

EMBEDDED
OPEN SOURCE
SUMMIT

A root canal static analysis based audit of Zephyr

Munawar Hafiz, OpenRefactory



#EmbeddedOSSummit @openrefactory



Background: Munawar Hafiz



Ph.D. from UIUC

Formative work on feasibility of
fixing security automatically



Senior Soft. Eng. at Coverity

Team released first commercial
vulnerability
detection tools for JavaScript and Android



Assistant Professor at Auburn

First research on practical bug fixing tools.
Best Student Research in all disciplines of CS
in 2013 (Awarded at the Turing Award Event)



OpenRefactory CEO

Started OpenRefactory with seed money
awarded by the National Science Foundation



Background: OpenRefactory

Intelligent Code Repair (iCR) Static Analysis Tool



**Finds more
critical bugs**



**Less than 5%
false positives**



**Automatically
synthesizes fixes**



Background: Open Source Security Audit

- Scale thorough security audit
- Work with Alpha-Omega, Linux Foundation and Python Software Foundation
- Scan top 2000 Python open source projects
- Detect 32 kinds of critical security problems
- Report the bug to maintainers and work with them to fix the bugs
- Last 5 months: 350 bugs, 120 security related, 40 high severity, 60% fixed



Background: How This Talk Was Planned?

- OpenRefactory is working on iCR for C
- We wanted understand the need of the C developers
- Zephyr is a major C developer community
- There is an active push to improve the code quality overall



Plan: Root Canal Static Analysis Based Audit

- Root Canal Term Borrowed from Refactoring Literature
- Interview maintainers about current practices
- Evaluate the results from current practices
- Analyze source code using other solutions



Current Practices

- Coverity scanner used in the CI/CD pipeline
- We looked into the bugs filed by the Coverity scanner between Jan 1, 2024 and Mar 31, 2024 (Q1 of 2024)
- These reports are public (Search with “Coverity” issue tag)
- The bug reports typically do not have much explanation. More inside the Coverity portal which has restricted access.



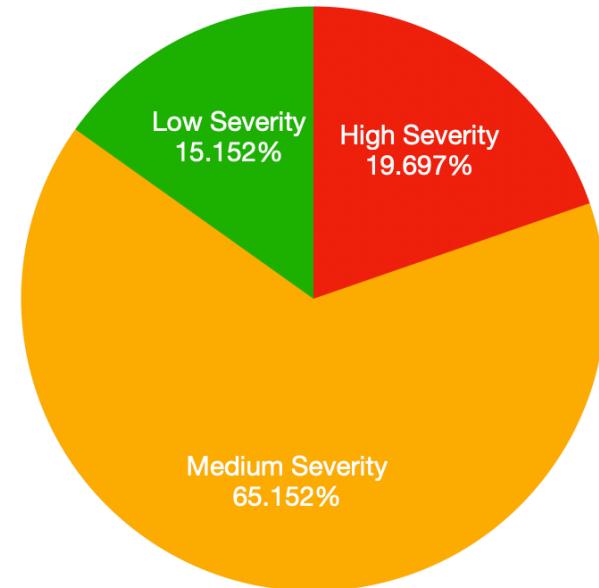
Coverity Code Scan (Q1 2024): Bug Resolution Status

- Total 77 bugs reported in 90 days, 11 duplicates
- 40 (35 unique) bugs were fixed ($35/66 = 53\%$)
- 9 false warnings, 7 WONTFIX issues ($16/66 = 24\%$)
- 17 Pending, 15 unique ($15/66 = 23\%$)



Coverity Code Scan (Q1 2024): Bug Severity

- Total 77 bugs reported in 90 days, 11 duplicates
- 13 High Severity
- 43 Medium Severity
- 10 Low Severity



Sample High Severity Bug, 1/4

zephyr / arch / common / shared_irq.c

Code Blame 211 lines (163 loc) · 5.28 KB ⚡ Code 55% faster with GitHub Copilot

Older Newer

2 months ago arch: Add support for dynamic ... ↗

```
36     #ifdef CONFIG_DYNAMIC_INTERRUPTS
37
38     static struct k_spinlock lock;
39
40     void z_isr_install(unsigned int irq, void (*routine)(const void *),
41                         const void *param)
42     {
43         struct z_shared_isr_table_entry *shared_entry;
44         struct _isr_table_entry *entry;
```

2 months ago arch: sw_isr_table: Update sh... ↗

```
45         struct _isr_table_entry *client;
```

7 months ago arch: Add support for dynamic ... ↗

```
46         unsigned int table_idx;
47         int i;
48         k_spinlock_key_t key;
49
50         table_idx = z_get_sw_isr_table_idx(irq);
51
52         /* check for out of bounds table index */
53         if (table_idx >= CONFIG_NUM_IRQS) {
54             return;
55         }
56
57         shared_entry = &z_shared_sw_isr_table[table_idx];
58         entry = &_sw_isr_table[table_idx];
```



Sample High Severity Bug, 2/4

[zephyr / include / arch / xtensa / irq.h](#)

