

# Finding Bugs in Open Source Systems Code using Coccinelle

Julia Lawall (University of Copenhagen)

Joint work with  
Gilles Muller, René Rydhof Hansen, Jesper Andersen,  
Nicolas Palix  
DIKU-AU-INRIA

October 24, 2009

Bugs: They're everywhere!



# Our focus

## Bugs in Linux

- ▶ Linux is critical software.
  - Used in embedded systems, desktops, servers, etc.
- ▶ Linux is very large.
  - Over 12 000 .c files
  - Over 7 million lines of code
  - Increase of almost 50% since 2006.
- ▶ Linux has both more and less experienced developers.
  - Maintainers, contributors, developers of proprietary drivers

## Bug: !x&y

Author: Al Viro <viro@ZenIV.linux.org.uk>

wmi: (!x & y) strikes again

```
diff --git a/drivers/acpi/wmi.c b/drivers/acpi/wmi.c
```

```
@@ -247,7 +247,7 @@
```

```
    block = &wblock->gblock;
```

```
    handle = wblock->handle;
```

```
- if (!block->flags & ACPI_WMI_METHOD)
```

```
+ if (!(block->flags & ACPI_WMI_METHOD))
```

```
    return AE_BAD_DATA;
```

```
if (block->instance_count < instance)
```

## Bug: dereference of a possibly NULL value

**Author:** Mariusz Kozlowski <m.kozlowski@tuxland.pl>

tun/tap: Fix crashes if open() /dev/net/tun and then poll() it.

```
diff --git a/drivers/net/tun.c b/drivers/net/tun.c
@@ -486,12 +486,14 @@
- struct sock *sk = tun->sk;
+ struct sock *sk;
  unsigned int mask = 0;

  if (!tun)
    return POLLERR;

+ sk = tun->sk;
```

## Bug: Bad testing of ERR\_PTR values

Author: Julien Brunel <brunel@diku.dk>

```
[S390] drivers/s390: Use an IS_ERR test rather
than a NULL test
```

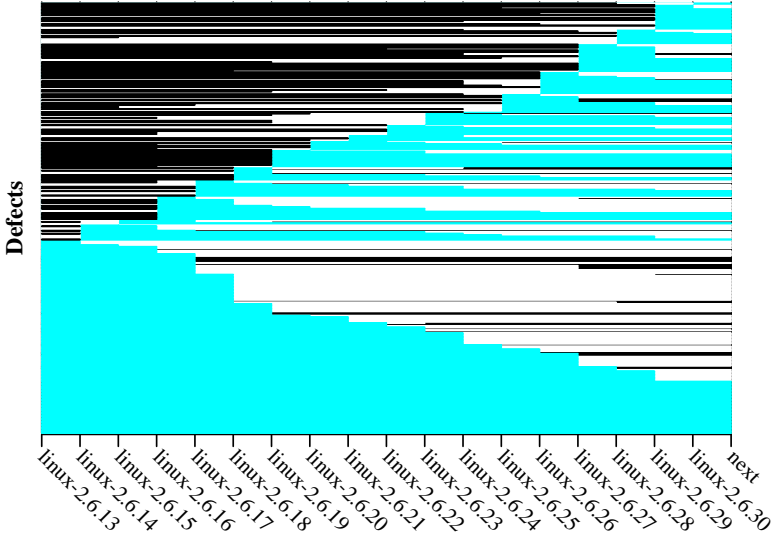
```
diff --git a/drivers/s390/char/tape_char.c
        b/drivers/s390/char/tape_char.c
@@ -109,7 +109,7 @@
```

```
    /* The current idal buffer is not correct. Allocate a
    new one. */
    new = idal_buffer_alloc(block_size, 0);
-   if (new == NULL)
+   if (IS_ERR(new))
        return -ENOMEM;

    if (device->char_data.idal_buf != NULL)
```



# Lifetime of NULL pointer dereference bugs



# Goal: Find and fix bugs in C code

Approach: Coccinelle: <http://coccinelle.lip6.fr/>

- ▶ Static analysis to find patterns in source code.
- ▶ Automatic transformation to fix bugs.
- ▶ User configurable, based on patch notation (**semantic patches**).

## Examples

- ▶ `!x&y`: mix of boolean and bit operations
- ▶ Dereference of values that might be NULL.
- ▶ Inconsistent error checking.

## !x&y issues

“x” and “y” are arbitrary expressions.

