Hunting and fixing bugs all over the Linux kernel

Gustavo A. R. Silva
gustavo@embeddedor.com
@embeddedgus

The Linux Foundation's Core Infrastructure Initiative

Kernel Recipes
September 26, 2019
Paris, France
Who am I?

• Background in Embedded Systems.
• RTOS
• Embedded Linux.
• Volunteer at @kidsoncomputers
• Board of directors at @kidsoncomputers
Who am I?

• Background in Embedded Systems.
• RTOS
• Embedded Linux.
• Volunteer at @kidsoncomputers
• Board of directors at @kidsoncomputers
• Don’t speak Portuguese. :(
Agenda

• Coverity.
• Some bugs.
• Ancient bugs.
• Beyond bug fixing (KSPP).
• -Wimplicit-fallthrough.
• Super powers and responsibility.
• Results.
• Bonus.
Coverity

- Static code analyzer.
- Tons of false positives (This applies to all static code analyzers).
Coverity

- Static code analyzer.
- Tons of false positives (This applies to all static code analyzers).
- Helpful:

  $ git log --shortstat --author="Gustavo A. R. Silva"
grep Coverity | wc -l
  582
Coverity high impact issues

- Memory – illegal accesses (out-of-bounds access).
- Resource leaks (memory leaks).
- Uninitialized variables.
Coverity medium impact issues

- NULL pointer dereferences (before/after null check, explicit null dereference).
- Integer handling issues (bad bit shift operation).
- API usage errors (arguments in wrong order).
- Control flow issues.
Coverity work

• Look at every issue.
• Access to Coverity scans on mainline.
• Weekly scans every -rc.
• Now access to daily Coverity scans.
• Fix bugs in linux-next before they hit mainline.
Some Bugs
Incorrect type of variable

- while (counter < 10) {
+ while (counter < 1000) {
    value = dm_read_reg(cp110->base.ctx, addr);
    if (get_reg_field_value(
        value,
        FBC_STATUS,
        FBC_ENABLE_STATUS) == enabled)
        break;

-    msleep(10);
+    udelay(100);
    counter++;
}
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

```
1 files changed, 1 insertions, 1 deletions

diff --git a/drivers/gpu/drm/amd/display/dc/dce110/dce110_compressor.c
index e2994d337044..111c4921987f 100644
--- a/drivers/gpu/drm/amd/display/dc/dce110/dce110_compressor.c
+++ b/drivers/gpu/drm/amd/display/dc/dce110/dce110_compressor.c
@@ -143,7 +143,7 @@ static void wait_for_fbc_state_changed(
  struct dce110_compressor *cp110,
  bool enabled)
 {
-  uint8_t counter = 0;
+  uint16_t counter = 0;
    uint32_t addr = mmFBC_STATUS;
    uint32_t value;
```
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

```c
while (counter < 10) {
    value = dm_read_reg(cp110->base.ctx, addr);
    if (get_reg_field_value(
        value,
        FBC_STATUS,
        FBC_ENABLE_STATUS) == enabled)
        break;
    - msleep(10);
    + udelay(100);
    counter++;
}
```
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

- `uint8_t` → `[0-255]`
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

- uint8_t → [0-255]
- while (counter < 1000) - is always true.
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

- uint8_t → [0-255]
- while (counter < 1000) - is always true.
- uint16_t → [0-65,535]
Fix type of variable

- commit fe78627d430435d22316fe39f2012ece31bf23c2

- `uint8_t` → [0-255]
- `while (counter < 1000)` - is always true.
- `uint16_t` → [0-65,535]
- `while (counter < 1000)` - can be true or false.
Inconsistent IS_ERR and PTR_ERR

commit 52e17089d1850774d2ef583cdef2b060b84fca8c
Inconsistent IS_ERR and PTR_ERR

- commit 52e17089d1850774d2ef583cdef2b060b84fca8c
- \texttt{pinctrl != priv->vdev}
Inconsistent IS(ERR) and PTR(ERR)

- commit 52e17089d1850774d2ef583cdef2b060b84fca8c

- \texttt{pinctrl \neq priv->vdev}

- \texttt{PTR(ERR)(priv->vdev) \rightarrow PTR(ERR)(pinctrl)
Fix inconsistent IS_ERR and PTR_ERR

- commit 2b7db29b79190f7ad5c32f63594ba08b9b9171ea

```
Diffstat
-rw-r--r-- drivers/staging/media/imx/imx-media-csi.c 2

1 files changed, 1 insertions, 1 deletions

diff --git a/drivers/staging/media/imx/imx-media-csi.c b/drivers/staging/media/imx-media-csi.c
index 16cab40156ca..aeab05f682d9 100644
--- a/drivers/staging/media/imx/imx-media-csi.c
+++ b/drivers/staging/media/imx/imx-media-csi.c
@@ -1799,7 +1799,7 @@ static int imx_csi_probe(struct platform_device *pdev)
    priv->dev->of_node = pdata->of_node;
    pinctrl = devm_pinctrl_get_select_default(priv->dev);
    if (IS_ERR(pinctrl)) {
-       ret = PTR_ERR(priv->dev);
+       ret = PTR_ERR(pinctrl);
       dev_dbg(priv->dev, "devm_pinctrl_get_select_default() failed: %d\n", ret);
       if (ret != -ENODEV)
```
Fix inconsistent IS_ERR and PTR_ERR

