous prototype for 'new_func'
make[1]: Leaving directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'
$ rm ${KBUILD_OUTPUT}/drivers/of/overlay.o
$ make W=1 drivers/of/overlay.o
make[1]: Entering directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'
  GEN      Makefile
  CALL    /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/checksyscalls.sh
  CALL    /local/frowand_nobackup/src/git_linus/linux--5.4-rc/scripts/atomic/check-atomics.sh
  CC      drivers/of/overlay.o
/local/frowand_nobackup/src/git_linus/linux--5.4-rc/drivers/of/overlay.c:26:6: warning: no previous prototype for 'new_func'
make[1]: Leaving directory `/local/frowand_nobackup/src/git_linus/build/dragon_linus_5.4-rc'
```

Examples of noise

gcc warnings include file line number

```
-drivers/of/overlay.c:776:2: warning: conversion to 'size_t' from 'int' may change the sign of the operand [-Wconversion]
-drivers/of/overlay.c:835:2: warning: conversion to 'long unsigned int' from 'int' may change the sign of the operand [-Wconversion]
+drivers/of/overlay.c:779:2: warning: conversion to 'size_t' from 'int' may change the sign of the operand [-Wconversion]
+drivers/of/overlay.c:838:2: warning: conversion to 'long unsigned int' from 'int' may change the sign of the operand [-Wconversion]
```

Examples of noise

gcc warnings include file line number

```
-drivers/of/overlay.c:776:2: warning: conversion to 'size_t' from 'int' may change the sign of the operand [-Wconversion]
-drivers/of/overlay.c:835:2: warning: conversion to 'long unsigned int' from 'int' may change the sign of the operand [-Wconversion]
+drivers/of/overlay.c:779:2: warning: conversion to 'size_t' from 'int' may change the sign of the operand [-Wconversion]
+drivers/of/overlay.c:838:2: warning: conversion to 'long unsigned int' from 'int' may change the sign of the operand [-Wconversion]
```

Line numbers can be filtered out before diff of old and new warnings

Resulting in another annoyance:

The warning message in the diff does not show the line number, so it is hard to find the offending line

Examples of noise

Some macro expansions incorporate line number

The incorporated line number may be visible in a warning message

```
-drivers/of/overlay.c:1139:2: warning: pointer of type 'void *' used in arithmetic [-Wpointer-arith]
-drivers/of/overlay.c:1139:1: warning: redundant redeclaration of '__compiletime_assert_1139' [-Werror=macro-redefined]
-drivers/of/overlay.c:1139:1: note: previous declaration of '__compiletime_assert_1139' was here
-drivers/of/overlay.c:1139:2: warning: pointer of type 'void *' used in arithmetic [-Wpointer-arith]
+drivers/of/overlay.c:1142:2: warning: pointer of type 'void *' used in arithmetic [-Wpointer-arith]
+drivers/of/overlay.c:1142:1: warning: redundant redeclaration of '__compiletime_assert_1142' [-Werror=macro-redefined]
+drivers/of/overlay.c:1142:1: note: previous declaration of '__compiletime_assert_1142' was here
+drivers/of/overlay.c:1142:2: warning: pointer of type 'void *' used in arithmetic [-Wpointer-arith]
```

Solution: proof of concept tool

A set of scripts and programs, controlled by script “check_warn”

Revisit initial example

```
$ git diff 4f5cafb5cb84 d85d7811732f
diff --git a/drivers/of/overlay.c b/drivers/of/overlay.c
index c423e94baf0f..435f2272306a 100644
--- a/drivers/of/overlay.c
+++ b/drivers/of/overlay.c
@@ -23,6 +23,9 @@
```

```
#include "of_private.h"
```

```
+void new_func(phandle phandle) { }
```

```
+EXPORT_SYMBOL(new_func);
```

```
+
```

```
/**
```

```
* struct target - info about current target node as recursing through overlay
```

```
* @np: node where current level of overlay will be appl
```

Revisit initial example

