Bip - Bug #643

Build failure with GCC 9: print buffer sizes for data from localtime

2019-01-25 11:05 - Adam Williamson

Status:	New	**Start date:**	2019-01-25
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Patch Available:		**Branch:**	
Found in Versions:		**Security:**	
Confirmed:	No	**Help Needed:**	

Description

So the Red Hat compiler team very kindly pre-checked build of a bunch of Fedora packages with GCC 9 (which will be showing up in Fedora Rawhide soon), and they flagged up two issues in Bip which will cause it to fail to build with GCC 9. This is the second issue (well, set of similar and related issues).

I'll just paste Jeff Law's explanation here, because he explains it much better than I could:

The second set of problems are all closely related in this code:

```
char *logfile, year[5], day[3], month[3], hour[3], *tmp, *logdir;
struct tm *now;
time_t s;
char *dest = bip_strdup(destination);

strtolower(dest);
logfile = (char *)bip_malloc(MAX_PATH_LEN + 1);

time(&s);
now = localtime(&s);
snprintf(year, 5, "%04d", now->tm_year + 1900);
snprintf(day, 3, "%02d", now->tm_mday);
snprintf(month, 3, "%02d", now->tm_mon + 1);
snprintf(hour, 3, "%02d", now->tm_hour);
snprintf(logfile, MAX_PATH_LEN, "%s/%s", conf_log_root,
conf_log_format);
replace_var(logfile, "%u", logdata->user->name, MAX_PATH_LEN);
replace_var(logfile, "%n", logdata->network, MAX_PATH_LEN);
replace_var(logfile, "%c", dest, MAX_PATH_LEN);
replace_var(logfile, "%Y", year, MAX_PATH_LEN);
replace_var(logfile, "%d", day, MAX_PATH_LEN);
replace_var(logfile, "%m", month, MAX_PATH_LEN);
replace_var(logfile, "%h", hour, MAX_PATH_LEN);
```

Nothing guarantees the resulting ranges for the data returned by localtime. So the compiler has to make some worst case assumptions for how many bytes will be necessary to print now->tm_year and other fields.

So our first recommendation would be to verify the resulting now->tm_year has a range that allows it to be printed with just 4 bytes of data. Alternately you could increase the size of the buffer.

THere's similar issues with the tm->mon field. While the API of localtime guarantees that it'll be in the range [0, 11] there's no way for the compiler to know tm->mon is constrained in that way. The fix would be similar. Verify the range of tm->mon or increase the size of the buffer.

History

#1 - 2019-02-10 18:16 - Dan Horák

The problem goes away with commit 814d54c6 (switching to strftime).

#2 - 2019-02-11 23:44 - Adam Williamson

It does indeed, confirmed. Please do go ahead and close this.