NAME

RedZone - buffer corruptions detector

SYNOPSIS

options KDB
options DDB
options DEBUG_REDZONE

DESCRIPTION

RedZone detects buffer underflow and buffer overflow bugs at runtime. Currently **RedZone** only detects buffer corruptions for memory allocated with malloc(9). When such corruption is detected two backtraces are printed on the console. The first one shows from where memory was allocated, the second one shows from where memory was freed. By default the system will not panic when buffer corruption is detected. This can be changed by setting the *vm.redzone.panic* sysctl(8) variable to 1. The amount of extra memory allocated for **RedZone**'s needs is stored in the *vm.redzone.extra_mem* sysctl(8) variable.

EXAMPLE

The example below shows the logs from the detection of a buffer underflow and a buffer overflow.

```
REDZONE: Buffer underflow detected. 2 bytes corrupted before 0xc8688580 (16 bytes allocated).
Allocation backtrace:
#0 0xc0583e4e at redzone_setup+0x3c
#1 0xc04a23fa at malloc+0x19e
#2 0xcdeb69ca at redzone_modevent+0x60
#3 0xc04a3f3c at module_register_init+0x82
#4 0xc049d96a at linker_file_sysinit+0x8e
#5 0xc049dc7c at linker_load_file+0xed
#6 0xc04a041f at linker_load_module+0xc4
#7 0xc049e883 at kldload+0x116
#8 0xc05d9b3d at syscall+0x325
#9 0xc05c944f at Xint0x80_syscall+0x1f
Free backtrace:
#0 0xc0583f92 at redzone_check+0xd4
#1 0xc04a2422 at free+0x1c
#2 0xcdeb69a6 at redzone_modevent+0x3c
#3 0xc04a438d at module_unload+0x61
#4 0xc049e0b3 at linker_file_unload+0x89
#5 0xc049e979 at kern_kldunload+0x96
#6 0xc049ea00 at kldunloadf+0x2c
```

```
#7 0xc05d9b3d at syscall+0x325
#8 0xc05c944f at Xint0x80_syscall+0x1f

REDZONE: Buffer overflow detected. 4 bytes corrupted after 0xc8688590 (16 bytes allocated).
Allocation backtrace:
#0 0xc0583e4e at redzone_setup+0x3c
#1 0xc04a23fa at malloc+0x19e
#2 0xcdeb69ca at redzone_modevent+0x60
#3 0xc04a3f3c at module_register_init+0x82
#4 0xc049d96a at linker_file_sysinit+0x8e
#5 0xc049dc7c at linker_load_file+0xed
#6 0xc04a041f at linker_load_module+0xc4
#7 0xc049e883 at kldload+0x116
#8 0xc05d9b3d at syscall+0x325
#9 0xc05c944f at Xint0x80_syscall+0x1f
Free backtrace:
#0 0xc0584020 at redzone_check+0x162
#1 0xc04a2422 at free+0x1c
#2 0xcdeb69a6 at redzone_modevent+0x3c
#3 0xc04a438d at module_unload+0x61
#4 0xc049e0b3 at linker_file_unload+0x89
#5 0xc049e979 at kern_kldunload+0x96
#6 0xc049ea00 at kldunloadf+0x2c
#7 0xc05d9b3d at syscall+0x325
#8 0xc05c944f at Xint0x80_syscall+0x1f
```

SEE ALSO

sysctl(8), malloc(9), memguard(9)

HISTORY

RedZone first appeared in FreeBSD 7.0.

AUTHORS

Pawel Jakub Dawidek <*pjd@FreeBSD.org*>

BUGS

Currently, **RedZone** does not cooperate with memguard(9). Allocations from a memory type controlled by memguard(9) are simply skipped, so buffer corruptions will not be detected there.