• commit 2b7db29b79190f7ad5c32f63594ba08b9b9171ea

Easily caught using Coccinelle.
potential integer overflows

- commit 6f3472a993e7cb63cde5d818dcabc8e42fc03744

```c
feedback_divider =
-   (uint64_t)(target_pix_clk_khz * ref_divider * post_divider);
+   (uint64_t)(target_pix_clk_khz * ref_divider * post_divider);
feedback_divider *= 10;
/* additional factor, since we divide by 10 afterwards */
feedback_divider *= (uint64_t)(calc_pll_cs->fract_fb_divider_factor);
```

```c
/*Actual calculated value*/
- actual_calc_clk_khz = (uint64_t)(feedback_divider *
  calc_pll_cs->fract_fb_divider_factor) +
+ actual_calc_clk_khz = (uint64_t)feedback_divider *
  calc_pll_cs->fract_fb_divider_factor +
fract_feedback_divider;
actual_calc_clk_khz *= calc_pll_cs->ref_freq_khz;
actual_calc_clk_khz =
```
use-after-free

- commit 594619497f3d6d4b8d8440e6d380e8da9dcc9eeb

```
Diffstat
-rw-r--r--  drivers/net/ethernet/mellanox/mlx5/core/fpga/ipsec.c 3  
1 files changed, 2 insertions, 1 deletions

diff --git a/drivers/net/ethernet/mellanox/mlx5/core/fpga/ipsec.c  
b/drivers/net/ethernet/mellanox/mlx5/core/fpga/ipsec.c
--- a/drivers/net/ethernet/mellanox/mlx5/core/fpga/ipsec.c
+++ b/drivers/net/ethernet/mellanox/mlx5/core/fpga/ipsec.c
@@ -1061,8 +1061,9 @@ static int fpga_ipsec_fs_create_fte(struct mlx5_core...

    rule->ctx = mlx5_fpga_ipsec_fs_create_sa_ctx(dev, fte, is_egress);
    if (IS_ERR(rule->ctx)) {
+       int err = PTR_ERR(rule->ctx);
         kfree(rule);
-      return PTR_ERR(rule->ctx);
+      return err;
    }

    rule->fte = fte;
```
Incorrect bitwise operator

- commit e146471f588e4b8dcd7994036c1b47cc52325f00
- Introduced on Jul 14, 2019.
- Fixed on Jul 18, 2019.
- Never hit mainline.

```
diff --git a/drivers/net/ethernet/marvell/mvpp2/mvpp2_debugfs.c b/drivers/net/et
index 02dfeb13ccc..f9744a61e5dd 100644
--- a/drivers/net/ethernet/marvell/mvpp2/mvpp2_debugfs.c
+++ b/drivers/net/ethernet/marvell/mvpp2/mvpp2_debugfs.c
@@ -245,7 +245,7 @@ static int mvpp2_dbgfs_flow_c2_enable_show(struct seq_file *

        enabled = !(c2.attr[2] | MVPP22_CLS_C2_ATTR2_RSS_EN);
+        enabled = !(c2.attr[2] & MVPP22_CLS_C2_ATTR2_RSS_EN);

        seq_printf(s, "%-d\n", enabled);
```

- #define MVPP22_CLS_C2_ATTR2_RSS_EN BIT(30)
- The use of the bitwise OR operator '|' always leads to true.

Fix “missing return” in switch

• commit c5b974bee9d2ceae4c441ae5a01e498c2674e100

```
diff --git a/drivers/iio/accel/sca3000.c b/drivers/iio/accel/sca3000.c
index 4dceb75e3586..4964561595f5 100644
--- a/drivers/iio/accel/sca3000.c
+++ b/drivers/iio/accel/sca3000.c
@@ -797,6 +797,7 @@ static int sca3000_write_raw(struct iio_dev *indio_dev,
    mutex_lock(&st->lock);
    ret = sca3000_write_3db_freq(st, val);
    mutex_unlock(&st->lock);
+    return ret;
    default:
    return -EINVAL;
```
resource leaks

• commit 3b4acbb92dbda4829e021e5c6d5410658849fa1c

**perf script: Fix memory leaks in list_scripts()**

In case memory resources for *buf* and *paths* were allocated, jump to *out* and release them before return.

```c
diff --git a/tools/perf/ui/browsers/scripts.c b/tools/perf/ui/browsers/scripts.c
--- a/tools/perf/ui/browsers/scripts.c
+++ b/tools/perf/ui/browsers/scripts.c
@@ -133,8 +133,10 @@ static int list_scripts(char *script_name, bool *custom
    int key = ui_browser__input_window("perf script command",
           "Enter perf script command line (without perf script_args, ",", 0);

-   if (key != K_ENTER)
-       return -1;
+   if (key != K_ENTER) {
+       ret = -1;
+       goto out;
+   }
    sprintf(script_name, "%s script %s", perf, script_args);
} else if (choice < num + max_std) {
    strcpy(script_name, paths[choice]);
```
Ancient Bugs
Incorrect bitwise operator

- The use of the bitwise OR operator '|' always leads to true.

```c
diff --git a/tools/perf/tests/evsel-tp-sched.c b/tools/perf/tests/evsel-tp-sched.c
index 5f8501c68da4..5cbbab70bcddd0 100644
--- a/tools/perf/tests/evsel-tp-sched.c
+++ b/tools/perf/tests/evsel-tp-sched.c
@@ -17,7 +17,7 @@ static int perf_evsel__test_field(struct perf_evsel *evsel, c
        return -1;
    }

-    is_signed = !(field->flags | TEP_FIELD_IS_SIGNED);
+    is_signed = !(field->flags & TEP_FIELD_IS_SIGNED);
    if (should_be_signed && !is_signed) {
        pr_debug("%s: ":%s"") signedness(%d) is wrong, should be %d\n",
            evsel->name, name, is_signed, should_be_signed);
```
Incorrect bitwise operator

