Debugging Kernel OOPs

Neependra Khare

STEC India

November 3, 2011

Agenda

- What is OOPs
- Typical OOPs
- Kernel Symbol Table
- Examples
- Debugging a running Kernel with gdb

What is OOPs

- What's the most common Bug in user space program.

What is OOPs

- What's the most common Bug in user space program.
  - The segfault.

What is OOPs

- What's the most common Bug in user space program.
  - The segfault.
- What is the most common bug in the Linux kernel?

What is OOPs

- What's the most common Bug in user space program.
  - The segfault.
- What is the most common bug in the Linux kernel?
  - The segfault.
  - Except here:-
    - The notion of a segfault is much more complicated
    - When the kernel de-references an invalid pointer, it's not called a segfault – it's called an "oops".

Typical OOPs

Typical OOPs

- Kernel exception handler
  - Kills offending process
  - Prints registers, stack trace with symbolic info

Typical OOPs

- Kernel exception handler
  - Kills offending process
  - Prints registers, stack trace with symbolic info
- Some exceptions non-recoverable (panic())
  - Prints message on console, halts kernel
  - Oops in interrupt handler, idle (0) or init (1)

Typical OOPs

- Kernel exception handler
  - Kills offending process
  - Prints registers, stack trace with symbolic info
- Some exceptions non-recoverable (panic())
  - Prints message on console, halts kernel
  - Oops in interrupt handler, idle (0) or init (1)
- Oops generated by macros:
  - BUG(), BUG_ON(condition)

Typical OOPs

```
port_pc lp parport psmouse serio_raw pcspkr i2c_piix4 i2c_core evdev ext3 jbd mb
cache sg sr_mod cdrom sd_mod ata_piix pata_acpi ata_generic libata ne2k_pci 8390
 scsi_mod dock thermal_sys fuse
<4>[ 3281.830718]
<4>[ 3281.830830] Pid: 4389, comm: cat Not tainted (2.6.26-rc5 #1)
<4>[ 3281.830954] EIP: 0060:[<d08ae035>] EFLAGS: 00000292 CPU: 0
<4>[ 3281.831118] EIP is at read_proc+0x35/0x50 [hello]
<4>[ 3281.831235] EAX: 0000000a EBX: 00000000 ECX: fffffffb EDX: cb3b5f38
<4>[ 3281.831362] ESI: 00000000 EDI: 00000400 EBP: cb3b5f04 ESP: cb3b5ef4
<4>[ 3281.831489]  DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
<0>[ 3281.831626] Process cat (pid: 4389, ti=cb3b4000 task=ccc2f0c0 task.ti=cb3b
4000)
<0>[ 3281.831759] Stack: ccbcf000 d08ae0ef 00000000 d08ae000 cb3b5f48 c01cab24 0
0000400 cb3b5f38
<0>[ 3281.832153]        00000000 00000400 00000000 08051000 ccbcf000 00000000 0
0000400 cc4d9a80
<0>[ 3281.832526]        00000000 00000001 cc4d9a80 fffffffb c01ca980 cb3b5f6c c
01c626d cb3b5f9c
<0>[ 3281.832929] Call Trace:
<0>[ 3281.832972]  [<d08ae000>] ? read_proc+0x0/0x50 [hello]
<0>[ 3281.832972]  [<c01cab24>] ? proc_file_read+0x1a4/0x2a0
<0>[ 3281.832972]  [<c01ca980>] ? proc_file_read+0x0/0x2a0
<0>[ 3281.832972]  [<c01c626d>] ? proc_reg_read+0x5d/0x90
<0>[ 3281.832972]  [<c018ee74>] ? vfs_read+0x94/0x160
[0]more> _
```

Tainted kernels

Some oops reports contain the string 'Tainted: ' after the program counter.

- 'G' if all modules loaded have a GPL or compatible license,
- 'P' any proprietary module has been loaded.
- 'F' if any module was force loaded by "insmod -f", ' ' if all modules were loaded normally.
- 'M' if any processor has reported a Machine Check Exception, ' ' if no Machine Check Exceptions have occurred.
- etc.

