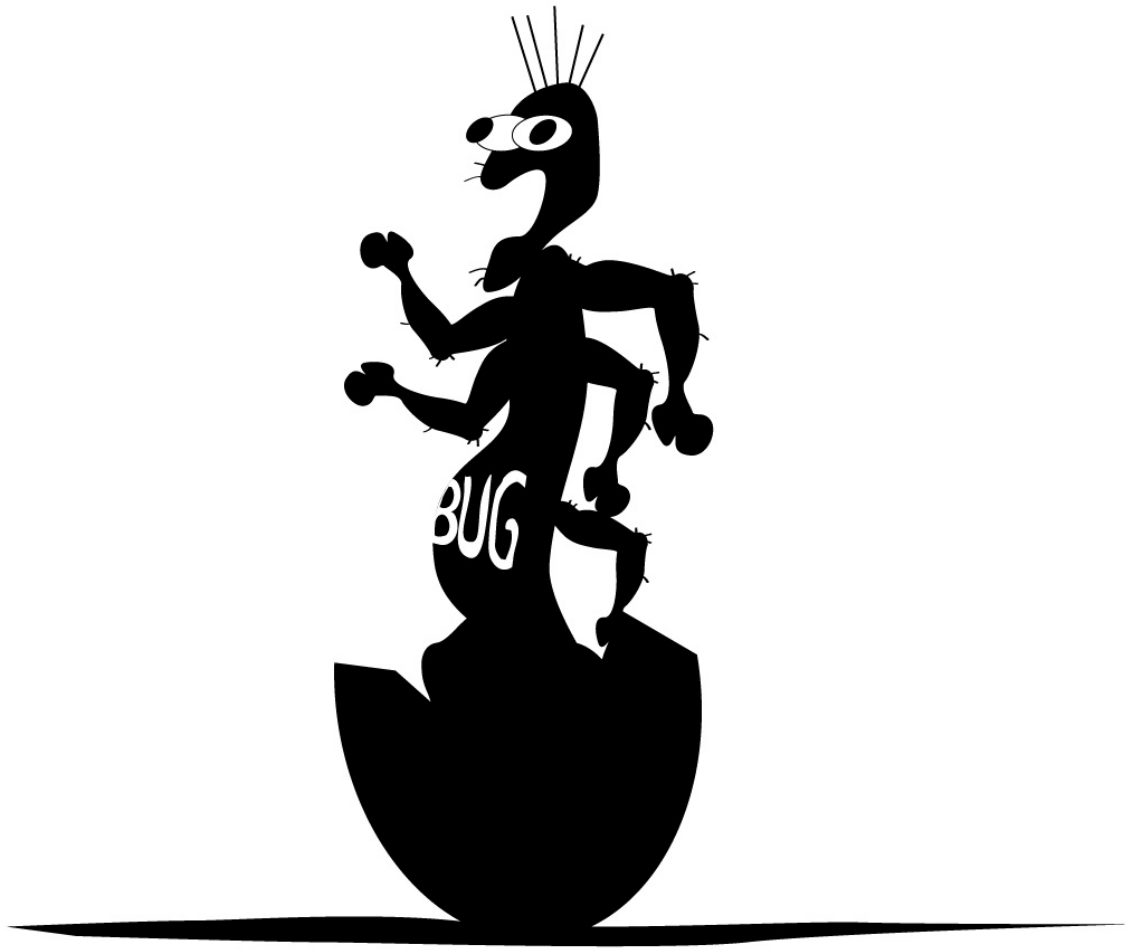
Finding Bugs in Open Source Systems Code using Coccinelle

Julia Lawall (University of Copenhagen/INRIA-Regal)

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

February 7, 2010

Bugs: They’re everywhere!