• commit 489338a717a0dfbbd5a3fabccf172b78f0ac9015
• 7-year-old bug (Tue Sep 18 11:56:28 2012).

```
diff --git a/tools/perf/tests/evsel-tp-sched.c b/tools/perf/tests/evsel-tp-sched.c
index 5f8501c68da4..5cbba70bcedd0 100644
--- a/tools/perf/tests/evsel-tp-sched.c
+++ b/tools/perf/tests/evsel-tp-sched.c
@@ -17,7 +17,7 @@ static int perf_evsel__test_field(struct perf_evsel *evsel, c
       return -1;
 }

- is_signed = !!(field->flags | TEP_FIELD_IS_SIGNED);
+ is_signed = !!(field->flags & TEP_FIELD_IS_SIGNED);
     if (should_be_signed && !is_signed) {
         pr_debug("%s: \"%s\" signedness(%d) is wrong, should be %d\n",
             evsel->name, name, is_signed, should_be_signed);
```

• The use of the bitwise OR operator '|' always leads to true.
(!x & y) strikes again

- commit 07c69f1148da7de3978686d3af9263325d9d60bd

```
    Diffstat
-rw-r--r--  drivers/usb/gadget/udc/net2272.c  2

1 files changed, 1 insertions, 1 deletions

diff --git a/drivers/usb/gadget/udc/net2272.c b/drivers/usb/gadget/udc/net2272.c
index 660878a19505..b77f3126580e 100644
--- a/drivers/usb/gadget/udc/net2272.c
+++ b/drivers/usb/gadget/udc/net2272.c
@@ -2083,7 +2083,7 @@ static irqreturn_t net2272_irq(int irq, void *dev)
    #if defined(PLX_PCI_RDK2)
       /* see if PCI int for us by checking irqstat */
       intcsr = readl(dev->rdk2.fpga_base_addr + RDK2_IRQSTAT);
-      if (intcsr & (1 << NET2272_PCI_IRQ)) {
+      if (!(intcsr & (1 << NET2272_PCI_IRQ))) {
          spin_unlock(&dev->lock);
          return IRQ_NONE;
    }
```
(!x & y) strikes again

• commit 07c69f1148da7de3978686d3af9263325d9d60bd

```diff
Diffstat
-rw-r--r-- drivers/usb/gadget/udc/net2272.c 2
1 files changed, 1 insertions, 1 deletions

diff --git a/drivers/usb/gadget/udc/net2272.c b/drivers/usb/gadget/udc/net2272.c
index 660878a19505..b77f3126580e 100644
--- a/drivers/usb/gadget/udc/net2272.c
+++ b/drivers/usb/gadget/udc/net2272.c
@@ -2083,7 +2083,7 @@ static irqreturn_t net2272_irq(int irq, void *dev)
     }
     #if defined(PLX_PCI_RDK2)
         /* see if PCI int for us by checking irqstat */
-        intcsr = readl(dev->rdk2.fpga_base_addr + RDK2_IRQSTAT);
+        if (!intcsr & (1 << NET2272_PCI_IRQ)) {
             spin_unlock(&dev->lock);
             return IRQ_NONE;
         }
```
(!x & y) strikes again

Gustavo A. R. Silva @embeddedgus · Feb 1
(!x & y) strikes again:
git.kernel.org/pub/scm/linux/

This bug has been out there since 2011.

Kieran Bingham @kieranbingham
Replying to @embeddedgus

I wonder what effect these bugs had on the devices to go unnoticed for so long. In a years time someone's going to suddenly see something work correctly :-)

2:50 PM · Feb 1, 2019 · Twitter for Android
Beyond bug fixing
Kernel Self Protection Project
Kernel Self Protection Project

• Variable Length Arrays (VLA) removal.
Kernel Self Protection Project

- Variable Length Arrays (VLA) removal.
- Defense-in-depth with struct_size() helper.
Kernel Self Protection Project

- Variable Length Arrays (VLA) removal.
- Defense-in-depth with struct_size() helper.
- Switch case fall-through
Variable Length Arrays

- Exhaust the stack: write to things following it.
- Jump over guard pages.
- Easy to find with compiler flag: -Wvla
Variable Length Arrays

- Exhaust the stack: write to things following it.
- Jump over guard pages.
- Easy to find with compiler flag: -Wvla
- Eradicated from the kernel in Linux v4.20. :}
Defense-in-depth & struct_size()
Defense-in-depth & struct_size()

```c
287 /*
288 * Compute a*b+c, returning SIZE_MAX on overflow. Internal helper for
289 * struct_size() below.
290 */
291 static inline __must_check size_t __ab_c_size(size_t a, size_t b, size_t c)
292 {
293     size_t bytes;
294     if (check_mul_overflow(a, b, &bytes))
295         return SIZE_MAX;
296     if (check_add_overflow(bytes, c, &bytes))
297         return SIZE_MAX;
298     return bytes;
299 }

300 /**
301 * struct_size() - Calculate size of structure with trailing array.
302 * @p: Pointer to the structure.
303 * @member: Name of the array member.
304 * @n: Number of elements in the array.
305 *
306 * Calculates size of memory needed for structure @p followed by an
307 * array of @n @member elements.
308 *
309 * Return: number of bytes needed or SIZE_MAX on overflow.
310 */
311 #define struct_size(p, member, n)
312     __ab_c_size(n,
313         offsetof(*p)->member) + __must_be_array((p)->member),
314         offsetof(*p)))
315 ```
Defense-in-depth & struct_size()

- Bluetooth: mgmt: Use struct_size() helper
- Commit 72bb169e024a20203e6044a81d5e41ae6ee0645b

Bluetooth: mgmt: Use struct_size() helper

One of the more common cases of allocation size calculations is finding the size of a structure that has a zero-sized array at the end, along with memory for some number of elements for that array. For example:

```c
struct mgmt_rp_get_connections {
    ...
    struct mgmt_addr_info addr[0];
} __packed;
```

Make use of the struct_size() helper instead of an open-coded version in order to avoid any potential type mistakes.

So, replace the following form:

```
sizeof(*rp) + (i * sizeof(struct mgmt_addr_info));
```

with:

```c
struct_size(rp, addr, i)
```

Also, notice that, in this case, variable rp_len is not necessary, hence it is removed.

This code was detected with the help of Coccinelle.

Signed-off-by: Gustavo A. R. Silva <gustavo@embeddedor.com>
Signed-off-by: Marcel Holtmann <marcel@holtmann.org>
Defense-in-depth & struct_size()

- Bluetooth: mgmt: Use struct_size() helper
- Commit 72bb169e024a20203e6044a81d5e41ae6ee0645b

```c
diff --git a/net/bluetooth/mgmt.c b/net/bluetooth/mgmt.c
index 150114e33b20..acb7c6d5643f 100644
--- a/net/bluetooth/mgmt.c
+++ b/net/bluetooth/mgmt.c
@@ -2588,7 +2588,6 @@ static int get_connections(struct sock *sk, struct
{
    struct mgmt_rp_get_connections *rp;
    struct hci_conn *c;
-   size_t rp_len;
    int err;
    u16 i;

@@ -2608,8 +2607,7 @@ static int get_connections(struct sock *sk, struct
          }
-   rp_len = sizeof(*rp) + (i * sizeof(struct mgmt_addr_info));
+   rp = kmalloc(rp_len, GFP_KERNEL);
     if (!rp) {
         err = -ENOMEM;
         goto unlock;
```
Defense-in-depth & struct_size()

• One day I found something interesting...
Defense-in-depth & struct_size()

- One day I found something interesting...
- Commit cffaaf0c816238c45cd2d06913476c83eb50f682

```diff
diff --git a/drivers/iommu/dmar.c b/drivers/iommu/dmar.c
index dc9f14811e0f..58dc70bffdb5 100644
--- a/drivers/iommu/dmar.c
+++ b/drivers/iommu/dmar.c
@@ -144,7 +144,7 @@ dmar_alloc_pci_notify_info(struct pci_dev *dev, unsigned long
     for (tmp = dev; tmp = tmp->bus->self)
     level++;
-
+  size = sizeof(*info) + level * sizeof(struct acpi_dmar_pci_path);
     if (size <= sizeof(dmar_pci_notify_info_buf)) {
       info = (struct dmar_pci_notify_info *)dmar_pci_notify_info_buf;
     } else {
```
Defense-in-depth & struct_size()

- Commit 57384592c43375d2c9a14d82aebbd95fdda9e9d
Defense-in-depth & struct_size()

- Commit 57384592c43375d2c9a14d82aebbdcc95fdda9e9d

```c
struct dmar_pci_path {
    u8 bus;
    u8 device;
    u8 function;
};

struct dmar_pci_notify_info {
    struct pci_dev *dev;
    unsigned long event;
    u16 seg;
    u16 level;
    struct acpi_dmar_pci_path path[];
};
```
Defense-in-depth & struct_size()

- Commit 57384592c43375d2c9a14d82aebbdcc5fdd9e9d

- New structure `dmar_pci_path` contains an extra field: `u8 bus`;
Defense-in-depth & struct_size()

- Commit 57384592c43375d2c9a14d82aebbdcc95fdda9e9d

- New structure `dmar_pci_path` contains an extra field: `u8 bus`;

- Overflow: `info -> path[level].bus = tmp -> bus -> number;`
Defense-in-depth & struct_size()

• Commit 57384592c43375d2c9a14d82aebbdcc95fdda9e9d

• New structure `dmar_pci_path` contains an extra field: `u8 bus`;

```c
541 struct acpi_dmar_pci_path {
542     u8 device;
543     u8 function;
544     u8 bus;
545 };
```

• Overflow: \( \text{info} \rightarrow \text{path[level].bus} = \text{tmp} \rightarrow \text{bus} \rightarrow \text{number} \);

```c
- size = sizeof(*info) + level * sizeof(struct acpi_dmar_pci_path);
+ size = sizeof(*info) + level * sizeof(info->path[0]);
```
Defense-in-depth & struct_size()

- Commit 57384592c43375d2c9a14d82aebbd9c95fdda9e9d

- New structure `dmar_pci_path` contains an extra field: `u8 bus;`

- Overflow: info→path[level].bus = tmp→bus→number;
  
```diff
+ size = sizeof(*info) + level * sizeof(struct acpi_dmar_pci_path);
- size = sizeof(*info) + level * sizeof(info->path[0]);
```

- 4-year-old+ bug (Thu Oct 2 11:50:25 2014)
Defense-in-depth & \texttt{struct\_size()}

- iommu/vt-d: Use \texttt{struct\_size()} helper
- Commit 553d66cb1e8667aadb57e3804775c5ce1724a49b

