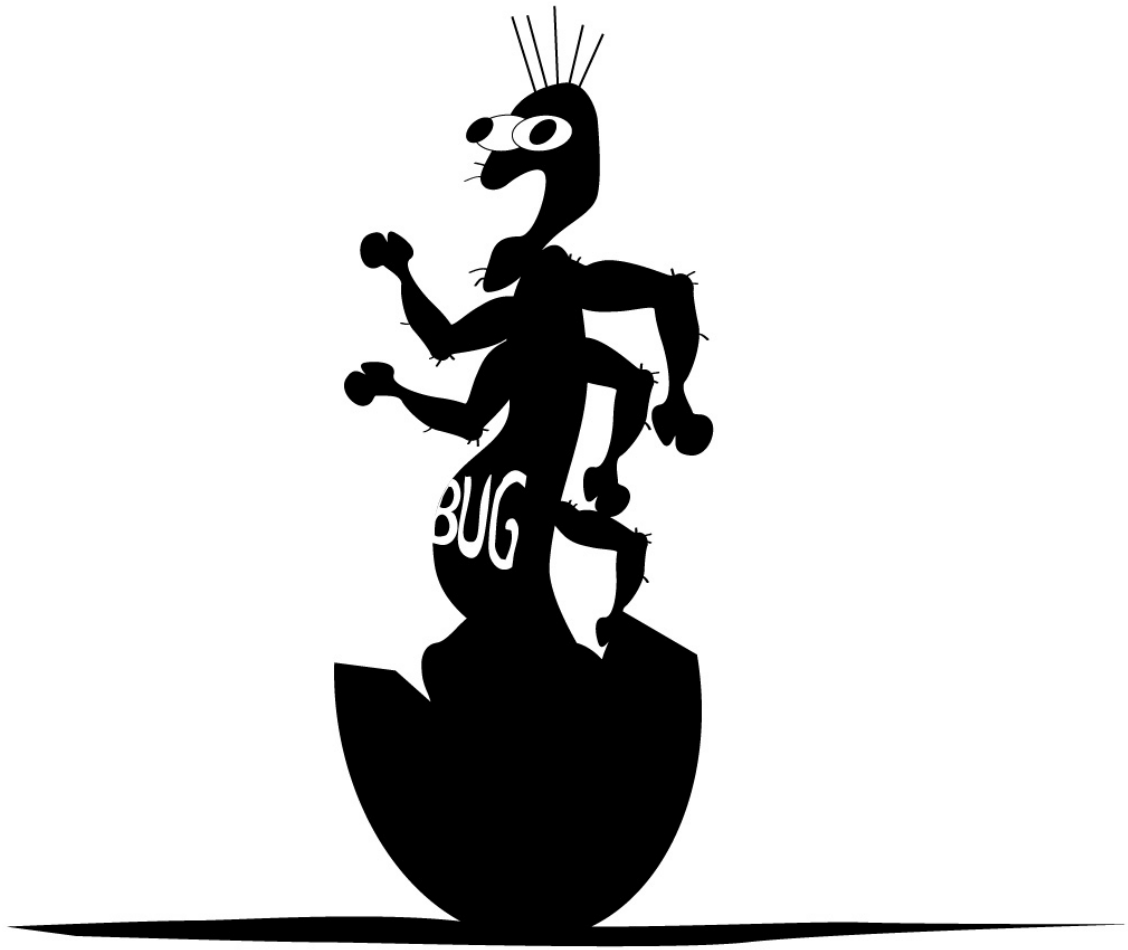
Bug Finding using Coccinelle

Julia Lawall (Inria/LIP6)

Joint work with
Gilles Muller, René Rydhof Hansen,
Nicolas Palix, Arie Middelkoop