```
$ git checkout v5.4-rc3
```

```
$$ warn_check -v --rf2 -s drivers/of/overlay.c
```

```
$ git checkout 5.4-rc3_test_v3
```

warn_check - lines of warnings

===== Number of lines in warning reports

===== W=0

OLD: 0

NEW: 0

===== W=1

OLD: 2

NEW: 3

===== W=2

OLD: 606

NEW: 609

===== W=3

OLD: 3295

NEW: 3297

warn_check - warnings count

=====
Count of each type of warning

=====
W=3

old	new	
-----	-----	
8	8	-Wattributes
29	29	-Wbad-function-cast
141	141	-Wcast-qual
146	146	-Wconversion
4	4	-Wpacked
277	277	-Wpadded
78	78	-Wpointer-arith
32	33	* -Wredundant-decls
481	481	-Wsign-conversion
3	3	-Wswitch-default

warn_check - warnings count

==== W=1

old	new	
1	1	-Wconflicts-rr
1	1	-Wconflicts-sr
0	1	* -Wmissing-prototypes

==== W=2

old	new	
12	12	-Wcast-align
4	4	-Wmissing-field-initializers
95	95	-Wnested-externs
14	16	* -Wshadow
18	18	-Wsign-compare
6	6	-Wunused-macros

==== W=3

old	new	
8	8	-Wattributes
29	29	-Wbad-function-cast
141	141	-Wcast-qual
146	146	-Wconversion
4	4	-Wpacked
277	277	-Wpadded
78	78	-Wpointer-arith
32	33	* -Wredundant-decls
481	481	-Wsign-conversion
3	3	-Wswitch-default

warn_check - warnings diff

```
##### filtered #####
```

```
@@ -3227,6 +3227,8 @@
```

```
filtered max 10
```

```
drivers/of/of_private.h::5: note: previous declaration of '__of_attach_node_sysfs' was here
drivers/of/of_private.h::13: warning: redundant redeclaration of '__of_detach_node_sysfs' [-Wredundant-decls]
drivers/of/of_private.h::6: note: previous declaration of '__of_detach_node_sysfs' was here
+drivers/of/overlay.c::1: warning: redundant redeclaration of 'new_func' [-Wredundant-decls]
+drivers/of/overlay.c::6: note: previous definition of 'new_func' was here
drivers/of/overlay.c::1: warning: padding struct size to alignment boundary [-Wpadded]
drivers/of/overlay.c::22: warning: padding struct to align 'cset' [-Wpadded]
drivers/of/overlay.c::1: warning: redundant redeclaration of 'of_overlay_notifier_register' [-Wredundant-decls]
```

```
<<<<< old <<<<<
```

```
drivers/of/of_private.h:66:5: note: previous declaration of '__of_attach_node_sysfs' was here
drivers/of/of_private.h:144:13: warning: redundant redeclaration of '__of_detach_node_sysfs' [-Wredundant-decls]
drivers/of/of_private.h:67:6: note: previous declaration of '__of_detach_node_sysfs' was here
drivers/of/overlay.c:44:1: warning: padding struct size to alignment boundary [-Wpadded]
drivers/of/overlay.c:75:22: warning: padding struct to align 'cset' [-Wpadded]
drivers/of/overlay.c:140:1: warning: redundant redeclaration of 'of_overlay_notifier_register' [-Wredundant-decls]
```

```
>>>>> new >>>>>
```

```
drivers/of/of_private.h:66:5: note: previous declaration of '__of_attach_node_sysfs' was here
drivers/of/of_private.h:144:13: warning: redundant redeclaration of '__of_detach_node_sysfs' [-Wredundant-decls]
drivers/of/of_private.h:67:6: note: previous declaration of '__of_detach_node_sysfs' was here
drivers/of/overlay.c:27:1: warning: redundant redeclaration of 'new_func' [-Wredundant-decls]
drivers/of/overlay.c:26:6: note: previous definition of 'new_func' was here
drivers/of/overlay.c:47:1: warning: padding struct size to alignment boundary [-Wpadded]
drivers/of/overlay.c:78:22: warning: padding struct to align 'cset' [-Wpadded]
drivers/of/overlay.c:143:1: warning: redundant redeclaration of 'of_overlay_notifier_register' [-Wredundant-decls]
```

Where is warn_check?

Long term:

There is active discussion of whether to provide a location in the Linux kernel source tree for maintainer related tools

I would like to submit warn_check to that location

Where is it?

Short term:

github

warn_check has some rough edges, but it is useful in it's current form

While it is not in the Linux kernel source tree, while I am making it suitable for submission, I will mirror it on github at:

https://github.com/frowand/warn_check

Todo

Polish the code (smooth off the rough edges)

Supports .c, .s, .dts. Add support for:

- sparse
- smatch

Add multiple file support

Add option to specify `commit_old`, `commit_new`, check out the commits, optionally select all files in `commit_old..commit_new`

Submit to kernel.org

Call for help

I have found some of the causes of false detection of a new warning in a diff report through experience. I expect more such patterns to exist.

If you encounter other patterns please report them to me so that I can improve `warn_check`.

THE END

Thank you for your attention...

Questions?

How to get a copy of the slides

- 1) leave a business card with me
- 2) frank.rowand@sony.com
- 3) https://elinux.org/ELC_Presentations
- 4) <http://events.linuxfoundation.org>