[Code](#) [Blame](#) 131 lines (102 loc) · 2.94 KB Code 55% faster with GitHub Copilot

```
59
60     #ifdef CONFIG_2ND_LEVEL_INTERRUPTS
61     #ifdef CONFIG_3RD_LEVEL_INTERRUPTS
62     #define CONFIG_NUM IRQS (XCHAL_NUM_INTERRUPTS +\
63                           (CONFIG_NUM_2ND_LEVEL_AGGREGATORS +\
64                            CONFIG_NUM_3RD_LEVEL_AGGREGATORS) *\
65                           CONFIG_MAX_IRQ_PER_AGGREGATOR)
66     #else
67     #define CONFIG_NUM IRQS (XCHAL_NUM_INTERRUPTS +\
68                           CONFIG_NUM_2ND_LEVEL_AGGREGATORS *\
69                           CONFIG_MAX_IRQ_PER_AGGREGATOR)
70     #endif
71     #else
72     #define CONFIG_NUM IRQS XCHAL_NUM_INTERRUPTS
73     #endif
```

[zephyr / include / sw_isr_table.h](#)

[Code](#) [Blame](#) 86 lines (71 loc) · 2.17 KB Code 55% faster with GitHub Copilot

```
71                                     irq, tlags, (void *)&tunc, (const void *)param}
72
73 #define IRQ_TABLE_SIZE (CONFIG_NUM IRQS - CONFIG_GEN_IRQ_START_VECTOR)
74
75 #ifdef CONFIG_DYNAMIC_INTERRUPTS
```



Sample High Severity Bug, 3/4

[zephyr / arch / common / shared_irq.c](#)

Code Blame 211 lines (163 loc) · 5.28 KB ⚡ Code 55% faster with GitHub Copilot

Older ⚡ Newer

2 months ago arch: Add support for dynamic ... ↗

```
36     #ifdef CONFIG_DYNAMIC_INTERRUPTS
37
38     static struct k_spinlock lock;
39
40     void z_isr_install(unsigned int irq, void (*routine)(const void *),
41                         const void *param)
42     {
43         struct z_shared_isr_table_entry *shared_entry;
44         struct _isr_table_entry *entry;
```

2 months ago arch: sw_isr_table: Update sh... ↗

```
45         struct _isr_table_entry *client;
```

7 months ago arch: Add support for dynamic ... ↗

```
46         unsigned int table_idx;
47         int i;
48         k_spinlock_key_t key;
49
50         table_idx = z_get_sw_isr_table_idx(irq);
51
52         /* check for out of bounds table index */
53         if (table_idx >= CONFIG_NUM_IRQS) {
54             return;
55         }
56
57         shared_entry = &z_shared_sw_isr_table[table_idx];
58         entry = &_sw_isr_table[table_idx];
```



Sample High Severity Bug, 4/4

```
▼ ⌂ 4 ████ arch/common/shared_irq.c □
  .. @@ -50,7 +50,7 @@ void z_isr_install(unsigned int irq, void (*routine)(const void *),
50    50         table_idx = z_get_sw_isr_table_idx(irq);
51    51
52    52         /* check for out of bounds table index */
53 - 53     - if (table_idx >= CONFIG_NUM_IRQS) {
53 + 53     + if (table_idx >= IRQ_TABLE_SIZE) {
54    54             return;
55    55     }
56    56
  .. @@ -170,7 +170,7 @@ int z_isr_uninstall(unsigned int irq,
170   170         table_idx = z_get_sw_isr_table_idx(irq);
171   171
172   172         /* check for out of bounds table index */
173 - 173     - if (table_idx >= CONFIG_NUM_IRQS) {
173 + 173     + if (table_idx >= IRQ_TABLE_SIZE) {
174   174             return -EINVAL;
175   175     }
176   176
  ..
```



Sample Medium Severity Bug

```
v 14 subss/bluetooth/audio/shell/bap.c
..@ -2511,10 +2511,10 @@ static void stream_released_cb(struct bt_bap_stream *stream)
2511 2511
2512 2512         if (bap_stream->ep != NULL) {
2513 2513             struct bt_bap_ep_info ep_info;
2514 +         int err;
2515
2515 -         bt_bap_ep_get_info(bap_stream->ep, &ep_info);
2516 -
2517 -         if (ep_info.state != BT_BAP_EP_STATE_CODEC_CONFIGURED &&
2516 +         err = bt_bap_ep_get_info(bap_stream->ep, &ep_info);
2517 +         if (err == 0 && ep_info.state != BT_BAP_EP_STATE_CODEC_CONFIGURED &&
2518 2518             ep_info.state != BT_BAP_EP_STATE_IDLE) {
2519 2519                 group_can_be_deleted = false;
2520 2520                 break;
..@ + -3408,10 +3408,12 @@ static int cmd_recv_stats(const struct shell *sh, size_t argc, char *argv[])
3408 3408     static void print_ase_info(struct bt_bap_ep *ep, void *user_data)
3409 3409     {
3410 3410         struct bt_bap_ep_info info;
3411 +         int err;
3412
3412 -         bt_bap_ep_get_info(ep, &info);
3413 -         printk("ASE info: id %u state %u dir %u\n", info.id, info.state,
3414 -               info.dir);
3413 +         err = bt_bap_ep_get_info(ep, &info);
3414 +         if (err == 0) {
3415 +             printk("ASE info: id %u state %u dir %u\n", info.id, info.state, info.dir);
3416 +         }
3415 3417     }
3416 3418
3417 3419     static int cmd_print_ase_info(const struct shell *sh, size_t argc, char *argv[])
..@
```



