Lynn Gao
CS 5959

1. gcov coverage 4/2/2014:

gcov vfprintf.c
File 'vfprintf.c'
Lines executed:65.40% of 367
vfprintf.c:creating 'vfprintf.c.gcov'

2. uncovered code:

ULONG (145) case ULONG: arg->i = va_arg(*ap, unsigned long);
ULLONG (147) case ULLONG: arg->i = va_arg(*ap, unsigned long long);
USHORT (149) case USHORT: arg->i = (unsigned short)va_arg(*ap, int);

This section wasn't covered because I didn't put in correct values for print types for unsigned long, unsigned long long, and unsigned short

int subtraction and padding (174-178)

```
    70:  170:static void pad(MUSL_FILE *f, char c, int w, int l, int fl)
     -:  171:{
     -:  172:char pad[256];
    70:  173:        if (fl & (LEFT_ADJ | ZERO_PAD) || l >= w) return;
 #####:  174:        l = w - l;
 #####:  175:        memset(pad, c, l>sizeof pad ? sizeof pad : l);
 #####:  176:        for (; l >= sizeof pad; l -= sizeof pad)
 #####:  177:        out(f, pad, sizeof pad);
 #####:  178:        out(f, pad, l);
     -:  179:}
```

I don't completely understand what is going on in this section, but I can see that values are being subtracted from an int so I think that I am not testing some int values properly

long double negation, addition, MARK_POS, PAD_POS (225-229, 246, 255-264)

```
29:#define PAD_POS    (1U<<' '-' ')
30:#define MARK_POS   (1U<<'+'-' ')
```

Areas with addition and negation of long double was uncovered because I might not have been testing negative long doubles, I also wasn't testing MARK_POS and PAD_POS

INF, NAN, MAX, adding integer to a char (233-239)

These areas are uncovered because I didn't incorporate testing INF, NAN, and MAX for ints or chars

adding integer to a char (227, 229, 249-250)

Didn't have a test case adding an int to a char

long double that is a float (246)

I tested with floats and longs but I don't think I tested long doubles (the correct way?)

Not sure (252-253, 268-271, 595-596, 651-652, 672-673)
Rounding precision? (385)

There are some things with pointers and pre-decrementing that I am not covering with my fuzzer and the rest I don't know what the program is trying to do

using type uint32_t and uint64_t (308-313)

My fuzzer did not include testing with types uint32_t and uint64_t

buffer bits?, pointer arithmetic (275-293)
pointer arithmetic (200-202, 315, 317, 336, 379-380, 588)
pointer stuff (685, 689)
stuff with pointers - read field width (499-508)
Field width? (499-508)

There looks like some pointer math happening that I might not be testing or doing at all in my fuzzer

stuff with pointers – read precision (513-520)

My fuzzer isn't handling or testing any pointer arithmetic or printing out pointers

MIN, MAX (385, 389, 570-572)

Wasn't testing MIN or MAX

EOVERFLOW (471-472)

Never tested anything over INT_MAX that would overflow

%% format specifier (485-487)

Didn't test %% format specifier

NOARG (537)

Not testing NOARG or validity of argument type, how do you do this?

BARE, LPRE, LLPRE, HPRE, HHPRE, ZTPRE, JPRE (559-572)

I didn't test any of these cases, it seems like stuff with pointers of types int, long, long long, unsigned short, unsigned char, size_t, and uintmax_t

INTMAX_MAX (584)

Not testing INTMAX_MAX or if any values ever was greater than it

m sterror(errno) (605)

For type "m" I am not testing if its value is a strerror(errno)

NL_ARGMAX (654-658)

I wasn’t testing values up to NL_ARGMAX, also I am not sure what this is

flock(f) (678-681, 685-689)

I am not testing FLOCK(f), looks like pointer stuff