Kernel Symbol Table

```
c04a1210 T vfs_readv
c04a1525 T vfs_read
c04a7a18 T vfs_readlink
c04ac37a T vfs_readdir
...
c068b262 t net_rx_action
c068c4a0 T netif_rx
c068d01f T netif_rx_ni
c068d055 T netdev_rx_csum_fault
c0698a01 T __netpoll_rx
c0853400 r __ksymtab_netif_rx
```

Kernel Symbol Table

```
c04a1210 T vfs_readv
c04a1525 T vfs_read
c04a7a18 T vfs_readlink
c04ac37a T vfs_readdir
...
c068b262 t net_rx_action
c068c4a0 T netif_rx
c068d01f T netif_rx_ni
c068d055 T netdev_rx_csum_fault
c0698a01 T __netpoll_rx
c0853400 r __ksymtab_netif_rx
```

- System.map

Kernel Symbol Table

```
c04a1210 T vfs_readv
c04a1525 T vfs_read
c04a7a18 T vfs_readlink
c04ac37a T vfs_readdir
...
c068b262 t net_rx_action
c068c4a0 T netif_rx
c068d01f T netif_rx_ni
c068d055 T netdev_rx_csum_fault
c0698a01 T __netpoll_rx
c0853400 r __ksymtab_netif_rx
```

- System.map
- /proc/kallsyms

System.map File

System.map is a "phone directory" list of function in a particular build of a kernel.

```
[root@nkhare ~]# ls -l /boot/System.map*
lrwxrwxrwx  1 root root      25 2009-08-26 17:05 /boot/System.map -> /boot/System.map-2.6.29.5
-rw-r--r--. 1 root root 1274567 2009-05-28 03:09 /boot/System.map-2.6.29.4-167.fc11.i686.PAE
-rw-r--r--  1 root root 1225104 2009-08-26 17:05 /boot/System.map-2.6.29.5
-rw-r--r--  1 root root 1257149 2009-06-17 09:02 /boot/System.map-2.6.29.5-191.fc11.i586
```

How it is produced?

System.map File

System.map is a "phone directory" list of function in a particular build of a kernel.

```
[root@nkhare ~]# ls -l /boot/System.map*
lrwxrwxrwx  1 root root      25 2009-08-26 17:05 /boot/System.map -> /boot/System.map-2.6.29.5
-rw-r--r--. 1 root root 1274567 2009-05-28 03:09 /boot/System.map-2.6.29.4-167.fc11.i686.PAE
-rw-r--r--  1 root root 1225104 2009-08-26 17:05 /boot/System.map-2.6.29.5
-rw-r--r--  1 root root 1257149 2009-06-17 09:02 /boot/System.map-2.6.29.5-191.fc11.i586
```

How it is produced?

- When you compile the kernel.

System.map File

System.map is a "phone directory" list of function in a particular build of a kernel.

```
[root@nkhare ~]# ls -l /boot/System.map*
lrwxrwxrwx  1 root root      25 2009-08-26 17:05 /boot/System.map -> /boot/System.map-2.6.29.5
-rw-r--r--. 1 root root 1274567 2009-05-28 03:09 /boot/System.map-2.6.29.4-167.fc11.i686.PAE
-rw-r--r--  1 root root 1225104 2009-08-26 17:05 /boot/System.map-2.6.29.5
-rw-r--r--  1 root root 1257149 2009-06-17 09:02 /boot/System.map-2.6.29.5-191.fc11.i586
```

How it is produced?

- When you compile the kernel.
- nm vmlinux

/proc/kallsyms

/proc/kallsyms

- Created on the fly when a kernel boots up.

/proc/kallsyms

- Created on the fly when a kernel boots up.
- Kernel data which is given the illusion of being a disk file.
  # file /proc/kallsyms
  /proc/kallsyms: empty
- Symbol for custom modules

```
# grep mycdrv /proc/kallsyms
f7d14000 t mycdrv1_read [char_read_write]
f7d14014 t mycdrv1_release [char_read_write]
f7d1402f t mycdrv1_open [char_read_write]
f7d140ec r mycdrv1_fops [char_read_write]
f7d1404a t mycdrv1_write [char_read_write]
```

OOPs Example 1

```
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: BUG: unable to handle kernel NULL pointer dereference at (null)
Aug 28 21:22:55 localhost kernel: IP: [<f7d18008>] mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: *pde = 5f136067

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:Oops: 0002 [#1] SMP
Aug 28 21:22:55 localhost kernel: Oops: 0002 [#1] SMP

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0C09:00/PNP0C0A:00/power_su
Aug 28 21:22:55 localhost kernel: last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0
Aug 28 21:22:55 localhost kernel: Modules linked in: char_read_write ppp_synctty n_hdlc ppp_deflate zlib_deflat
Aug 28 21:22:55 localhost kernel: i2c_core video output [last unloaded: scsi_wait_scan]
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: Pid: 4716, comm: cat Tainted: G        W  (2.6.29.5-191.fc11.i586 #1) 7735AE7
Aug 28 21:22:55 localhost kernel: EIP: 0060:[<f7d18008>] EFLAGS: 00010286 CPU: 0
Aug 28 21:22:55 localhost kernel: EIP is at mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: EAX: f3cb9d80 EBX: f3cb9d80 ECX: 00008000 EDX: 0861a000
Aug 28 21:22:55 localhost kernel: ESI: 0861a000 EDI: f7d18000 EBP: f1a67f5c ESP: f1a67f5c
Aug 28 21:22:55 localhost kernel: DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
```