CVEs in Zephyr

Year	Number of CVEs
2017	3
2019	1
2020	29
2021	19
2022	7
2023	24
2024	2

Source: <https://docs.zephyrproject.org/latest/security/vulnerabilities.html>



Recent Buffer Overflow

fs: fuse: buffer overflow vulnerability in the Zephyr FS

High ceolin published GHSA-mh67-4h3q-p437 on Feb 20

Package
zephyr (zephyr)

Affected versions
≤ 3.5

Patched versions
None

Severity
High 7.3 / 10

Description

Summary

Possible buffer overflow in `is_mount_point`:

https://github.com/zephyrproject-rtos/zephyr/blob/main/subsys/fs/fuse_fs_access.c#L69

Details

If the string passed via path parameter is PATH_MAX long, sprintf will overflow dir_path by one byte.

```
static bool is_mount_point(const char *path)
{
    char dir_path[PATH_MAX];

    sprintf(dir_path, "%s", path);
    return strcmp(dirname(dir_path), "/") == 0;
}
```

CVSS base metrics

Attack vector	Local
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	Low
Availability	High

CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:H

Weaknesses

CWE-120

Credits

 sploititem

Reporter



EMBEDDED
OPEN SOURCE
SUMMIT

Source: <https://github.com/zephyrproject-rtos/zephyr/security/advisories/GHSA-mh67-4h3q-p437>



Proposed Fix

zephyr / subsys / fs / fuse_fs_access.c

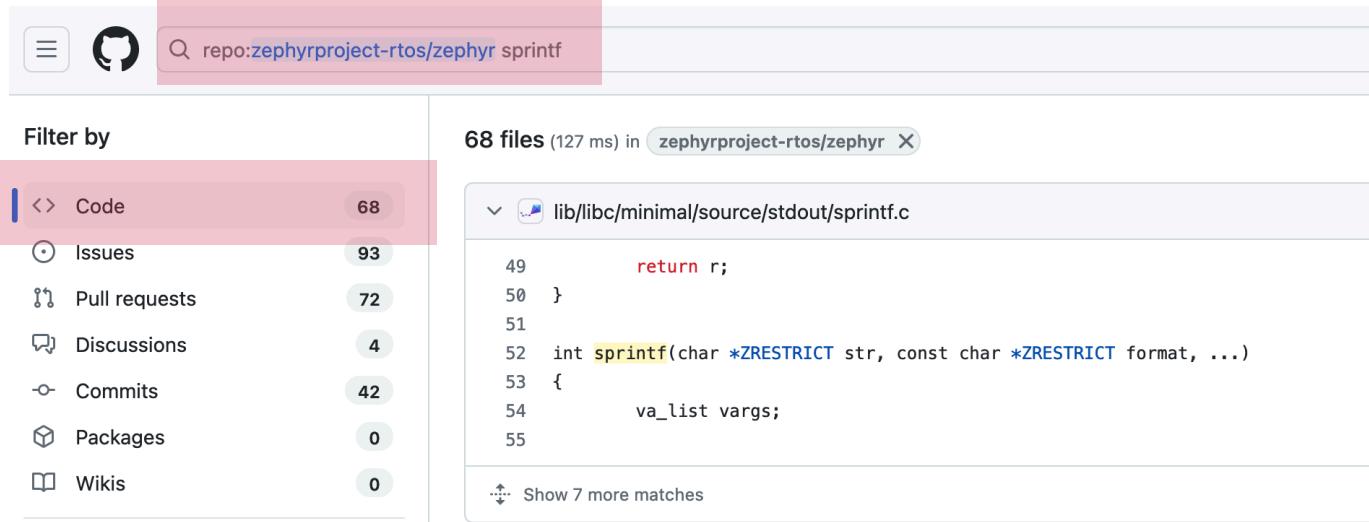
Code

Blame 557 lines (445 loc) · 10.2 KB

```
65     static bool is_mount_point(const char *path)
66     {
67         char dir_path[PATH_MAX];
68         size_t len;
69
70         len = strlen(path);
71         if (len >= sizeof(dir_path)) {
72             return false;
73         }
74
75         memcpy(dir_path, path, len);
76         dir_path[len] = '\0';
77         return strcmp(dirname(dir_path), "/") == 0;
78     }
79 }
```



Are There Similar Problems? 1/7



repo:zephyrproject-rtos/zephyr sprintf

Filter by

- Code 68
- Issues 93
- Pull requests 72
- Discussions 4
- Commits 42
- Packages 0
- Wikis 0

68 files (127 ms) in zephyrproject-rtos/zephyr

lib/libc/minimal/source/stdout/sprintf.c

```
49     return r;
50 }
51
52 int sprintf(char *ZRESTRICT str, const char *ZRESTRICT format, ...)
53 {
54     va_list vargs;
55 }
```

Show 7 more matches



EMBEDDED
OPEN SOURCE
SUMMIT

Source: <https://github.com/search?q=repo%3Azephyrproject-rtos%2Fzephyr%20sprintf&type=code>



Are There Similar Problems? 2/7

[zephyr / samples / boards / 96b_argonkey / sensors / src / main.c](#)

