Bro, Where is my code?

Complier can optimize your code out, breaks your error checking logic

The Problem

Pointer overflow: if (p +128 < P)

```
int foo(char *buf, char *buf_end, uint off) {
    if (buf + off >= buf_end) return -1;
   /* compiler optimzes this red line out,
    * results in pointer overflow.
    */
    if (buf + off < buf) return -1;
    /* access buf[0..off-1] */
    return 0;
}    /* seen in kernel, chrome, python */
```

Other Undefined Behavior

Signed integer overflow

if (x + 32 < x)

Oversized shift

If (! (1<< x))

Null pointer dereferece

p->node; if (p)

The Consequency

The irony of the problem is that the carefully written boundary check is NO-OPS after compiler optimization.

Note even without –O in the gcc compiler flags, gcc does optimization and increasingly aggressive.

Usage (1)

repo:5000/mitstack:3 is a docker image built specifically to detect undefined behavior in C language.

Once can launch a container with the build directory such as

%docker run –rm –v /home/mulin:/home/mulin –it repo:5000/mitstack:3

Usage(2)

After lunch the docker image, you can build panos inside the container the same way as before except you need to augment the make with the following:

%/root/foobar/build/bin/stack-build make -i -k

(-i and -k to ignore errors and continue on error)

%/root/foobar/build/bin/poptck

Usage(3)

The results are in YAML format as the following, you can then inspect the source to check whether it is a problem (there are false positives)

```
---
bug: anti-simplify
model: |
  %215 = icmp ne i8* %201, null, !dbg !481
  -->  true
stack:
  - /home/mulin/src/libs/common/base/src/foo.c:260:0
ncore: 1
core:
  - /home/mulin/src/libs/common/base/src/foo.c:255:0
    - pointer overflow
---
```