Toybox Bug Analysis

Austin Mordahl

June 15, 2018

File	lsm.h
Line	63
Description	Uninitialized variable: result
Number of Configurations	432[1]

Code Sample

```
static inline char *lsm_context(void)
{
  int ok = 0;
  char *result;

  if (CFG_TOYBOX_SMACK) ok = smack_new_label_from_self(&result) > 0;
  else ok = getcon(&result) == 0;

  return ok ? result : strdup("?");
}
```

Status	False positive.
Remarks	In configurations including `TOYBOX_SMACK` and `TOYBOX_SELINUX` `smack_new_label_from_self` and `getcon` are replaced with the value -1, respectively. In other configurations, `*result` is an out parameter.

[1] The actual cppcheck bug reports listed various C source code files which included this header as the source of the bug, even though `lsm.h` was the actual source. This is the number of total occurrences of the bug across multiple files.

File	base64.c
Line	35
Description	Expression `'this.base64.columns&&++*x == this.base64.columns'` depends on order of evaluation of side effects.
Number of Configurations	478

Code Sample

```
static void wraputchar(int c, int *x)
{
  putchar(c);
  TT.total++;
  if (TT.columns && ++*x == TT.columns) {
    *x = 0;
    xputc('\n');
  };
}
```

Status	False positive.
Remarks	Although `TT.colums` appears twice in the same expression, it is modified neither time. Thus, the order of evaluation of side effects does not matter.