Subject: Re: [PATCH] Report that kernel is tainted if there were an OOPS before (v2)
Posted by Jan Engelhardt on Thu, 31 May 2007 19:50:50 GMT
View Forum Message <> Reply to Message

This seems to have died and remain unmerged as of yet... resubmit?

On Apr 17 2007 12:58, Pavel Emelianov wrote:

>Date: Tue, 17 Apr 2007 12:58:47 +0400
>From: Pavel Emelianov <xemul@sw.ru>
>To: Andrew Morton <akpm@osdl.org>,
>   Linux Kernel Mailing List <linux-kernel@vger.kernel.org>,
>   linux-arch@vger.kernel.org
>Cc: randy.dunlap@oracle.com, devel@openvz.org
>Subject: [PATCH] Report that kernel is tainted if there were an OOPS before
>   (v2)
>
>If the kernel OOPS-ed or BUG-ed then it probably should
>considered as tainted. Use die_counter introduced by many
>architectures to determine whether or not the kernel died.
>
>This saves a lot of time explaining oddities in the
>calltrace seen via SysRq-P.
>
>Signed-off-by: Pavel Emelianov <xemul@openvz.org>
>Signed-off-by: Kirill Korotaev <dev@openvz.org>
>Cc: Randy Dunlap <randy.dunlap@oracle.com>
>
>---
>
>diff --git a/Documentation/oops-tracing.txt b/Documentation/oops-tracing.txt
>index ea55ea8..dddbfe0 100644
>--- a/Documentation/oops-tracing.txt
>+++ b/Documentation/oops-tracing.txt
>@@ -240,6 +240,10 @@ characters, each representing a particul
>   7: 'U' if a user or user application specifically requested that the
>   Tainted flag be set, ' ' otherwise.
>
>+ 8: 'D' if a kernel has died recently, i.e. there was an OOPS or BUG
>+   before, ' ' otherwise. This is useful when seeing the calltrace
>+   from SysRq-P output.
>+
> The primary reason for the 'Tainted: ' string is to tell kernel
> debuggers if this is a clean kernel or if anything unusual has
> occurred. Tainting is permanent: even if an offending module is

```
>diff --git a/arch/arm/kernel/traps.c b/arch/arm/kernel/traps.c
>index b3ffba1..764526c 100644
>--- a/arch/arm/kernel/traps.c
>+++ b/arch/arm/kernel/traps.c
>@@ -207,7 +207,6 @@ void show_stack(struct task_struct *tsk,
> static void __die(const char *str, int err, struct thread_info *thread, struct pt_regs *regs)
> {
>  struct task_struct *tsk = thread->task;
>- static int die_counter;
>
>  printk("Internal error: %s: %x [#%d]\n", str, err, ++die_counter);
>  print_modules();
>diff --git a/arch/avr32/kernel/traps.c b/arch/avr32/kernel/traps.c
>index c8b7153..0829ebc 100644
>--- a/arch/avr32/kernel/traps.c
>+++ b/arch/avr32/kernel/traps.c
>@@ -156,8 +156,6 @@ static DEFINE_SPINLOCK(die_lock);
>
> void NORET_TYPE die(const char *str, struct pt_regs *regs, long err)
> {
>- static int die_counter;
>-
>  console_verbose();
>  spin_lock_irq(&die_lock);
>  bust_spinlocks(1);
>diff --git a/arch/i386/kernel/traps.c b/arch/i386/kernel/traps.c
>index 35d1c38..8531901 100644
>--- a/arch/i386/kernel/traps.c
>+++ b/arch/i386/kernel/traps.c
>@@ -438,7 +438,6 @@ void die(const char * str, struct pt_reg
>   .lock_owner =  -1,
>   .lock_owner_depth = 0
>  };
>- static int die_counter;
>  unsigned long flags;
>
>  oops_enter();
>diff --git a/arch/ia64/kernel/traps.c b/arch/ia64/kernel/traps.c
>index 5bfb8be..7a1e8d8 100644
>--- a/arch/ia64/kernel/traps.c
>+++ b/arch/ia64/kernel/traps.c
>@@ -47,7 +47,6 @@ die (const char *str, struct pt_regs *re
>   .lock_owner =  -1,
>   .lock_owner_depth = 0
>  };
>- static int die_counter;
>  int cpu = get_cpu();
>
```