```diff
diff --git a/drivers/iommu/dmar.c b/drivers/iommu/dmar.c
index 9c49300e9fb7..6d969a172fbb 100644
--- a/drivers/iommu/dmar.c
+++ b/drivers/iommu/dmar.c
@@ -145,7 +145,7 @@ dmar_alloc_pci_notify_info(struct pci_dev *dev, unsigned long
for (tmp = dev; tmp; tmp = tmp->bus->self)
    level++;
-    size = sizeof(*info) + level * sizeof(info->path[0]);
+    size = \texttt{struct\_size}(info, path, level);
if (size <= sizeof(dmar_pci_notify_info \_buf)) {
    info = (struct dmar_pci_notify_info *)dmar_pci_notify_info \_buf;
} else {
```
Defense-in-depth & struct_size()

- iommu/vt-d: Use struct_size() helper
- Commit 553d66cb1e8667adb57e3804775c5ce1724a49b
- Could have prevented: 57384592c43375d2c9a14d82ae8bdc95fdda9e9d

```c
size = sizeof(*info) + level * sizeof(info->path[0]);
size = struct_size(info, path, level);
if (size <= sizeof(dmar_pci_notify_info_buf)) {
    info = (struct dmar_pci_notify_info *)dmar_pci_notify_info_buf;
} else {
```
Defense-in-depth & struct_size()

• Commit 76497732932f15e7323dc805e8ea8dc11bb587cf

--- a/drivers/net/ethernet/chelsio/cxgb3/l2t.h  
+++ b/drivers/net/ethernet/chelsio/cxgb3/l2t.h  
@@ -75,8 +75,8 @@

```c
struct foo {
    int stuff;
    struct boo array[];

```
Defense-in-depth & struct\_size()

- Commit 76497732932f15e7323dc805e8ea8dc11bb587cf
- 8-year-old bug (Tue Sep 6 13:59:13 2011).

**cxgb3/l2t: Fix undefined behaviour**

The use of zero-sized array causes undefined behaviour when it is not the last member in a structure. As it happens to be in this case.

Also, the current code makes use of a language extension to the C90 standard, but the preferred mechanism to declare variable-length types such as this one is a flexible array member, introduced in C99:

```c
struct foo {
    int stuff;
    struct boo array[];
};
```

```diff
--- a/drivers/net/ethernet/chelsio/cxgb3/l2t.h
+++ b/drivers/net/ethernet/chelsio/cxgb3/l2t.h
@@ -75,8 +75,8 @@ struct l2t_data {
     struct l2t_entry *rover;        /* starting point for next alloc */
   atomic_t nfree;                  /* number of free entries */
   rwlock_t lock;
-  struct l2t_entry l2tab[0];
+  struct l2t_entry l2tab[];
}
```

```
Defense-in-depth & struct_size()

- Commit 76497732932f15e7323dc805e8ea8dc11bb587cf
- 8-year-old bug (Tue Sep 6 13:59:13 2011).
- Bugfix backported all the way down to LTS Linux v3.16.74

```c
struct foo {
    int stuff;
    struct boo array[];
};
```

```diff
--- a/drivers/net/ethernet/chelsio/cxgb3/l2t.h
+++ b/drivers/net/ethernet/chelsio/cxgb3/l2t.h
@@ -75,8 +75,8 @@ struct l2t_data {
     struct l2t_entry *rover;    /* starting point for next al
     atomic_t nfree;              /* number of free entries */
     rwlock_t lock;
-    struct l2t_entry l2tab[0];
     struct rcu_head rcu_head;   /* to handle rcu cleanup */
+    struct l2t_entry l2tab[];
};
```
Switch case fall-through
Switch case fall-through

- Common Weakness Enumeration.
  CWE-484: Omitted Break Statement in Switch:

  “The program omits a break statement within a switch or similar construct, causing code associated with multiple conditions to execute. This can cause problems when the programmer only intended to execute code associated with one condition.”
Switch case fall-through

- Common Weakness Enumeration. CWE-484: Omitted Break Statement in Switch:

“The program omits a break statement within a switch or similar construct, causing code associated with multiple conditions to execute. This can cause problems when the programmer only intended to execute code associated with one condition.”

- Prone to error.
Switch case fall-through

• Common Weakness Enumeration. CWE-484: Omitted Break Statement in Switch:

“The program omits a break statement within a switch or similar construct, causing code associated with multiple conditions to execute. This can cause problems when the programmer only intended to execute code associated with one condition.”

• Prone to error.

“To enable -Wimplicit-fallthrough in Firefox, I had to annotate 287 intentional fallthroughs.”
- Chris Peterson. TPM on Mozilla’s Firefox team.
-Wimplicit-fallthrough
-Wimplicit-fallthrough

• Commit 7607a121f4617840fe645c65f090af6403738031

**dmaengine: fsldma: Mark expected switch fall-through**

Mark switch cases where we are expecting to fall through.