Normally “y” is a constant.

```
!file->f_flags & O_NONBLOCK
```

If y is a ! expression, the term might be valid:

```
if (!ixgb_clean_rx_irq(adapter) &  
    !ixgb_clean_tx_irq(adapter))  
    break;
```

The pattern might extend over multiple lines:

```
if (!state->card->  
    ac97_status & CENTER_LFE_ON)  
    val &= ~DSP_BIND_CENTER_LFE;
```

# Finding and fixing !x&y bugs using Coccinelle

```
@@  
expression E;  
constant C;  
@@
```

- !E & C

+ !(E & C)

- ▶ E is an arbitrary expression.
- ▶ C is an arbitrary constant.

# Example

## Semantic patch:

```
@@ expression E; constant C; @@  
- !E & C  
+ !(E & C)
```

## Original code:

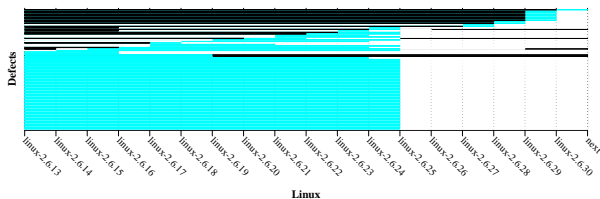
```
if (!state->card->  
    ac97_status & CENTER_LFE_ON)  
    val &= ~DSP_BIND_CENTER_LFE;
```

## Generated code:

```
if (!(state->card->ac97_status & CENTER_LFE_ON))  
    val &= ~DSP_BIND_CENTER_LFE;
```

# Results

- ▶ 96 instances in Linux from 2.6.13 (August 2005) to v2.6.28 (December 2008)
- ▶ 58 in 2.6.20 (February 2007)
- ▶ 2 in Linux-next (October 10, 2009)



# NULL pointer dereference issues

**Bug pattern:** Dereference before a NULL test:

@@

```
expression x;  
identifier fld;
```

@@

```
* x->fld
```

```
...
```

```
* x == NULL
```

- ▶ Either `x->fld` can crash, or `x == NULL` is unnecessary.
- ▶ Based on a Coverity rule.
- ▶ Isomorphisms cause `x == NULL` to also match eg `!x`.

## False positives

- ▶ **x can be modified between `x->fld` and `x == NULL`.**

```
bridge = bridge->bus->self;
if (!bridge || prev_bridge == bridge) ...
```

- ▶ **`x->fld` can be protected by another `NULL` test.**

```
if (sp && sp->user) { ... }
pvr2_channel_disclaim_stream(cp);
if (!sp) break;
```

- ▶ **There can be another execution path to `x == NULL`.**

```
if (!f->inocache && ino == 1) {
    f->inocache = jffs2_alloc_inode_cache();
    if (!f->inocache) { ... }
    ...
}
if (!f->inocache) { ... }
```

## Discarding modifications to x

```
@@
expression x,E;
identifier fld;
@@
(
  x = E
|
  * x->fld
  ... when != x = E
  when != &x
* x == NULL
)
```

## Discarding references protected via another NULL test

```
@s@
expression x,E;
identifier fld;
@@
(
  x = E
|
  x == NULL || (<+... x->fld ...+)
|
  x != NULL && (<+... x->fld ...+)
|
* x->fld
  ... when != x = E
    when != &x
* x == NULL
)
```

## Discarding cases with another path to the NULL test

```
@match@ expression x,E; identifier fld; position p1,p2; @@  
( x = E  
| x == NULL || (<+... x->fld ...+>)  
| x != NULL && (<+... x->fld ...+>)  
|  
  x@p1->fld  
  ... when != ( x = E | &x )  
  x@p2 == NULL  
)
```

```
@other_match exists@  
expression match.x, E1, E2; position match.p1,match.p2;  
@@
```

```
( x = E1 | &x )  
... when != ( x = E2 | &x | x@p1 )  
x@p2
```

```
@ script:python depends on !other_match && !other_match1@  
p1 << match.p1; p2 << match.p2;  
@@  
cocci.print_main("null ref",p1)  
cocci.print_secs("null text",p2)
```

# Results

159 reports for Linux 2.6.30

- ▶ 144 real bugs
- ▶ Lots of work to look through them!

Some have a common structure:

```
struct sock *sk = tun->sk;  
unsigned int mask = 0;  
  
if (!tun) return POLLERR;
```

Make a rule for these cases, and get them out of the way.

- ▶ Find and fix in this case.