Code

Blame 356 lines (295 loc) · 9.7 KB

```
53         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_X, &accel_x);
54         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Y, &accel_y);
55         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Z, &accel_z);
56 #ifdef ARGONKEY_TEST_LOG
57     sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
58                                         (double)out_ev(&accel_y),
59                                         (double)out_ev(&accel_z));
60     printk("TRIG %s\n", out_str);
61 #endif
62
63     /* lsm6dls gyro */
64     sensor_sample_fetch_chan(dev, SENSOR_CHAN_GYRO_XYZ);
65     sensor_channel_get(dev, SENSOR_CHAN_GYRO_X, &gyro_x);
66     sensor_channel_get(dev, SENSOR_CHAN_GYRO_Y, &gyro_y);
67     sensor_channel_get(dev, SENSOR_CHAN_GYRO_Z, &gyro_z);
```



EMBEDDED
OPEN SOURCE
SUMMIT

Source: https://github.com/zephyrproject-rtos/zephyr/blob/5ff8249d198135492d0080efc9be47802004618e/samples/boards/96b_argonkey/sensors/src/main.c#L57



Are There Similar Problems? 3/7

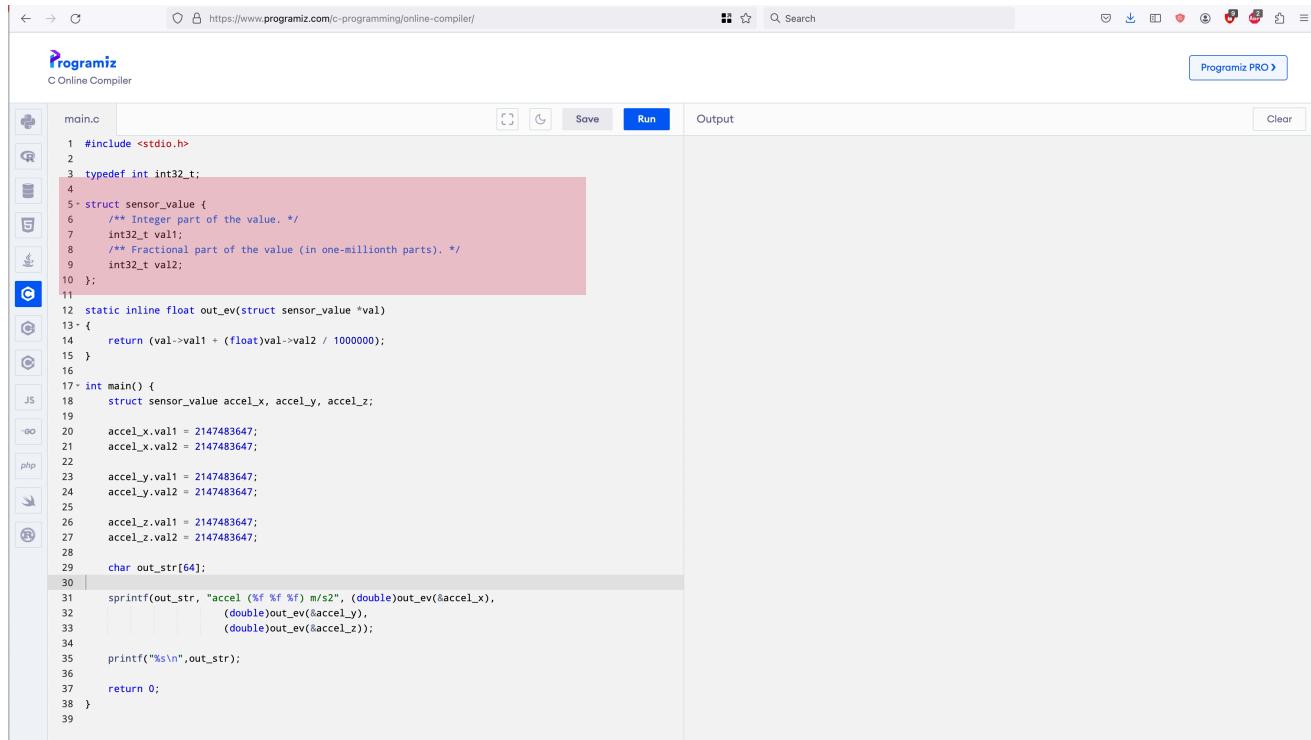
zephyr / samples / boards / 96b_argonkey / sensors / src / main.c

Code Blame 356 lines (295 loc) · 9.7 KB

```
32     static int lsm6dsl_trig_cnt;
33     #ifdef CONFIG_LSM6DSL_TRIGGER
34     static void lsm6dsl_trigger_handler(const struct device *dev,
35                                         const struct sensor_trigger *trig)
36     {
37         #ifdef ARGONKEY_TEST_LOG
38             char out_str[64];
39         #endif
40             struct sensor_value accel_x, accel_y, accel_z;
41             struct sensor_value gyro_x, gyro_y, gyro_z;
42
43             #if defined(CONFIG_LSM6DSL_EXT0_LIS2MDL)
44                 struct sensor_value magn_x, magn_y, magn_z;
45             #endif
46             #if defined(CONFIG_LSM6DSL_EXT0_LPS22HB)
47                 struct sensor_value press, temp;
48             #endif
49
50             lsm6dsl_trig_cnt++;
51
52             sensor_sample_fetch_chan(dev, SENSOR_CHAN_ACCEL_XYZ);
53             sensor_channel_get(dev, SENSOR_CHAN_ACCEL_X, &accel_x);
54             sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Y, &accel_y);
55             sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Z, &accel_z);
56             #ifdef ARGONKEY_TEST_LOG
57                 sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
58                                         (double)out_ev(&accel_y),
59                                         (double)out_ev(&accel_z));
60                 printk("TRIG %s\n", out_str);
61             #endif
62         }
63     }
```