Fix the following warning (Building: powerpc-ppa8548_defconfig powerpc):

drivers/dma/fsldma.c: In function ‘fsl_dma_chan_probe’:  
drivers/dma/fsldma.c:1165:26: warning: this statement may fall through [-Wimplicit-fallthrough=]  
    chan->toggle_ext_pause = fsl_chan_toggle_ext_pause;  
      ^-------------------

drivers/dma/fsldma.c:1166:2: note: here  
    case FSL_DMA_IP_83XX:
    ^--

**Diffstat**

```
diff --git a/drivers/dma/fsldma.c b/drivers/dma/fsldma.c
index 23e0a356f167..ad72b3f42ffa 100644
--- a/drivers/dma/fsldma.c
+++ b/drivers/dma/fsldma.c
@@ -1163,6 +1163,7 @@ static int fsl_dma_chan_probe(struct fsldma_device *
     switch (chan->feature & FSL_DMA_IP_MASK) {
       case FSL_DMA_IP_85XX:
         chan->toggle_ext_pause = fsl_chan_toggle_ext_pause;
+        /* Fall through */
       case FSL_DMA_IP_83XX:
         chan->toggle_ext_start = fsl_chan_toggle_ext_start;
         chan->set_src_loop_size = fsl_chan_set_src_loop_size;
```
-Wimplicit-fallthrough

• Tons of warnings (2300+).
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
-Wimplicit-fallthrough

• Tons of warnings (2300+). Just on x86.
• Where do I even begin?
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
• Strategy:
-Wimplicit-fallthrough

- Tons of warnings (2300+). **Just on x86.**
- Where do I even begin?
- Count warnings in each file.
- x86 headers. Tons of warnings.
- Strategy: address x86, first.
• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
• Strategy: address x86, first.
• x86 gate keeper:
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
• Strategy: address x86, first.
• x86 gate keeper: **tglx.**
• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
• Strategy: address x86, first.
• x86 gate keeper: **tglx.**
• What could possibly go wrong?
-Wimplicit-fallthrough

- Tons of warnings (2300+). **Just on x86.**
- Where do I even begin?
- Count warnings in each file.
- x86 headers. Tons of warnings.
- Strategy: address x86, first.
- x86 gate keeper: **tglx.**
- What could possibly go wrong?
- First patch (2017) :)


-Wimplicit-fallthrough

- Tons of warnings (2300+). **Just on x86.**
- Where do I even begin?
- Count warnings in each file.
- x86 headers. Tons of warnings.
- Strategy: address x86, first.
- x86 gate keeper: **tglx.**
- What could possibly go wrong?
- First patch (2017) :) - **Flamed :/**
Tons of warnings (2300+). Just on x86.
Where do I even begin?
Count warnings in each file.
x86 headers. Tons of warnings.
Strategy: address x86, first.
x86 gate keeper: tglx.
What could possibly go wrong?
First patch (2017) :) - Flamed :/
Abort.
-Wimplicit-fallthrough

• Tons of warnings (2300+). **Just on x86.**
• Where do I even begin?
• Count warnings in each file.
• x86 headers. Tons of warnings.
• Strategy: address x86, first.
• x86 gate keeper: **tglx.**
• What could possibly go wrong?
• First patch (2017) :) - **Flamed :/**
• **Abort.** Rethink strategy.
-Wimplicit-fallthrough

- Tons of warnings (2300+). **Just on x86.**
- Where do I even begin?
- Count warnings in each file.
- x86 headers. Tons of warnings.
- Strategy: address x86, first.
- x86 gate keeper: **tglx.**
- What could possibly go wrong?
- First patch (2017) :) - **Flamed :/**
- **Abort.** Rethink strategy.
- Warnings finally addressed in 2019.
Unintentional fall-through bugs
Unintentional fall-through bugs

Gustavo A. R. Silva
@embeddedgus

A bugfix for a 12-year-old bug has been finally pulled and will be merged into mainline soon:

`git.kernel.org/pub/scm/linux/...

Yep; this bug has been out there since 2007. Briefly explained: the problem is that the code always returns "success" even on failure.

8:18 AM · May 8, 2019 · Twitter Web Client`
Unintentional fall-through bugs

- commit 1cbd7a64959d33e7a2a1fa2bf36a62b350a9fcbd
- Recently applied to LTS Linux v3.16.74 (a couple of days ago).

```diff
diff --git a/drivers/platform/x86/sony-laptop.c b/drivers/platform/x86/sony-laptop.c
index 4bf6ba3f78e6..2058445fc456 100644
--- a/drivers/platform/x86/sony-laptop.c
+++ b/drivers/platform/x86/sony-laptop.c
@@ -4424,14 +4424,16 @@ sony_pic_read_possible_resource(struct acpi_i
                return AE_OK;
            }
            
            +    case ACPI_RESOURCE_TYPE_END_TAG:
            +        return AE_OK;
            +
            default:
                printk("Resource %d isn't an IRQ nor an IO port\n",
                        resource->type);
            +        return AE_CTRL_TERMINATE;

-           case ACPI_RESOURCE_TYPE_END_TAG:
-                return AE_OK;
-            }
-            return AE_CTRL_TERMINATE;
```
Unintentional fall-through bugs

Bugs, bugs, ancient bugs! Another years-old bug found while working on the -Wimplicit-fallthrough stuff: lore.kernel.org/patchwork/patc...

This one was introduced in January 2012.