Our focus

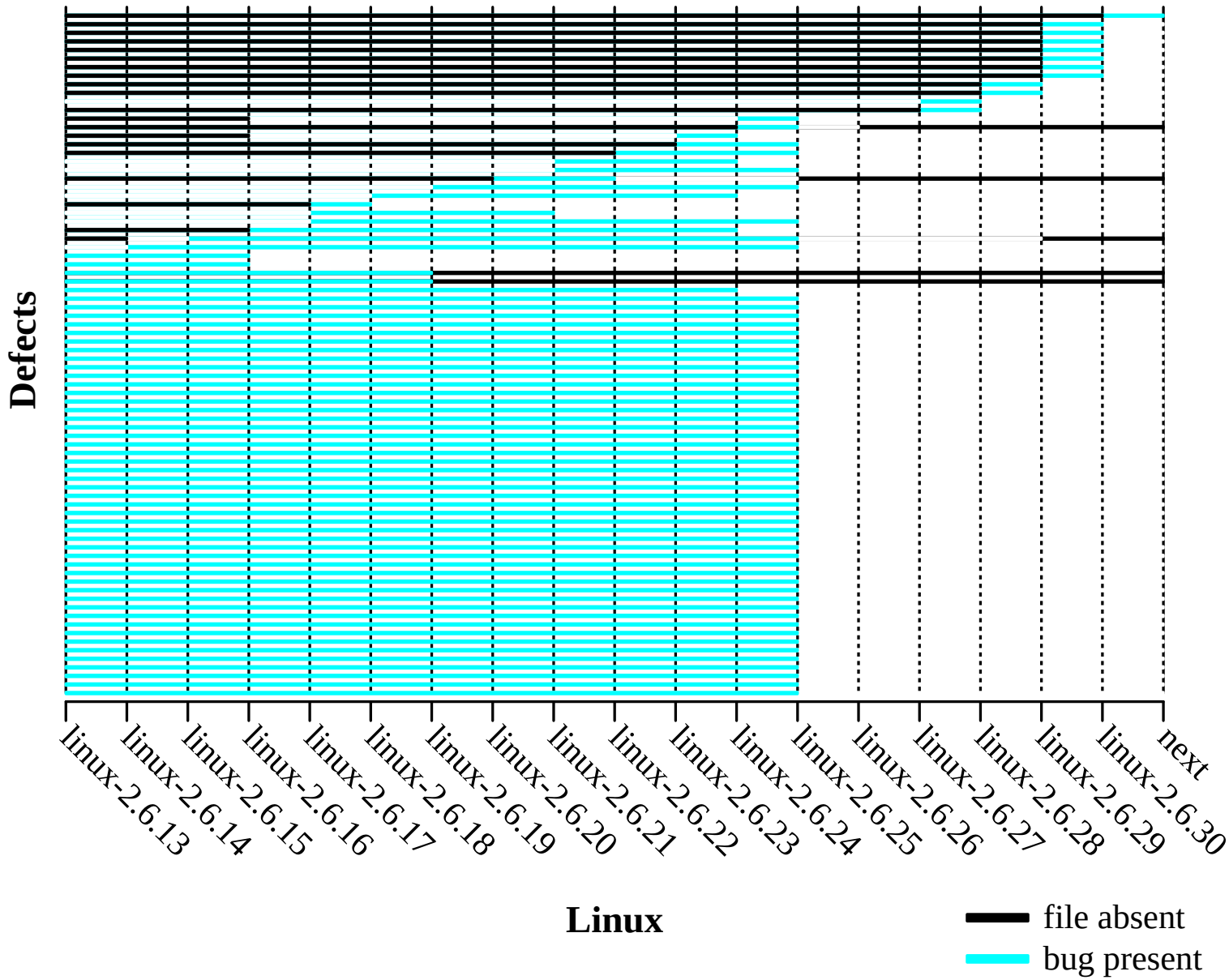
Bugs in the Linux kernel

- Linux is critical software.
  - Used in embedded systems, desktops, servers, etc.
- Linux is very large.
  - Almost 14 000 .c files
  - Over 8 million lines of code
  - Increase of almost 50% since 2006.
- Linux has both more and less experienced developers.
  - Maintainers, contributers, 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)
```

Lifetime of !x&y bugs



Goal: Find and fix bugs in C code

Approach: Coccinelle: `http://coccinelle.lip6.fr/`

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

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

Original code:

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

Semantic patch:

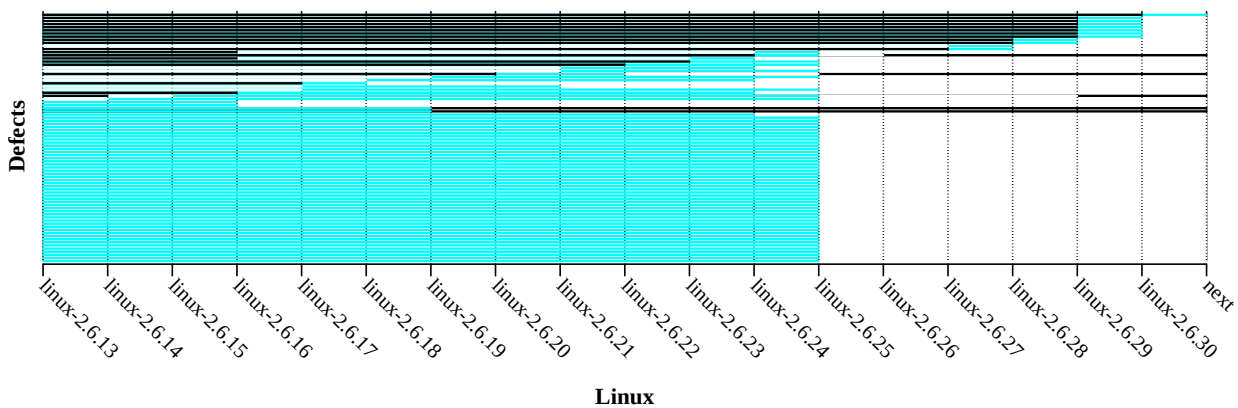
```
@@ expression E; constant C; @@
- !E & C
+ !(E & C)
```

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)



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;
```