Are There Similar Problems? 4/7



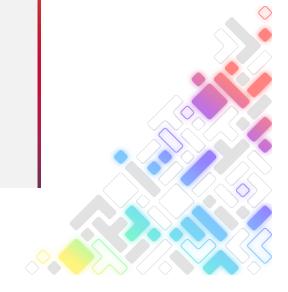
The screenshot shows a web-based C compiler interface on the Programiz website. The code editor contains the following C code:

```
main.c
1 #include <stdio.h>
2
3 typedef int int32_t;
4
5 struct sensor_value {
6     /** Integer part of the value. */
7     int32_t val1;
8     /** Fractional part of the value (in one-millionth parts). */
9     int32_t val2;
10 };
11
12 static inline float out_ev(struct sensor_value *val)
13 {
14     return (val->val1 + (float)val->val2 / 1000000);
15 }
16
17 int main() {
18     struct sensor_value accel_x, accel_y, accel_z;
19
20     accel_x.val1 = 2147483647;
21     accel_x.val2 = 2147483647;
22
23     accel_y.val1 = 2147483647;
24     accel_y.val2 = 2147483647;
25
26     accel_z.val1 = 2147483647;
27     accel_z.val2 = 2147483647;
28
29     char out_str[64];
30
31     sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
32             (double)out_ev(&accel_y),
33             (double)out_ev(&accel_z));
34
35     printf("%s\n", out_str);
36
37     return 0;
38 }
```

The code defines a `sensor_value` structure with integer and fractional parts, and a `main` function that calculates the output in m/s² using the `out_ev` function.



