tool for memory leak detection

Simon Guo

Agenda

- Problem to be solved
- Tool usage/demonstration
- Tool implementation

1. Problem to be solved

- An OamCfg PR complains memory leak during weekend stability run since old release. The memory increases with a slow manner– about 1M per 1~2 hours.

- This issue resides at both Solaris platform and linux platform. Solaris platform memory increase will be much slower and be harder to debug.

- Exiting linux memory leak detecting tool in glibc – mtrace, it only prints 1 level caller address. The address always indicated some place in library, which has little value to help with debug.

- The practice to try a compiled glibc with enhancement mtrace(to print more levels) is unreasonably and miserably successful.

- Need a lightweight linux tool to dump memory leak with enough function trace level. No source code modification is allowed.

2. Tool usage - 1

- - It is a library called: libsmtrace.so
- Step 1) Export $MALLOC_TRACE environment varaible to some filename.
- Step 2) launch target program with $LD_PRELOAD=/usr/lib/libsmtrace.so
- Step 3) Turn on trace via configuration port 22222:
  - nc –u 127.0.0.1 22222
  - o
- Ctrl + c to quit
- Step 4) run some testing
- Step 5) Turn off trace
  - nc –u 127.0.0.1 22222
  - c
- Ctrl + c to quit
- Step 6) Check output at $MALLOC_TRACE file

- My Ubuntu machine: /mnt/oldBoot/home/simon/test/mtracestub

2. Tool usage - 2

- $MALLOC_TRACE file will be like:

- *@|5807| [0x80484bd] |0x08048501 0xb75d8935 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000| + 0x9817098 0x400*

- *@|5807| [0x80484d6] |0x08048524 0xb75d8935 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000| - 0x9817098*

- @|5807| - process id 5807
- "+ 0x9817098 0x400" - "+" means allocation. The memory chunk starts at + 0x9817098, and with size 0x400.
- "- 0x9817098 " - "-" means deallocation. The memory chunk at 0x9817098 is deallocated.
- "[0x80484bd] |0x08048501 0xb75d8935 " - function back trace.

2. Tool usage – parse result

- It is easy to write some tool to decode the tool output. I prefer to use awk for this decode tool:
  - simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ ./myawk ./nn.log
  - Old deallocate(no allocate record) 0x8a4a098:[0x80484f8]
  - Leak records:
  - @|4575| [0x80484d5] |0xb75d0935 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000| + 0x8a4a098 0x400
  - Summary:1 leaks. allocated=10 deallocated=9 standalone_deallocated=1, dup_allocated=0

3. Library implementation

- $LD_PRELOAD(from man page)

- *A whitespace-separated list of additional, user-specified, ELF shared libraries to be loaded before all others. This can be used to selectively override functions in other shared libraries.*

- *Each library has an "init" function, which will be executed when library is loaded. In libsmtrace.so, a thread is created to listen on port 22222, in the init function.*

- *If user turns on the trace with "o" command, the thread function will replace glibc memory allocation hooks with self-defined memory hook functions.* When the targeted process invokes malloc(), it will actually call self-deinfed memory hook functions. The memory hook functions will dump additional function traces and call standard glibc memory API, like malloc().

- If user turn off the trace with "c" command, those hooks will be removed.

- A simple awk script will be able to parse the result.

3. Library implementation - 2

```
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ cat setpre
export LD_PRELOAD=$PWD/libsmtrace.so
export MALLOC_TRACE=$PWD/nn.log
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ . setpre
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ ./test &
[1] 2073
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ Memory leak
detection lib has been attached successfully


simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ . nosetpre
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ cat nosetpre
export LD_PRELOAD=
```

```
simon@thunderCat:/mnt/oldBoot/home/simon/test/mtracestub$ nc -u
127.0.0.1 22222
```

- Questions