4:45 PM · Feb 18, 2019 · Twitter Web Client
Unintentional fall-through bugs

- commit cc5034a5d293dd620484d1d836aa16c6764a1c8c

```diff
diff --git a/drivers/gpu/drm/radeon/evergreen_cs.c b/drivers/gpu/drm/radeon/evergreen_cs.c
index f471537c852f..1e14c6921454 100644
--- a/drivers/gpu/drm/radeon/evergreen_cs.c
+++ b/drivers/gpu/drm/radeon/evergreen_cs.c
@@ -1299,6 +1299,7 @@ static int evergreen_cs_handle_reg(struct radeon_cs
       return -EINVAL;
 }
 ib[idx] += (u32)((reloc->gpu_offset >> 8) & 0xffffffff);
+   break;
   case CB_TARGET_MASK:
       track->cb_target_mask = radeon_get_ib_value(p, idx);
       track->cb_dirty = true;
```
Unintentional fall-through bugs

- commit cc5034a5d293dd620484d1d836aa16c6764a1c8c
- 7-year-old bug.

```diff
diff --git a/drivers/gpu/drm/radeon/evergreen_cs.c b/drivers/gpu/drm/radeon/evergreen_cs.c
index f471537c852f..1e14c6921454 100644
--- a/drivers/gpu/drm/radeon/evergreen_cs.c
+++ b/drivers/gpu/drm/radeon/evergreen_cs.c
@@ -1299,6 +1299,7 @@ static int evergreen_cs_handle_reg(struct radeon_cs
                 return -EINVAL;
             
             ib[idx] += (u32)((reloc->gpu_offset >> 8) & 0xffffffff);
+            break;
             
             case CB_TARGET_MASK:
                 track->cb_target_mask = radeon_get_ib_value(p, idx);
                 track->cb_dirty = true;
```
Unintentional fall-through bugs

- commit cc5034a5d293dd620484d1d836aa16c6764a1c8c
- 7-year-old bug.
- Bugfix applied to multiple stable trees.

```diff
diff --git a/drivers/gpu/drm/radeon/evergreen_cs.c b/drivers/gpu/drm/radeon/evergreen_cs.c
index 471537c852f..1e14c6921454 100644
--- a/drivers/gpu/drm/radeon/evergreen_cs.c
+++ b/drivers/gpu/drm/radeon/evergreen_cs.c
@@ -1299,6 +1299,7 @@ static int evergreen_cs_handle_reg(struct radeon_cs
    return -EINVAL;
>
    ib[idx] += (u32)((reloc->gpu_offset >> 8) & 0xffffffff);
+   break;
    }

  case CB_TARGET_MASK:
    track->cb_target_mask = radeon_get_ib_value(p, idx);
```
Unintentional fall-through bugs

Friday night, casually building ARCH=sh for the first time, I found a 10-year-old bug thanks to -Wimplicit-fallthrough:
lore.kernel.org/patchwork/patc...

This is the perfect way to officially start my weekend. Cheers! 🍻
Unintentional fall-through bugs

- commit 1ee1119d184bb06af921b48c3021d921bbd85bac
Unintentional fall-through bugs

• commit 1ee1119d184bb06af921b48c3021d921bbd85bac
• 10-year-old bug.
Unintentional fall-through bugs

- commit 1ee1119d184bb06af921b48c3021d921bbd85bac
- 10-year-old bug.
- Bugfix applied to multiple stable trees.

```diff
diff --git a/arch/sh/kernel/hw_breakpoint.c b/arch/sh/k
index 3bd010b4c55f..f10d64311127 100644
--- a/arch/sh/kernel/hw_breakpoint.c
+++ b/arch/sh/kernel/hw_breakpoint.c
@@ -157,6 +157,7 @@ int arch_bp_generic_fields(int sh_l
     switch (sh_type) {
         case SH_BREAKPOINT_READ:
             *gen_type = HW_BREAKPOINT_R;
-          break;
+          break;
         case SH_BREAKPOINT_WRITE:
             *gen_type = HW_BREAKPOINT_W;
             break;
```
I wonder what effect these bugs had on the devices to go unnoticed for so long. In a years time someone's going to suddenly see something work correctly :-)

2:50 PM · Feb 1, 2019 · Twitter for Android
-Wimplicit-fallthrough
After almost two years of work, -Wimplicit-fallthrough will be finally globally enabled in Linux v5.3. I'll go grab a beer. Have a great weekend everybody. 🐧

git.kernel.org/pub/scm/linux/

Merge tag 'Wimplicit-fallthrough-5.3-rc2' of git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-5.3'.

Pull Wimplicit-fallthrough enablement from Gustavo A. R. Silva:
"This marks switch cases where we are expecting to fall through, and globally enables the -Wimplicit-fallthrough option in the main Makefile.

Finally, some missing-break fixes that have been tagged for -stable:
- drm/amdkfd: Fix missing break in switch statement
- drm/amdgpu/gfx10: Fix missing break in switch statement
Worth it
On Tue, Aug 13, 2019 at 09:38:51PM +0800, Jonathan Cameron wrote:
> This got caught by the implicit fall through detection but is
> a bug rather than missing marking.
>
> Reported-by: 0-DAY kernel test infrastructure
> Signed-off-by: Jonathan Cameron <Jonathan.Cameron@huawei.com>
> Fixes: 741172d18e8a ("iio: light: noa1305: Add support for NOA1305")
> ---
> drivers/iio/light/noa1305.c | 1 +
> 1 file changed, 1 insertion(+)
>
> diff --git a/drivers/iio/light/noa1305.c b/drivers/iio/light/noa1305.c
> index 7b859ae1044d..5ebfbc52f541 100644
> --- a/drivers/iio/light/noa1305.c
> +++ b/drivers/iio/light/noa1305.c
> @@ -85,6 +85,7 @@ static int noa1305_scale(struct noa1305_priv *priv, int
>     case NOA1305_INTEGR_TIME_400MS:
>         *val = 100;
>         *val2 = 77 * 4;
> -    break;
>     case NOA1305_INTEGR_TIME_200MS:
>         *val = 100;
>         *val2 = 77 * 2;
>     --
>  2.20.1
>
Gustavo, your work caught a bug _before_ it hit Linus's tree this time :)
I'll go queue this up now, thanks for the fast response Jonathan.

greg k-h
Super powers and responsibility
My own tree
My own tree

- Why?
My own tree

• Why?