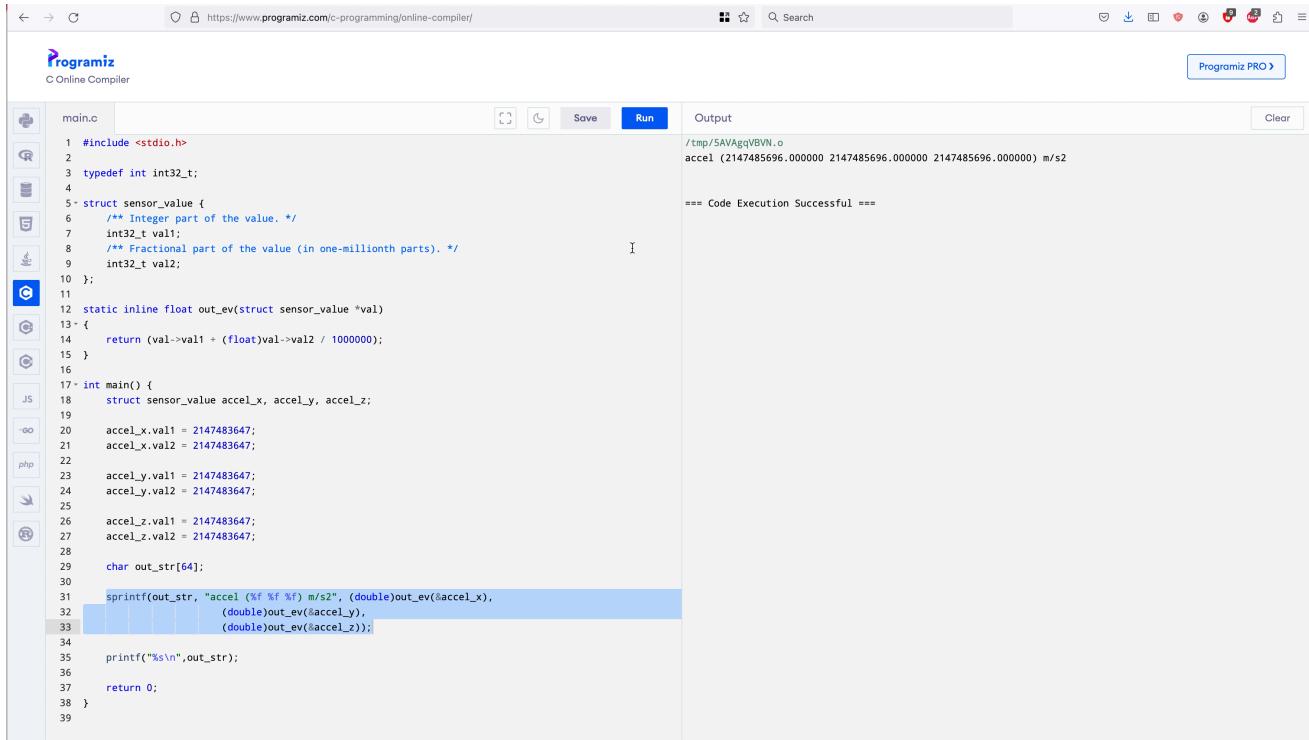
Are There Similar Problems? 5/7



A screenshot of the Programiz Online Compiler interface. The URL in the address bar is <https://www.programiz.com/c-programming/online-compiler/>. The code editor contains C code for calculating sensor values. The code includes definitions for a sensor value structure, a static inline function `out_ev` for summing integer and fractional parts, and a main function that initializes sensor values and prints them to `out_str`.

```
main.c
1 #include <stdio.h>
2
3 typedef int int32_t;
4
5 struct sensor_value {
6     /** Integer part of the value. */
7     int32_t val1;
8     /** Fractional part of the value (in one-millionth parts). */
9     int32_t val2;
10 };
11
12 static inline float out_ev(struct sensor_value *val)
13 {
14     return (val->val1 + (float)val->val2 / 1000000);
15 }
16
17 int main() {
18     struct sensor_value accel_x, accel_y, accel_z;
19
20     accel_x.val1 = 2147483647;
21     accel_x.val2 = 2147483647;
22
23     accel_y.val1 = 2147483647;
24     accel_y.val2 = 2147483647;
25
26     accel_z.val1 = 2147483647;
27     accel_z.val2 = 2147483647;
28
29     char out_str[64];
30
31     sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
32             (double)out_ev(&accel_y),
33             (double)out_ev(&accel_z));
34
35     printf("%s\n", out_str);
36
37     return 0;
38 }
```

Are There Similar Problems? 6/7



The screenshot shows a web-based online compiler interface on the Programiz website. The code editor contains C code named 'main.c' which defines a struct for sensor values, calculates the fractional part of the value, and then prints the acceleration values for three axes (x, y, z) in m/s². The output window shows the compiled binary file and the successful execution of the program.

```
main.c
1 #include <stdio.h>
2
3 typedef int int32_t;
4
5 struct sensor_value {
6     /** Integer part of the value. */
7     int32_t val1;
8     /** Fractional part of the value (in one-millionth parts). */
9     int32_t val2;
10 };
11
12 static inline float out_ev(struct sensor_value *val)
13 {
14     return (val->val1 + (float)val->val2 / 1000000);
15 }
16
17 int main() {
18     struct sensor_value accel_x, accel_y, accel_z;
19
20     accel_x.val1 = 2147483647;
21     accel_x.val2 = 2147483647;
22
23     accel_y.val1 = 2147483647;
24     accel_y.val2 = 2147483647;
25
26     accel_z.val1 = 2147483647;
27     accel_z.val2 = 2147483647;
28
29     char out_str[64];
30
31     sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
32             (double)out_ev(&accel_y),
33             (double)out_ev(&accel_z));
34
35     printf("%s\n", out_str);
36
37     return 0;
38 }
```

Output

```
/tmp/5AVAgqVBVN.o
accel (2147485696.000000 2147485696.000000 2147485696.000000) m/s2
== Code Execution Successful ==
```



Are There Similar Problems? 7/7

zephyr / samples / boards / 96b_argonkey / sensors / src / main.c

Code

Blame 356 lines (295 loc) · 9.7 KB

```
53         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_X, &accel_x);
54         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Y, &accel_y);
55         sensor_channel_get(dev, SENSOR_CHAN_ACCEL_Z, &accel_z);
56 #ifdef ARGONKEY_TEST_LOG
57     sprintf(out_str, "accel (%f %f %f) m/s2", (double)out_ev(&accel_x),
58                                         (double)out_ev(&accel_y),
59                                         (double)out_ev(&accel_z));
60     printf("TRIG %s\n", out_str);
61 #endif
62
63     /* lsm6dls gyro */
64     sensor_sample_fetch_chan(dev, SENSOR_CHAN_GYRO_XYZ);
65     sensor_channel_get(dev, SENSOR_CHAN_GYRO_X, &gyro_x);
66     sensor_channel_get(dev, SENSOR_CHAN_GYRO_Y, &gyro_y);
67     sensor_channel_get(dev, SENSOR_CHAN_GYRO_Z, &gyro_z);
```



EMBEDDED
OPEN SOURCE
SUMMIT

Source: https://github.com/zephyrproject-rtos/zephyr/blob/5ff8249d198135492d0080efc9be47802004618e/samples/boards/96b_argonkey/sensors/src/main.c#L57



Beyond Coverity: CodeChecker

- **CodeChecker introduced in Release v3.5 (October 2023)**
- **Easy way for end users to run an open source static analysis tool**
- **0 bugs found saying that “CodeChecker” was used (Q1 2024)**

The screenshot shows the CodeChecker website homepage. At the top, there's a navigation bar with links to Home, Quick Howto, Analyzer User Guide, Web User Guide, a search bar, and GitHub edit links. On the left, a sidebar lists "Main features" including Command line C/C++ Analysis, Web-based report storage, Command line features, Usage flow, User documentation, C/C++ Analysis, Web based report management, Common Tools, Helper Scripts, and an Install guide. Below the sidebar, there's information about installing via pip or snap, and links for Mac OS X, Docker, Visual Studio Code plugin, and GitHub Actions CI. In the center, there's a large graphic of a bug inside a target symbol, with the text "CodeChecker" below it. At the bottom, there's a GitHub Actions CI status bar showing "codechecker-tests" passing, "chat on gitter" active, "docs" passing, and "openssf scorecard 8.8". A note states that CodeChecker is a static analysis infrastructure built on the LLVM/Clang Static Analyzer toolchain, replacing scan-build in a Linux or macOS (OS X) development environment.