OOPs Example 1

```
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel:   BUG: unable to handle kernel NULL pointer dereference at (null)
Aug 28 21:22:55 localhost kernel: IP: [<f7d18008>]   mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: *pde = 5f136067

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:Oops: 0002 [#1] SMP
Aug 28 21:22:55 localhost kernel: Oops: 0002 [#1] SMP

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0C09:00/PNP0C0A:00/power_su
Aug 28 21:22:55 localhost kernel: last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0
Aug 28 21:22:55 localhost kernel: Modules linked in: char_read_write ppp_synctty n_hdlc ppp_deflate zlib_deflat
Aug 28 21:22:55 localhost kernel: i2c_core video output [last unloaded: scsi_wait_scan]
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: Pid: 4716, comm: cat Tainted: G        W  (2.6.29.5-191.fc11.i586 #1) 7735AE7
Aug 28 21:22:55 localhost kernel: EIP: 0060:[<f7d18008>] EFLAGS: 00010286 CPU: 0
Aug 28 21:22:55 localhost kernel: EIP is at mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: EAX: f3cb9d80 EBX: f3cb9d80 ECX: 00008000 EDX: 0861a000
Aug 28 21:22:55 localhost kernel: ESI: 0861a000 EDI: f7d18000 EBP: f1a67f5c ESP: f1a67f5c
Aug 28 21:22:55 localhost kernel: DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
```

OOPs Example 1

```
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: BUG: unable to handle kernel NULL pointer dereference at (null)
Aug 28 21:22:55 localhost kernel: IP: [<f7d18008>] mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: *pde = 5f136067

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:Oops: 0002 [#1] SMP
Aug 28 21:22:55 localhost kernel: Oops: 0002 [#1] SMP

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0C09:00/PNP0C0A:00/power_su
Aug 28 21:22:55 localhost kernel: last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0
Aug 28 21:22:55 localhost kernel: Modules linked in: char_read_write ppp_synctty n_hdlc ppp_deflate zlib_deflat
Aug 28 21:22:55 localhost kernel: i2c_core video output [last unloaded: scsi_wait_scan]
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel:  Pid: 4716, comm: cat Tainted: G        W  (2.6.29.5-191.fc11.i586 #1) 7735AE
Aug 28 21:22:55 localhost kernel: EIP: 0060:[<f7d18008>] EFLAGS: 00010286 CPU: 0
Aug 28 21:22:55 localhost kernel: EIP is at mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: EAX: f3cb9d80 EBX: f3cb9d80 ECX: 00008000 EDX: 0861a000
Aug 28 21:22:55 localhost kernel: ESI: 0861a000 EDI: f7d18000 EBP: f1a67f5c ESP: f1a67f5c
Aug 28 21:22:55 localhost kernel: DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
```

OOPs Example 1

```
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: BUG: unable to handle kernel NULL pointer dereference at (null)
Aug 28 21:22:55 localhost kernel: IP: [<f7d18008>] mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: *pde = 5f136067

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:Oops: 0002 [#1] SMP
Aug 28 21:22:55 localhost kernel: Oops: 0002 [#1] SMP

Message from syslogd@localhost at Aug 28 21:22:55 ...
 kernel:last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0C09:00/PNP0C0A:00/power_su
Aug 28 21:22:55 localhost kernel: last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0
Aug 28 21:22:55 localhost kernel: Modules linked in: char_read_write ppp_synctty n_hdlc ppp_deflate zlib_deflat
Aug 28 21:22:55 localhost kernel: i2c_core video output [last unloaded: scsi_wait_scan]
Aug 28 21:22:55 localhost kernel:
Aug 28 21:22:55 localhost kernel: Pid: 4716, comm: cat Tainted: G        W  (2.6.29.5-191.fc11.i586 #1) 7735AE7
Aug 28 21:22:55 localhost kernel: EIP: 0060:[<f7d18008>] EFLAGS: 00010286 CPU: 0
Aug 28 21:22:55 localhost kernel: EIP is at mycdrv1_read+0x8/0x14 [char_read_write]
Aug 28 21:22:55 localhost kernel: EAX: f3cb9d80 EBX: f3cb9d80 ECX: 00008000 EDX: 0861a000
Aug 28 21:22:55 localhost kernel: ESI: 0861a000 EDI: f7d18000 EBP: f1a67f5c ESP: f1a67f5c
Aug 28 21:22:55 localhost kernel: DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
```