• Stuck at 90%.
My own tree

• Why?

• Stuck at 90%.

• Patches deliberately ignored.
My own tree

- Why?
- Stuck at 90%.
- Patches deliberately ignored.
- Forced to bypass people to get the job done.
Results
Contributions
Contributions

200+ commits upstream (KR2017)
Contributions

200+ commits upstream (KR2017)
750+ commits upstream (KR2018)
Contributions

200+ commits upstream (KR2017)
750+ commits upstream (KR2018)
1400+ commits upstream (KR2019)
Categories (10+)

- NULL pointer dereferences.
- Spectre vulnerabilities.
- API usage errors.
- Code maintainability issues.
- Constification.
- Control flow issues.
- Uninitialized variables.
- Incorrect expression.
- Integer handling issues.
- Miscellaneous
Types (38+)

• Variable Length Arrays (VLA)
• Integer overflows
• Bad memory allocation
• Dereference after null check.
• Dereference before null check.
• Dereference null return value.
• Explicit null dereference.
• Missing null check on return value.
• Arguments in wrong order.
• Ignored error return code.
• Unused value.
• Unused code.
• Unnecessary static on local variable.
• Missing return in switch
• Logical vs. bitwise operator
• Wrong operator used

• Spectre V1
• Memory leaks
• ‘Constant’ variable guards dead code.
• Missing break in switch.
• Uninitialized scalar variable.
• Array compared against 0.
• Identical code for different branches.
• Self assignment.
• Macro compares unsigned to 0.
• Code refactoring.
• Print error message on failure.
• Unnecessary cast on kmalloc.
• Use sizeof(*var) in kmalloc.
• Double free
• Copy-paste errors
• Read from pointer after free
Subsystems & Components impacted (38+)

- alsa-devel
- linux-arm-msm
- linuxmediatek
- linux-samsung-soc
- ath10k
- linux-block
- linux-mmc
- linux-scsi
- ceph-devel
- linux-clk
- linux-nfs
- linux-wireless
- linux-media

- cifs-client
- linux-crypto
- linux-omap
- linux-wpan
- dri-devel
- linux-dmaengine
- linux-parisc
- platform-driver-x86
- intel-gfx
- linux-fbdev
- linux-pci
- spi-devel-general
- linux-arm-kernel

- kvm
- linux-fpga
- linux-pm
- target-devel
- linux-acpi
- linux-iio
- linux-rdma
- tpmdd-devel
- linux-rockchip
- linux-input
- linux-renesas-soc
- xen-devel
Stable trees impacted (20)

- 5.3.y
- 5.2.y
- 5.1.y
- 5.0.y
- 4.20.y
- 4.19.y (LTS)
- 4.18.y
- 4.17.y
- 4.16.y
- 4.15.y
- 4.14.y (LTS)
- 4.13.y
- 4.12.y
- 4.11.y
- 4.10.y
- 4.9.y (LTS)
- 4.4.y (LTS)
- 4.1.y
- 3.18.y
- 3.16.y (LTS)
Stable trees impacted (20)

- 5.3.y
- 5.2.y
- 5.1.y
- 5.0.y
- 4.20.y
- 4.19.y (LTS)
- 4.18.y
- 4.17.y
- 4.16.y
- 4.15.y
- 4.14.y (LTS)
- 4.13.y
- 4.12.y
- 4.11.y
- 4.10.y
- 4.9.y (LTS)
- 4.4.y (LTS)
- 4.1.y
- 3.18.y
- 3.16.y (LTS)

Stable trees impacted (20)

- 5.3.y
- 5.2.y
- 5.1.y
- 5.0.y
- 4.20.y\(^{[2]}\)
- 4.19.y (LTS)
- 4.18.y
- 4.17.y
- 4.16.y
- 4.15.y
- 4.14.y (LTS)
- 4.13.y
- 4.12.y\(^{[1]}\)
- 4.11.y
- 4.10.y
- 4.9.y (LTS)
- 4.4.y (LTS)
- 4.1.y
- 3.18.y
- 3.16.y (LTS)

Stable trees impacted (20)

- 5.3.y\(^3\)
- 5.2.y
- 5.1.y
- 5.0.y
- 4.20.y\(^2\)
- 4.19.y (LTS)
- 4.18.y
- 4.17.y
- 4.16.y
- 4.15.y
- 4.14.y (LTS)
- 4.13.y
- 4.12.y\(^1\)
- 4.11.y
- 4.10.y
- 4.9.y (LTS)
- 4.4.y (LTS)
- 4.1.y
- 3.18.y
- 3.16.y (LTS)

\(^2\) VLAs eradicated from kernel. December 2018.
\(^3\) -Wimplicit-fallthrough globally enabled by default. September 2019
Bonus
Code of Conduct
My experience with CoC
My experience with CoC

• 1480 files changed, 3920 (+), 2961 (-)
My experience with CoC

- 1480 files changed, 3920 (+), 2961 (-)
- 1846 interactions in general.
My experience with CoC

● 1480 files changed, 3920 (+), 2961 (-)
● 1846 interactions in general.
● Some interesting “feedback”:
  – “This crap… !!”
  – “I hate when… !!”
  – Contempt.
Flexibility and persistence.
*KSPP moto suggested by Alexander Popov.*
Thank you!

Gustavo A. R. Silva

gustavo@embeddedor.com
@embeddedgus