FB Infer Analysis Results

FB Infer consists of five checkers for detecting memory safety issues in C/C++/Java/Objective-C programs: null dereference checker, memory leak checker, use-after-free checker, resource leak checker, and empty vector access checker. We ran all the checkers on Drake. The checkers are fully automatic as memory safety is a generic property independent of the analyzed program’s functionality.

The number of reports produced by each checker is as follows:

|  |  |
| --- | --- |
| Null Dereference | 420 |
| Buffer Overrun | 52 |
| Memory Leak | 74 |
| Use After Free | 7 |
| Resource Leak | 5 |
| Empty Vector Access | 1 |

### Null Dereference Checker

After manual inspection, 138 out of 420 null dereference alarms turned out to be viable. The checker detected a feasible null dereference as well as missing null checks after allocating memory with malloc, strdup, getenv, and fopen.

For brevity, we describe 5 representative alarms as follows.

|  |  |
| --- | --- |
| **Location** | **Description** |
| drake/externals/ipopt/Ipopt/src/LinAlg/IpExpansionMatrix.cpp:371 | The pointer `compressed\_pos\_` is allocated memory only if NRows() > 0 (at line 361 and 362). If NCols() > 0 and NRows() <= 0, then it could be null and is dereferenced at line 371.  ExpansionMatrixSpace(...) :  expanded\_pos\_(NULL),  compressed\_pos\_(NULL) {  358: if (NCols()>0) {  359: expanded\_pos\_ = new Index[NCols()];  360: }  361: if (NRows()>0) {  362: compressed\_pos\_ = new Index[NRows()];  363: }  ...  367: for (Index i=0; i<NCols(); i++) {  ...  370: expanded\_pos\_[i]=ExpPos[i]-offset;  371: **compressed\_pos\_[ExpPos[i]-offset]** = i;  }  } |
| drake/externals/libbot/bot2-lcm-utils/src/tunnel/ldpc/ldpc\_scheme.cpp:306 | The ‘ESIofSymbols’ last assigned on line 284:  284: \*ESIofSymbols = (int\*) calloc(m\_nbSymbolsPerPkt, sizeof(int));  could be null and is dereferenced at line 306, column 5 **(\*ESIofSymbols)** |
| drake/externals/lcm/lcmgen/getopt.c:51 | The ‘arg’ last assigned on line 50 could be null and is dereferenced by call to `strstr()’ at line 51.  50: char \*arg = strdup(argv[i]);  51: char \*eq = **strstr(arg, "=")**; |
| drake/externals/libbot/bot2-procman/src/deputy/procman.c:101 | The `path` last assigned on line 100 could be null and is dereferenced by call to `strlen()` at line 101  100: char \*path = getenv ("PATH");  101: int newpathlen = **strlen (path)** + strlen(params->bin\_path) + 2; |
| drake/externals/ipopt/Ipopt/src/Algorithm/LinearSolvers/IpPardisoSolverInterface.cpp:639 | The pointer `mat\_file` last assigned on line 637 could be null and is dereferenced by call to `fprintf()` at line 639, column 7  637: mat\_file = fopen (mat\_name, "w");  639: **fprintf (mat\_file, "%d\n", N);** |

### Buffer Overflow Checker

After manual inspection, XX out of 52 buffer-overflow alarms turned out to be viable.

### Memory Leak, Use After Free, Resource Leak, and Empty Vector Access Checkers

After manual inspection, all of 87 alarms turned out to be false.