> if (die.lock_owner != cpu) {
>diff --git a/arch/mips/kernel/traps.c b/arch/mips/kernel/traps.c
>index 27c53f1..2c65120 100644
>--- a/arch/mips/kernel/traps.c
>+++ b/arch/mips/kernel/traps.c
>@@ -312,7 +312,6 @@ static DEFINE_SPINLOCK(die_lock);
>
> NORET_TYPE void ATTRIB_NORET die(const char * str, struct pt_regs * regs)
> {
>- static int die_counter;
> #ifdef CONFIG_MIPS_MT_SMTC
> unsigned long dvpret = dvpe();
> #endif /* CONFIG_MIPS_MT_SMTC */
>diff --git a/arch/powerpc/kernel/traps.c b/arch/powerpc/kernel/traps.c
>index bf6445a..13bc4c7 100644
>--- a/arch/powerpc/kernel/traps.c
>+++ b/arch/powerpc/kernel/traps.c
>@@ -105,7 +105,6 @@ int die(const char *str, struct pt_regs
>   .lock_owner = -1,
>   .lock_owner_depth = 0
> };
>- static int die_counter;
> unsigned long flags;
>
> if (debugger(regs))
>diff --git a/arch/ppc/kernel/traps.c b/arch/ppc/kernel/traps.c
>index 810f7aa..3a058fb 100644
>--- a/arch/ppc/kernel/traps.c
>+++ b/arch/ppc/kernel/traps.c
>@@ -76,7 +76,6 @@ DEFINE_SPINLOCK(die_lock);
>
> int die(const char * str, struct pt_regs * fp, long err)
> {
>- static int die_counter;
> int nl = 0;
> console_verbose();
> spin_lock_irq(&die_lock);
>diff --git a/arch/s390/kernel/traps.c b/arch/s390/kernel/traps.c
>index 90ca82b..659f41c 100644
>--- a/arch/s390/kernel/traps.c
>+++ b/arch/s390/kernel/traps.c
>@@ -251,8 +251,6 @@ static DEFINE_SPINLOCK(die_lock);
>
> void die(const char * str, struct pt_regs * regs, long err)
> {
>- static int die_counter;
>-
> debug_stop_all();