## A more constrained rule

@@

```
type T;  
identifier i, fld;  
expression E;  
statement S;
```

@@

- T i = E->fld;

+ T i;

... when != E

when != i

if (E == NULL) S

+ i = E->fld;

## Example

### Semantic patch:

```
@@ type T; identifier i, fld;
expression E; statement S; @@
- T i = E->fld;
+ T i;
  ... when != ( E | i )
  if (E == NULL) S
+ i = E->fld;
```

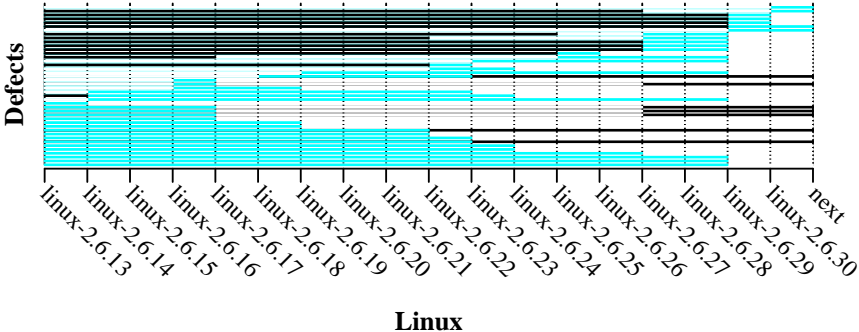
### Original code:

```
struct sock *sk = tun->sk;
unsigned int mask = 0;
if (!tun) return POLLERR;
```

### Generated code:

```
struct sock *sk;
unsigned int mask = 0;
if (!tun) return POLLERR;
sk = tun->sk;
```

# Results for the more constrained rule



# ERR\_PTR testing issues

## Example:

### ► Definition:

```
static inline struct idal_buffer *
idal_buffer_alloc(size_t size, int page_order) {
    ...
    ib = kmalloc(sizeof(struct idal_buffer) + ...,
                 GFP_DMA | GFP_KERNEL);
    if (ib == NULL)
        return ERR_PTR(-ENOMEM);
    ...
}
```

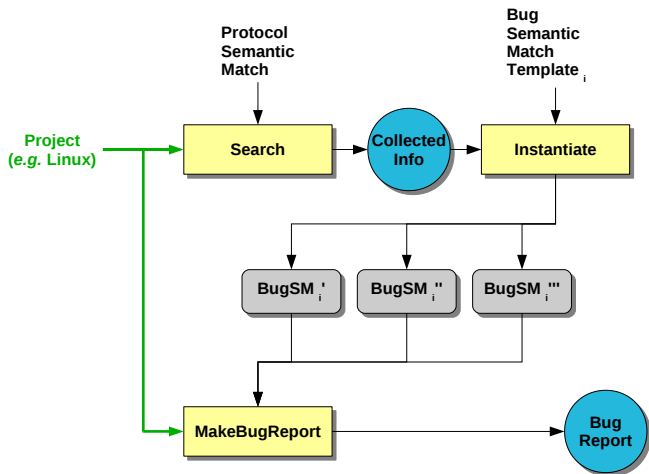
### ► Use:

```
new = idal_buffer_alloc(block_size, 0);
if (new == NULL) ...
```

## Observations:

- Function names not a priori known.
- Function definition and use might be in different files.

# Approach



## Finding functions that return ERR\_PTR

```
@direct exists@
identifier fn;
@@

fn(...) {
    ...
    return (ERR_PTR(...));
}

@script:python@
fn << direct.fn;
@@

print "category1: FN:%s" % fn
```

### Another rule for

```
ret = ERR_PTR(...)    ...    return ret;
```

## Finding call sites that don't test for ERR\_PTR

```
@r@  
expression x,E;  
position p;  
@@  
x@p = FN(...)  
... when != x = E  
( return x; | IS_ERR(x) )
```

```
@s@  
expression x;  
position p;  
@@  
IS_ERR(x@p = FN(...))
```

```
@@  
expression x;  
position p!=r.p,s.p;  
@@  
*x@p = FN(...)
```

## Results for an extended version [DSN09]

### Protocol finding results:

	classified	false positives
NULL only	1640	9
ERR_PTR only	478	1
NULL or ERR_PTR	112	9
Pointer only	623	5
unknown	7123	N/A

### Bug finding: inappropriate tests

	reported sites	bugs	false positives
NULL only	2	2	0
ERR_PTR only	26	19	7
Pointer only	44	23	21

### Bug finding: insufficient tests

	reported sites	bugs	false positives
NULL only	201	139	62
ERR_PTR only	21	17	4
NULL or ERR_PTR	11	5	6

# Conclusion

A patch-like program matching and transformation language

Over 350 patches created using Coccinelle accepted into Linux

Starting to be used by other Linux developers

Probable bugs found in gcc, postgresql, vim, amsn, pidgin, mplayer, openssl, vlc, wine

Coccinelle Users Day in Paris, November 25

<http://coccinelle.lip6.fr/>

**Kill bugs before they hatch!!!**



**COCCINELLE**

<http://coccinelle.lip6.fr/>