EMBEDDED
OPEN SOURCE
SUMMIT

Source: <https://codechecker.readthedocs.io/en/latest/>

Bug Filed By CodeChecker

```
v ⏪ 3 arch/x86/core/acpi.c ⏪
.... @@ -27,7 +27,7 @@ static void find_rsdp(void)
27   27 {
28   28     uint8_t *bda_seg, *zero_page_base;
29   29     uint64_t *search;
30  - 30     uintptr_t search_phys, rsdp_phys = 0U;
30  + 30     uintptr_t search_phys, rsdp_phys;
31   31     size_t search_length = 0U, rsdp_length;
32   32
33   33     if (is_rsdp_searched) {
.... @@ -94,7 +94,6 @@ static void find_rsdp(void)
94   94     search_length = 128 * 1024;
95   95     z_phys_map((uint8_t **)&search, search_phys, search_length, 0);
96   96
97  - 97     rsdp_phys = 0U;
98   97     for (int i = 0; i < 128*1024/8; i++) {
99   98       if (search[i] == ACPI_RSDP_SIGNATURE) {
100   99         rsdp_phys = search_phys + i * 8;
....
```

Source: <https://github.com/zephyrproject-rtos/zephyr/pull/61706/commits/11a28a5d93f02f41e71620b1c9a1bf1e1e13ece0>



Bug Filed By CodeChecker

```
✓ ⏷ 4 ████ lsm6dsv16x_reg.c □  
.. @@ -5330,8 +5330,8 @@ int32_t lsm6dsv16x_fsm_mode_set(stmdev_ctx_t *ctx, lsm6dsv16x_fsm_mode_t val)  
5330 5330     ret += lsm6dsv16x_read_reg(ctx, LSM6DSV16X_FSM_ENABLE, (uint8_t *)&fsm_enable, 1);  
5331 5331     if (ret != 0) { goto exit; }  
5332 5332  
5333 -     if ((val.fsm1_en | val.fsm2_en | val.fsm1_en | val.fsm1_en  
5334 -         | val.fsm1_en | val.fsm2_en | val.fsm1_en | val.fsm1_en) == PROPERTY_ENABLE)  
5333 +     if ((val.fsm1_en | val.fsm2_en | val.fsm3_en | val.fsm4_en  
5334 +         | val.fsm5_en | val.fsm6_en | val.fsm7_en | val.fsm8_en) == PROPERTY_ENABLE)  
5335 5335     {  
5336 5336         emb_func_en_b.fsm_en = PROPERTY_ENABLE;  
5337 5337     }  
..
```



EMBEDDED
OPEN SOURCE
SUMMIT

Source: <https://github.com/STMicroelectronics/lsm6dsv16x-pid/commit/9dca50889c8a4f7ea70b585a138e6841d4cc40c6>



Beyond Coverity: Snyk

iofbd > Projects > iofbd/zephyr main

Code Analysis

Overview History Settings

IMPORTED BY
it@iofbd.com

LIFECYCLE
+ Add a value

PROJECT OWNER
+ Add a project owner

ENVIRONMENT
+ Add a value

BUSINESS CRITICALITY
+ Add a value

ANALYSIS SUMMARY
8998 analyzed files (27%) Repo breakdown

Issues 257

Reset Search...

SEVERITY
High 0
Medium 228
Low 29

PRIORITY SCORE
Scored between 0 - 1000

STATUS
Open 257
Ignored 0

LANGUAGES
Python 212
C/C++ 42
Unknown 3

257 of 257 issues

M Path Traversal

SNYK CODE: CWE-23

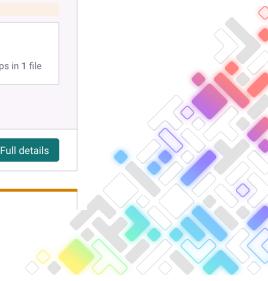
SCORE 611

284 list_drives()
285 else:
286 if not args.input:
287 error("Need input file")
288 with open(args.input, mode='rb') as f:

Unsanitized input from a command line argument flows into open, where it is used as a path. This may result in a Path Traversal vulnerability and allow an attacker to read arbitrary files.
scripts/build/u2conv.py

Learn about this type of vulnerability and how to fix it

Ignore Full details



What Does Snyk Find?

- 41 bugs
- 31 medium, 10 low
- 27 FP
- 14 worth a look

Bug Type	Severity	Total Bugs	TP	FP
Improper NULL termination	Low ▾	1	0	1
NULL Dereference	Medium ▾	1	0	1
Missing Release of Memory	Low ▾	1	0	1
Division By Zero	Medium ▾	2	2(potentially)	0
	Low ▾	1	0	1
Integer Overflow	Medium ▾	8	3(potentially)	5
	Low ▾	7	0	7
User Controlled Pointer	Medium ▾	9	9(potentially)	0
Use After Free	Medium ▾	8	0	8
Double Free	Medium ▾	3	0	3



Beyond Coverity: Semgrep

The screenshot shows the Semgrep dashboard with the following key metrics:

- Code**:
 - High severity: 28
 - Open findings: 306
 - PR/MR fix rate: 0%
- Supply Chain**:
 - Reachable vulns: 0
 - Unreachable vulns: 0
 - Undetermined vulns: 0

Most findings table (filtered for project name):

Project name	Open findings	High severity	Fix rate
fazledyn/zephyr	306	28	0%
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-

Rules summary table:

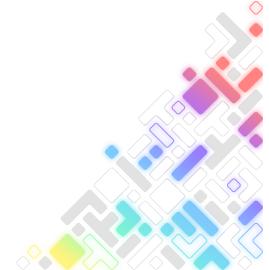
Rule	Ignored	Fix rate	Total
memset-removal	0%	0%	166
sizeof-pointer-type	0%	0%	55
subprocess-shell-true	0%	0%	12
avoid-pickle	0%	0%	9
var-in-href	0%	0%	7

New advisories section:

- External Control of File Name or Path in volta (Python - 2 days ago)
- Uncontrolled Resource Consumption in github.com/quic-go/quic-go (Go - 3 days ago)
- Insufficient Verification of Data Authenticity in @kindspells/astro-shield (JavaScript - 3 days ago)
- Resource Leak in @electron/packager (JavaScript - 6 days ago)
- Reliance on Uncontrolled Component in @workos-inc/authkit-nextjs (JavaScript - 6 days ago)



EMBEDDED
OPEN SOURCE
SUMMIT



What Does Semgrep Find?