> console_verbose();
> spin_lock_irq(&die_lock);
>diff --git a/arch/sh/kernel/traps.c b/arch/sh/kernel/traps.c
>index e9f168f..38ae9de 100644
>--- a/arch/sh/kernel/traps.c
>+++ b/arch/sh/kernel/traps.c
>@@ -78,8 +78,6 @@ DEFINE_SPINLOCK(die_lock);
>
> void die(const char * str, struct pt_regs * regs, long err)
> {
>- static int die_counter;
>-
> console_verbose();
> spin_lock_irq(&die_lock);
> bust_spinlocks(1);
>diff --git a/arch/sparc/kernel/traps.c b/arch/sparc/kernel/traps.c
>index dc9ffea..b9b4ddd 100644
>--- a/arch/sparc/kernel/traps.c
>+++ b/arch/sparc/kernel/traps.c
>@@ -89,7 +89,6 @@ void instruction_dump (unsigned long *pc
>
> void die_if_kernel(char *str, struct pt_regs *regs)
> {
>- static int die_counter;
> int count = 0;
>
> /* Amuse the user. */
>diff --git a/arch/sparc64/kernel/traps.c b/arch/sparc64/kernel/traps.c
>index dc652f2..556189b 100644
>--- a/arch/sparc64/kernel/traps.c
>+++ b/arch/sparc64/kernel/traps.c
>@@ -2208,7 +2208,6 @@ static inline struct reg_window *kernel_
>
> void die_if_kernel(char *str, struct pt_regs *regs)
> {
>- static int die_counter;
> extern void smp_report_regs(void);
> int count = 0;
>
>diff --git a/arch/x86_64/kernel/traps.c b/arch/x86_64/kernel/traps.c
>index e5403dc..93c4215 100644
>--- a/arch/x86_64/kernel/traps.c
>+++ b/arch/x86_64/kernel/traps.c
>@@ -570,7 +570,6 @@ void __kprobes oops_end(unsigned long fl
>
> void __kprobes __die(const char * str, struct pt_regs * regs, long err)
> {
>- static int die_counter;

```
>  printk(KERN_EMERG "%s: %04lx [%u] ", str, err & 0xffff,++die_counter);
> #ifdef CONFIG_PREEMPT
>  printk("PREEMPT ");
>diff --git a/arch/xtensa/kernel/traps.c b/arch/xtensa/kernel/traps.c
>index 693ab26..97cc4e2 100644
>--- a/arch/xtensa/kernel/traps.c
>+++ b/arch/xtensa/kernel/traps.c
>@@ -465,7 +465,6 @@ DEFINE_SPINLOCK(die_lock);
>
> void die(const char * str, struct pt_regs * regs, long err)
> {
>- static int die_counter;
>  int nl = 0;
>
>  console_verbose();
>diff --git a/include/linux/kernel.h b/include/linux/kernel.h
>index b8f8252..54c4fc4 100644
>--- a/include/linux/kernel.h
>+++ b/include/linux/kernel.h
>@@ -202,6 +202,7 @@ extern enum system_states {
> #define TAINT_USER   (1<<6)
>
> extern void dump_stack(void);
>+extern int die_counter;
>
> #ifdef DEBUG
> /* If you are writing a driver, please use dev_dbg instead */
>diff --git a/kernel/panic.c b/kernel/panic.c
>index 623d182..e75eea4 100644
>--- a/kernel/panic.c
>+++ b/kernel/panic.c
>@@ -150,23 +150,27 @@ EXPORT_SYMBOL(panic);
>  *  'R' - User forced a module unload.
>  *  'M' - Machine had a machine check experience.
>  *  'B' - System has hit bad_page.
>+ *  'D' - An OOPS or BUG has happened.
>  *  'U' - Userspace-defined naughtiness.
>  *
>  * The string is overwritten by the next call to print_taint().
>  */
>+
>+int die_counter;
>
> const char *print_tainted(void)
> {
>  static char buf[20];
>- if (tainted) {
>-  snprintf(buf, sizeof(buf), "Tainted: %c%c%c%c%c%c%c",
```

```
>+ if (tainted || die_counter) {
>+  snprintf(buf, sizeof(buf), "Tainted: %c%c%c%c%c%c%c%c",
>    tainted & TAINT_PROPRIETARY_MODULE ? 'P' : 'G',
>    tainted & TAINT_FORCED_MODULE ? 'F' : ' ',
>    tainted & TAINT_UNSAFE_SMP ? 'S' : ' ',
>    tainted & TAINT_FORCED_RMMOD ? 'R' : ' ',
>     tainted & TAINT_MACHINE_CHECK ? 'M' : ' ',
>    tainted & TAINT_BAD_PAGE ? 'B' : ' ',
>-   tainted & TAINT_USER ? 'U' : ' ');
>+   tainted & TAINT_USER ? 'U' : ' ',
>+   die_counter ? 'D' : ' ');
>  }
>  else
>   snprintf(buf, sizeof(buf), "Not tainted");
>-
>To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
>the body of a message to majordomo@vger.kernel.org
>More majordomo info at  http://vger.kernel.org/majordomo-info.html
>Please read the FAQ at  http://www.tux.org/lkml/
>
```