A NULL pointer dereference semantic patch

Bug pattern: Dereference before a NULL test:

```
@@
expression x;
identifier fld;
@@
* x->fld
  ...
* x == NULL
```

- Isomorphisms cause `x == NULL` to also match eg `!x`.

A NULL pointer dereference semantic patch

Bug pattern: Dereference before a NULL test:

```
@@
expression x;
identifier fld;
@@
* x->fld
  ...
* x == NULL
```

- Isomorphisms cause `x == NULL` to also match eg `!x`.

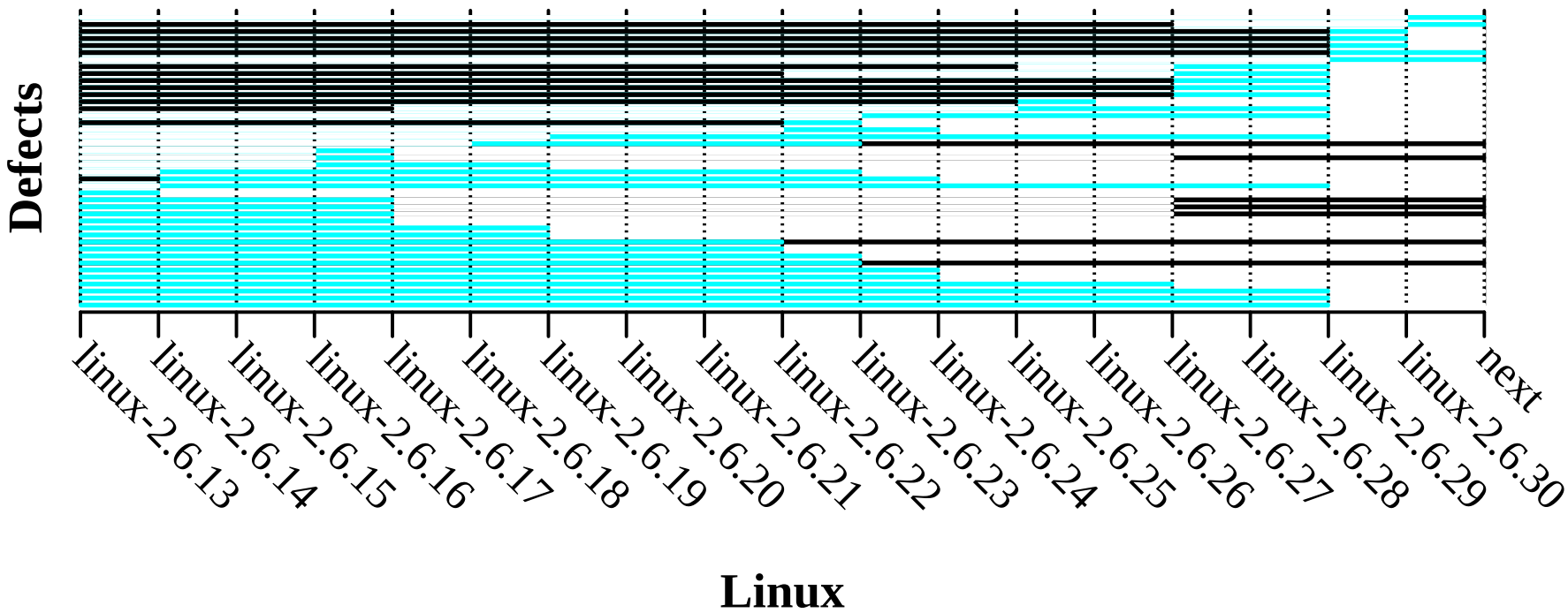
Lots of false positives:

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

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;
```

Results for the more constrained rule



Conclusion

A patch-like program matching and transformation language

Over 450 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

http://coccinelle.lip6.fr/

Kill bugs before they hatch!!!

COCCINELLE

http://coccinelle.lip6.fr/