- 233 bugs
- 3 high
- 60 medium, 170 low
- 2 TP, 231 FP

Bug Type	Severity	Total Bugs	TP	FP
Memset Removal	Low ▾	166	0	166
Sizeof Pointer Type	Medium ▾	55	0	55
Don't Call System	Low ▾	4	0	4
Local Variable Malloc Free	High ▾	3	0	3
Insecure Use strtok Function	Medium ▾	2	2 (Wontfix)	0
Snprintf Return Value	Medium ▾	2	0	2
Snprintf Source Size	Medium ▾	1	0	1



High Severity Bug, 1/4

Local Variable Malloc Free

This expression points to memory that has been freed. This can lead to a segmentation fault or memory corruption.

[zephyr / lib / libc / newlib / libc-hooks.c](#)

Code

Blame

576 lines (478 loc) · 13.2 KB

```
396     /* Close dynamic non-recursive lock */
397     void __retarget_lock_close(_LOCK_T lock)
398     {
399         __ASSERT_NO_MSG(lock != NULL);
400 #ifndef CONFIG_USERSPACE
401         free(lock);
402 #else
403         k_object_release(lock);
404 #endif /* !CONFIG_USERSPACE */
405 }
```



High Severity Bug, 2/4

[zephyr](#) / tests / lib / newlib / thread_safety / src / locks.c

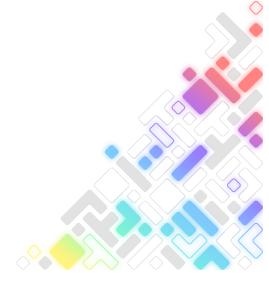
Code

Blame 465 lines (377 loc) · 13.3 KB

```
60     ZTEST(newlib_thread_safety_locks, test_retargetable_lock_sem)
61     {
62         _LOCK_T lock = NULL;
63
64         /* Dynamically allocate and initialise a new lock */
65         __retarget_lock_init(&lock);
66         zassert_not_null(lock, "non-recursive lock init failed");
67
68         /* Acquire lock and verify acquisition */
69         __retarget_lock_acquire(lock);
70         zassert_equal(__retarget_lock_try_acquire(lock), 0,
71                         "non-recursive lock acquisition failed");
72
73         /* Release lock and verify release */
74         __retarget_lock_release(lock);
75         zassert_not_equal(__retarget_lock_try_acquire(lock), 0,
76                         "non-recursive lock release failed");
77
78         /* Close and deallocate lock */
79         __retarget_lock_close(lock);
80     }
```



EMBEDDED
OPEN SOURCE
SUMMIT



High Severity Bug, 3/4

[zephyr](#) / [lib](#) / [libc](#) / [newlib](#) / [libc-hooks.c](#)

Code

Blame 576 lines (478 loc) · 13.2 KB

```
357
358     /* Create a new dynamic non-recursive lock */
359     void __retarget_lock_init(_LOCK_T *lock)
360     {
361         __ASSERT_NO_MSG(lock != NULL);
362
363         /* Allocate semaphore object */
364 #ifndef CONFIG_USERSPACE
365             *lock = malloc(sizeof(struct k_sem));
366 #else
367             *lock = k_object_alloc(K_OBJ_SEM);
368 #endif /* !CONFIG_USERSPACE */
369         __ASSERT(*lock != NULL, "non-recursive lock allocation failed");
370
371         k_sem_init((struct k_sem *)*lock, 1, 1);
372 #ifdef CONFIG_USERSPACE
373             k_object_access_all_grant(*lock);
374 #endif /* CONFIG_USERSPACE */
375     }
```



EMBEDDED
OPEN SOURCE
SUMMIT



High Severity Bug, 4/4

[zephyr / lib / libc / newlib / libc-hooks.c](#)

[Code](#) [Blame](#) 576 lines (478 loc) · 13.2 KB

```
396     /* Close dynamic non-recursive lock */
397     void __retarget_lock_close(_LOCK_T lock)
398     {
399         __ASSERT_NO_MSG(lock != NULL);
400 #ifndef CONFIG_USERSPACE
401         free(lock);
402 #else
403         k_object_release(lock);
404 #endif /* !CONFIG_USERSPACE */
405 }
```

[zephyr / lib / libc / picolibc / libc-hooks.c](#)

[Code](#) [Blame](#) 253 lines (209 loc) · 5.48 KB

```
110 // Direct conversion from _LOCK_T to void*
111 #define _LOCK_T void *
```



EMBEDDED
OPEN SOURCE
SUMMIT



Key Takeaways

- Zephyr has a good security posture
- Coverity does a pretty good job, but it does not find everything
- Other tools are occasionally useful, but they generate tons of noise
- An active focus should be on tools that can come up with fixes of the problems

