XNU 审计笔记

Brightiup

2018-03-28

目录

1 系统调用

1.1 系统调用种类以及入口

xnu 系统调用总共有四个种类,分别为 *unix*、*mach*、*machdep* 和 *diagnostics*,对应的处理函数入口分别为 *hndl_unix_scall64*、*hndl_mach_scall64*、*hndl_mdep_scall64* 和 *hndl_diag_scall64*。通过 *syscall* 指令可以直接进入这几个函数。*syscall* 指令的处理入口为 *hndl_syscall*,其实现如下:

```
// xnu/osfmk/x86_64/idt64.s
/*
 * 64bit Tasks
 * System call entries via syscall only:
 *
 *  r15  x86_saved_state64_t
 *  rsp  kernel stack
 *
 *  both rsp and r15 are 16—byte aligned
 *  interrupts disabled
 *  direction flag cleared
 */

Entry(hndl_syscall)
    TIME_TRAP_UENTRY

    movq    %gs:CPU_ACTIVE_THREAD,%rcx  /* get current thread     */
    movl    $—1, TH_IOTIER_OVERRIDE(%rcx)   /* Reset IO tier override to —1
     before handling syscall */
    movq    TH_TASK(%rcx),%rbx      /* point to current task  */

    /* Check for active vtimers in the current task */
    TASK_VTIMER_CHECK(%rbx,%rcx)

    /*
     * We can be here either for a mach, unix machdep or diag syscall,
     * as indicated by the syscall class:
     */
    movl    R64_RAX(%r15), %eax     /* syscall number/class */
    movl    %eax, %edx
    andl    $(SYSCALL_CLASS_MASK), %edx /* syscall class */
    cmpl    $(SYSCALL_CLASS_MACH<<SYSCALL_CLASS_SHIFT), %edx
    je  EXT(hndl_mach_scall64)
    cmpl    $(SYSCALL_CLASS_UNIX<<SYSCALL_CLASS_SHIFT), %edx
    je  EXT(hndl_unix_scall64)
```

```
    cmpl    $(SYSCALL_CLASS_MDEP<<SYSCALL_CLASS_SHIFT), %edx
    je  EXT(hndl_mdep_scall64)
    cmpl    $(SYSCALL_CLASS_DIAG<<SYSCALL_CLASS_SHIFT), %edx
    je  EXT(hndl_diag_scall64)

    /* Syscall class unknown */
    sti
    CCALL3(i386_exception, $(EXC_SYSCALL), %rax, $1)
    /* no return */
```

其中几种系统调用的分类序号为：

```
// xnu/osfmk/mach/i386/syscall_sw.h
#define SYSCALL_CLASS_NONE 0 /* Invalid */
#define SYSCALL_CLASS_MACH 1 /* Mach */
#define SYSCALL_CLASS_UNIX 2 /* Unix/BSD */
#define SYSCALL_CLASS_MDEP 3 /* Machine-dependent */
#define SYSCALL_CLASS_DIAG 4 /* Diagnostics */
#define SYSCALL_CLASS_IPC 5  /* Mach IPC */
```

系统调用号为 *SYSCALL_CLASS « 24 | dispatch_num* 的组合，例如 *SYSCALL_CLASS* 为 3，*dispatch_num* 为 0 则将进入 *hv_task_trap* 并最终进入 AppleHV.kext 对应的处理函数，示例如下：

```
asm ("mov $0x03000000, %rax; mov $0x04, %rdi; syscall")
```