September 21, 2012

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 18 000 .c files
  - Over 10.5 million lines of code
  - Increase of 8% since July 2011 (Linux 3.0).
- 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)
```

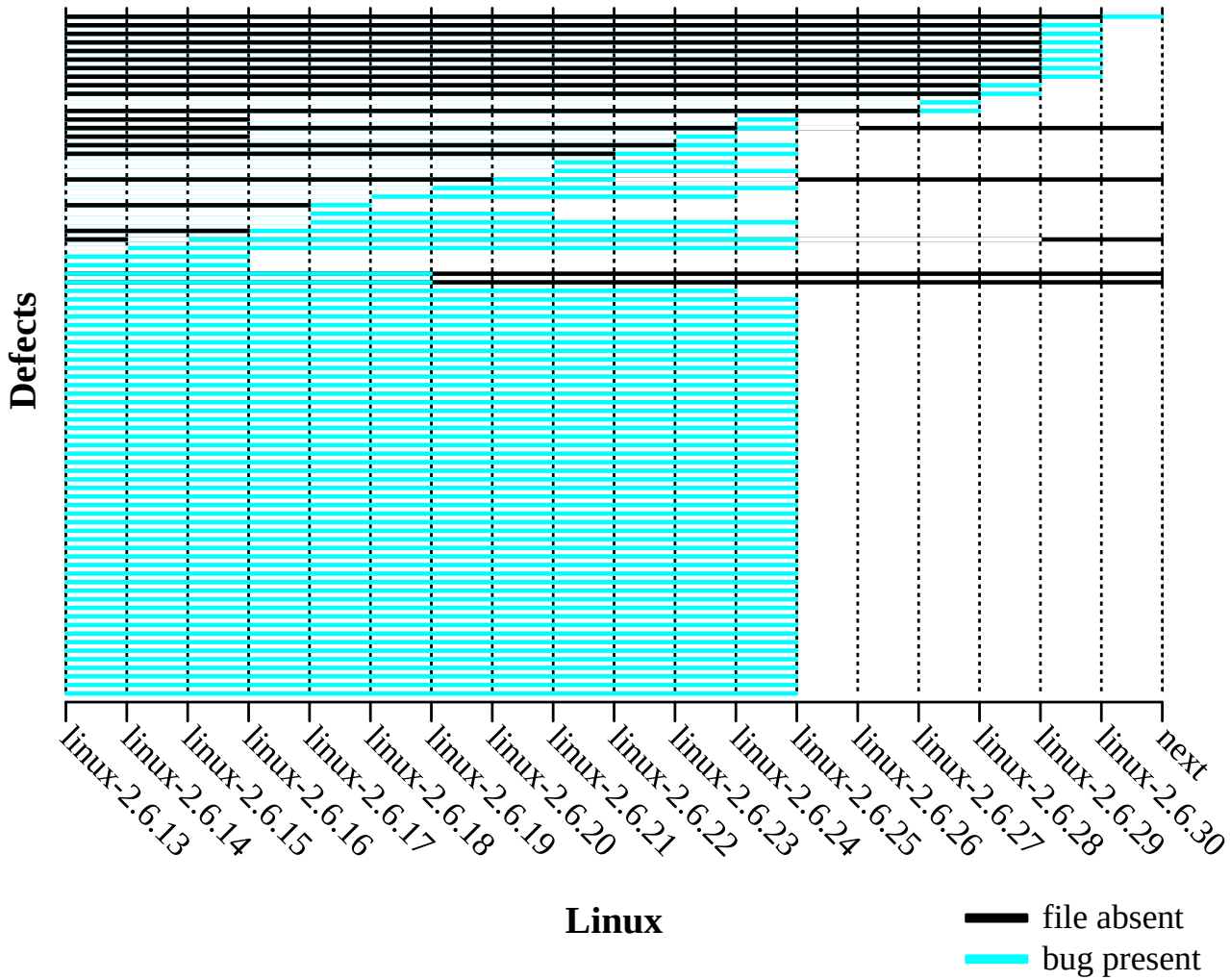
Bug: dereference of a possibly NULL value

```
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;
```

Issue

Isolated problems, but these bug types can occur many times

!x&y case:



Goal: Find and fix bugs in C code

Find once, fix everywhere.

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

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

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)
```

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:

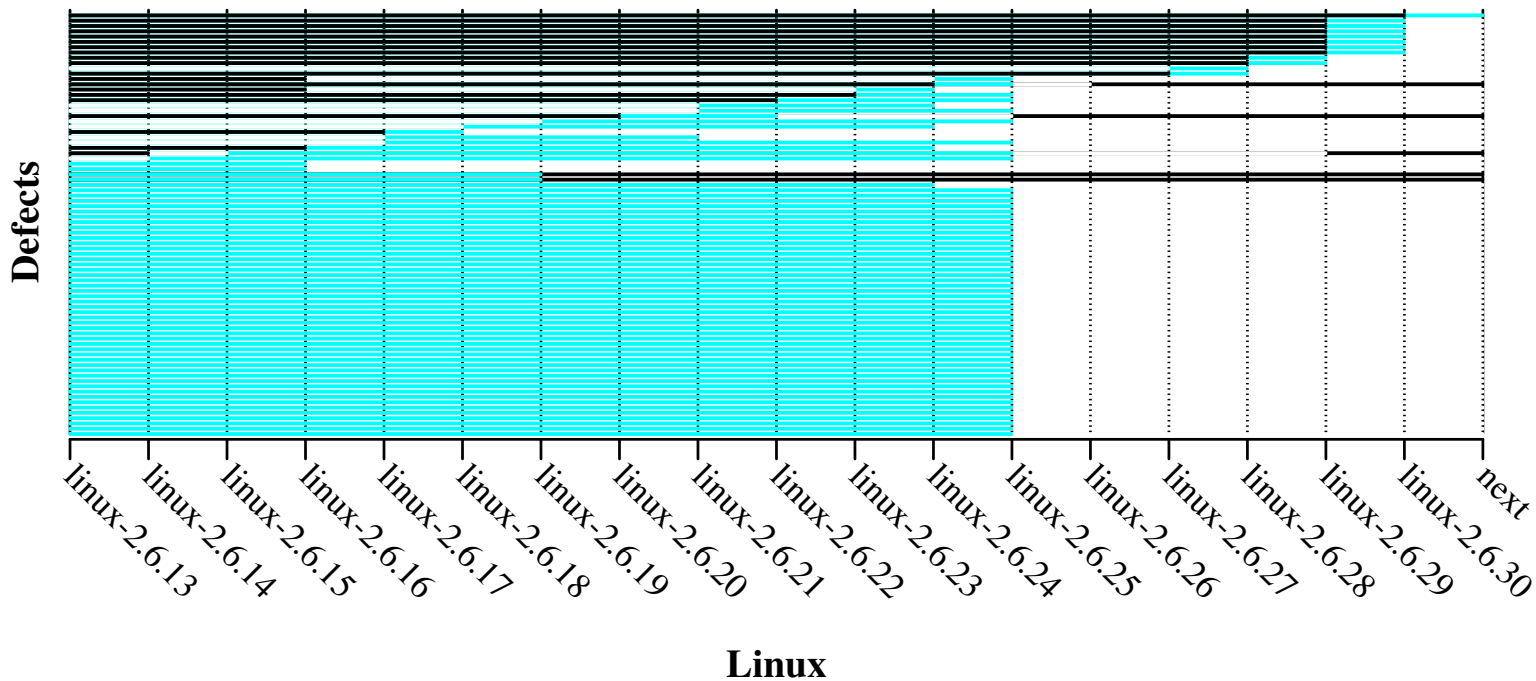
```
@@ expression E; constant C; @@
- !E & C
+ !(E & C)
```

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)



Other examples: dereference of a possibly NULL value

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

  T i = E->fld;

  ... when != E
      when != i
  if (E == NULL) S
```

Other examples: dereference of a possibly NULL value

```
@@
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;
```

Other examples

- Forgetting to initialize the return value.
- Testing the wrong value.
- Forgetting to free data, unlock locks, etc.
- Dereferencing freed data.
- Double-initializing the same variable, field, etc.
- And many others...

Conclusion

A patch-like program matching and transformation language

Over 1000 Coccinelle-based patches accepted into Linux

Coccinelle semantic patches available in the Linux source code

Used by other Linux developers

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

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