Bug Report from Guarding

Harish Patil and Charles Fischer

Computer Sciences Dept., University of Wisconsin —Madison

1210 W. Dayton St., Madison, WI 53706, USA.

e-mail: *{patil, fischer}@cs.wisc.edu*

Phone: 608-262-0018

Fax: 608-262-9777

April 2, 1996

We have used our system for pointer and array access checking to test programs from a variety of sources including SunOS 4.1.3 utilities, SPEC 92 benchmarks, and the SPLASH multiprocessor benchmarks. In the following we describe some of the errors we discovered. A utility called *fuzz* [1] was used to generate random input for the SunOs utilities. SPEC92 and SPLASH benchmarks were tested with their reference inputs, unless stated otherwise.

Benchmark: decompress

Source: SPEC92

Error #1: file compress.c

Function `getcode()` called from `decompress()`

```
1144   /* high order bits */
1145   code |= (*bp & rmask[bits]) << r_off;
```

Pointer ‘bp’ is used to traverse global character array `buf[BITS]`. While decompressing the reference input the number of bits per code changes to 16. (This condition can be forced by changing `#define INIT_BITS to 16`). Sometime after this happens, `bp` dereferences one location beyond the array `buf` at line 1145.

Check: Put ‘`assert((bp-&buf[0])<BITS);`’ before line 1145.

The purified program did not report any errors.

Benchmark: sc

Source: SPEC92

Error #1: File `sc.c`, in function `update()`:

```
213      /* Now pick up the counts again */
```

```
214      for (i = stcol, cols = 0, col = RESCOL;
215              (col + fwidth[i]) < COLS-1 && i < MAXCOLS; i++) {
```

An array bound violation occurs in the terminating condition of the for loop.The two operands of `&&` are in the wrong order `i < MAXCOLS` must come before indexing `fwidth[i]`.When `i` becomes `MAXCOLS (40)` there is an array bound violation. (Occurs twice for input.ref/loada2)

Error #2: Found error in file `lex.c`, in function `yylex()`:

```
114      for (tblp = linelim ? experres : statres; tblp->key; tblp++)
115              if (((tblp->key[0]^tokenst[0])&0137)==0
116               && tblp->key[tokenl]==0) {
```

Array `tokenst` contains the current token and `tokenl` is the length of the current token. The for loop traverses a table of reserved words to see if the current token is a reserved word. Pointer `tblp` is used to point to various entries of a table of reserved words.The first part of the if condition checks if the first letter of the current reserved word and the current token are the same. (It uses some clever bit manipulation and properties of the ASCII character set to ignore case differences.) The second part makes sure that the current reserved word is not longer than the current token. For input token `goto`, `tokenl` is `4`. There is a keyword `GET` in the table `statres`. The first part of the if condition is satisfied `((('G'^'g'&0137)==0)` is `TRUE`). For the second check, `tblp->key[tokenl]` accesses the 5th element of the current key `GET` which has length 4 (`'G','E','T','\0'`). Accessing the 5th element of an array of size 4 is clearly illegal.

The purified program misses these two errors. However, it reports 13 errors (uninitialized memory reads) in the libraries `libc` and `libcursses`.

Benchmark: cholesky

Source: SPLASH-1

Error #1: file util.U

Function `ReadSparse5()` defines `char type[3];`

```
166   type[3] = 0
```

Only valid indices are `0-2`. This error has been fixed in the latest release of the SPLASH benchmarks - SPLASH-2. The purified program did not report any error.

Error #2: file util.U

Function `ISort()`:

```
350 while (M.row[j-1] > tmp && j > lo){ ...
```

On some calls to `ISort()` when the value of `lo` is `0`, index of `M.row[]` becomes `-1` which is illegal. The statement should have been:

```
350 while (j > lo && M.row[j-1] > tmp){ ...
```

This error is still there in the latest release of the SPLASH benchmarks — SPLASH-2. The purified program did report this error and it also reported 9 uninitialized memory reads in the library function `sscanf()`.

Benchmark: locus

Source: SPLASH-1

Error #1: file ginput.U

Function `ReadSparse5()`:

```
201 if (NumberOfTerminals < 2){
...
203     free(NewWire);
204 }
```

`free(NewWire)` is followed by a dereference of `NewWire`:

```
222 NewWire->NumberOfGroups = IGrouptNumber;
```

The purified program did report this error.

Error #2: file timer.U

Function `ReportTimes()`

```
161 for(i=0; i <=Global->NumberOfWires; i++){
...
162 if(SegmentUsed[i] != 0) && (NetPinDistribution[i] != 0)){ ...
```

Valid indices for `SegmentUsed[]` and `NetPinDistribution[]` are `0-800`. For one of the reference inputs `Global->NumberOfWires` is `904` and invalid indices of the `SegmentUsed` and `NetPinDistribution` are accessed.

The purified program did report this error and it also reported one uninitialized memory read.

“locus” is not part of SPLASH-2.

Utility: col

Source: SunOs 4.1.3

Error #1: file col.c

```
229     c3 = *line;
```

Inside the program, there are many places where the pointer `line` is incremented and dereferenced without checking its validity.

The purified program does catch this error for some inputs. We have found a random input for which the purified program does not terminate (neither does the original program).

Utility: ul

Source: SunOs 4.1.3

Error #1: file ul.c

There are numerous array bound violations in function `filter()`. Array `obuf` is indexed using the variable `col`. If an input line contains >=512 characters `col` gets larger than the maximum index (511) allowed for `obuf`.

The purified program does catch this error for some inputs. We have constructed a sample input for which the purified program does not report this error.

References

[1] B. P. Miller, F. Lars, and B. So. An empirical study of the reliability of Unix utilities. *Communications of the ACM*, 33(12):32–44, December 1990.