OOPs Example 1

```
\# objdump  -S char-read-write.ko

static ssize_t mycdrv1_read (struct file *file, char __user * buf, size_t lbuf, loff_t * ppos)
{
   0: 55                    push   \%ebp
   1: 89 e5                 mov    \%esp,\%ebp
   3: e8 fc ff ff ff        call   4 <mycdrv1_read+0x4>
    *(int *)0 = 0;
   8: c7 05 00 00 00 00 00  movl   \$0x0,0x0
   f: 00 00 00
}
  12: 5d                    pop    \%ebp
  13: c3                    ret

00000014 <mycdrv1_release>:
    printk (" Opening : \%s:\n\n", MYDEV_NAME);
    return 0;
}
```

OOPs Example 2

```
Aug 28 21:40:59 localhost kernel: BUG: unable to handle kernel paging request at ffffffff
Aug 28 21:40:59 localhost kernel: IP: [<ffffffff>] 0xffffffff
Aug 28 21:40:59 localhost kernel: *pde = 00957067 *pte = 00000000

Message from syslogd@localhost at Aug 28 21:40:59 ...
 kernel:Oops: 0000 [#3] SMP
Aug 28 21:40:59 localhost kernel: Oops: 0000 [#3] SMP

Message from syslogd@localhost at Aug 28 21:40:59 ...
 kernel:last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0C09:00/PNP0C0A:00/power_su
Aug 28 21:40:59 localhost kernel: last sysfs file: /sys/devices/LNXSYSTM:00/device:00/PNP0A08:00/device:01/PNP0
Aug 28 21:40:59 localhost kernel: Modules linked in: char_read_write1 char_read_write ppp_synctty n_hdlc ppp_de
Aug 28 21:40:59 localhost kernel: drm i2c_algo_bit i2c_core video output [last unloaded: char_read_write1]
Aug 28 21:40:59 localhost kernel:
Aug 28 21:40:59 localhost kernel: Pid: 6050, comm: cat Tainted: G      D W  (2.6.29.5-191.fc11.i586 #1) 7735AE7
Aug 28 21:40:59 localhost kernel: EIP: 0060:[<ffffffff>] EFLAGS: 00010246 CPU: 0
Aug 28 21:40:59 localhost kernel: EIP is at 0xffffffff
Aug 28 21:40:59 localhost kernel: EAX: 00000001 EBX: ffffffff ECX: 00000000 EDX: ffffffff
Aug 28 21:40:59 localhost kernel: ESI: 09282000 EDI: ffffffff EBP: ffffffff ESP: f3cebf64
Aug 28 21:40:59 localhost kernel: DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
```

Debugging a running Kernel with gdb

```
\# gdb vmlinux /proc/kcore

(gdb) p jiffies_64
\$1 = 4295075081

(gdb)  core-file /proc/kcore

(gdb)  p jiffies_64
\$2 = 4295225679
```

Debugging a running Kernel with gdb

```
(gdb) p mycdrv1_read
No symbol "mycdrv1_read" in current context.
```

- Why?
- We need to educate about the module symbols

Debugging a running Kernel with gdb

cd /sys/module/chra_read_write/sections
cat .bss .data .text
0xf8e1e830
0xf8e1e6f4
0xf8e1e000

```
(gdb)  add-symbol-file /home/nkhare/Training/LinuxKernel/oops/char-dev-read-write-oop/char-read-write.ko 0xf8e1
add symbol table from file "/home/nkhare/Training/LinuxKernel/oops/char-dev-read-write-oop/char-read-write.ko"
.text_addr = 0xf8e1e000
.data_addr = 0xf8e1e6f4
.bss_addr = 0xf8e1e830
(y or n) y
Reading symbols from /home/nkhare/Training/LinuxKernel/oops/char-dev-read-write-oop/char-read-write.ko...done.
(gdb) p mycdrv1_read
\$1 = {ssize_t (struct file *, char *, size_t, loff_t *)} 0xf8e1e000 <mycdrv1_read>
(gdb) list mycdrv1_read
```

References

http://www.faqs.org/docs/Linux-HOWTO/Kernel-HOWTO.html
http://www.cs.fsu.edu/~baker/devices/notes/
http://linux.com/learn/linux-training