

Feature: The Kernel Exploit Time-line

xiando, 2004-06-18, v1.0.1

How the recent Linux kernel exploit issue proves Linux and Open Source is awesome

1. So I told the world, who could have known if they've just bothered to look
 2. Goods and bads
 3. The Kernel DOS Exploit Time-line
 4. What the press wrote
 - ◆ 4.1. English
 - ◆ 4.2. Norwegian
 5. Why was/is the original article changed over and over?
-

1. So I told the world, who could have known if they've just bothered to look

A 20 line long C program dubbed `evil.c`, capable of closing down Linux servers running unpatched versions of Kernel 2.4.xx and <2.6.7rc2, was made publicly available on this site 2004-06-11 [[/news/2004/06/11_kernel_crash](#)], after being submitted to the Linux Kernel Mailing list two days earlier ([2004-06-09 21:03](#)). My scoop turned out to be way much, much bigger than I ever imagined. It became very hot when slashdot ran it as a story and I eventually got "world famous" for the quote `Assume your kernel is vulnerable unless you have good reason to believe it is safe.`

Why did I release such dangerous source code without notifying the kernel developers, and before a patch was available? I am not sure I did..

When Linuxreviews posted the story *the code had already been publicly available* as gcc bug 15905 for two days (Opened: 2004-06-09), and the Kernel developers had been notified and were working a solution. The original on Linuxreviews article did point out that 2.4.26-rc3-Gentoo is immune to the exploit, and told the world where to get it and how to install it. Temporary, but bad, patches (`signal.c` etc) were added as they became available. Readers were at all times informed how they could protect themselves.

Some say publicly releasing exploit code before a patch is available is bad and against all common sense. **I am confident I would be sued and held liable by the vendor for damages if I had publicly released similar code that could exploit proprietary products, being OS or other software.** I probably took "open source mentality" a tad too far. Information wants, and should, be free, but some information is probably best kept among the white-hats.

Anyone could have picked this up two days before I posted a story about it.
All I did was add **fat types**.

2. Goods and bads

There are upsides and downsides to showing the world "evil" code. The GNU/Linux OS relies on millions of people who dedicate their valuable time to look at source code, to checking it for errors and to improving it. Yet none of them had spotted it was so easy to corrupt the Kernels FPU state. Why? Personally I have used GNU/Linux a number of years and I never bothered to read any of the kernel code. I have no clue how it works. I know it does work, and apart from the once in a century flaws that inevitably appear it has done so flawlessly. Even `evil.c` did not hurt me, the kernel I was running when it came to my attention was immune and I had to recompile another kernel version just to make sure it could do any damage.

All the eyes on the small `evil.c` made possible solutions appear extremely quickly, and the official fix was available before the story hit the major press.

3. The Kernel DOS Exploit Time-line

All dates are converted to UTC and ISO format.

- **2004-06-09 20:38** : Stian reports that evil.c can crash the kernel as gcc bug 15905. The code is now publicly available.
- **2004-06-09 21:03** : Stian reports evil.c to the Linux Kernel Mailing list (1) (2).
- **2004-06-11 20:01** : linuxreviews.org publishes evil.c, reports that 2.4.26-rc3-gentoo is immune and explains how to install this kernel version.
- **2004-06-12 13:14** : Stian Skjelstad posts a temporary solution (signal.c patch)
- **2004-06-12 18:45** : Sergey Vlasov suggests changing "asm volatile("fwait");" to "asm volatile("fnclex");"
- **2004-06-12 20:25** : Andi Kleen posts another possible solution which involves patching numerous files
- **2004/06/14 09:06** The story hits slashdot. The cat is really out of the bag. Black-hats start abusing the exploit. University and commercial servers start going down. White-hats start patching their kernels.
 - ◆ *At this point Andi Kleen's, Stian Skjelstad's and Sergey Vlasov patches were available on Linuxreviews in addition to a simple explanation on how to install 2.4.26-rc3-gentoo.*
- **2004/06/14 13:04** Official patches for 2.4.xx and 2.6.x are submitted to BitKeeper
- **2004/06/14 16:23** RedHat releases updated kernels (FEDORA-2004-171)
- **2004/06/14 20:37** Gentoo releases updated kernels (changelog)
- **2004/06/14 14:05** SuSE releases updated kernels (SuSE-SA:2004:017)
- **The following days** : The story hits the major press
- **2004/06/16 06:01** : Linux Kernel 2.6.7 is announced

5 days went by between the date *the first time the code was publicly released* and the date when the *official patch* became available. Less than **3** days passed before a *working temporary solution* was available. **2** days had gone by when I discovered kernel patch-sets what were immune.

The worlds *biggest* Linux vendor *RedHat* spent less than *4 hours* "twinning their thumbs" from the time the official patches became available to the time *patched kernels were ready and available on their ftp site*.

Consider that *CERT Coordination Center* has a **45** day waiting period before releasing bug advisories publicly. The Kernel patches became available in 1/9 of that!

The availability of patches does not mean all vulnerable systems are upgraded instantly. Upgrading the Linux kernel does require a certain level of skill. Luckily this exploit only affects hosting providers and other professionals who offer (paying) customers command line (shell) access and therefore should have this level of know-how. This is, and never was, in any way a threat to home users.

4. What the press wrote

4.1. English

- [New Linux Kernel Crash-Exploit discovered](#) (slashdot.org)
- [Flaw pops up in Linux kernel](#) (news.com)
- [New Kernel Crash-Exploit discovered using FPU Registers](#) (linuxelectrons.com)
- [Flaw In Linux Kernel](#) (hardwaregeeks.com)
- [Flaw pops up in Linux kernel](#) (ZDNet)
 - ◆ [Flaw pops up in Linux kernel](#) (ZDNet Australia)
 - ◆ [Rogue code can take down Linux systems](#) (ZDNet UK)
- [Flaw pops up in Linux kernel](#) (CNETAsia)
- [Flaw pops up in Linux kernel](#) (techrepublic.com)
- [Flaw pops up in Linux kernel](#) (globetechnology.com)
- [Linux Kernel Bug Found...and Quickly Fixed](#) (linuxworld.com)
- [Linux flaw exposes crash code threat](#) (silicon.com)
- [Kernel flaw makes Linux crash easy](#) (linuxworld.au)
- [Kernel flaw makes Linux crash easily](#) (computerworld.com)
- [Kernel flaw can crash Linux, warn security experts](#)
- [Kernel flaw makes Linux crash easy](#) (pcworld.nz)
- [New Linux Security Hole Found](#) (eweek.com)
- [Exploits found in Linux, CVS project](#) (geek.com)
- [Kernel Bug Makes Linux Easy To Crash](#) (cxotoday.com)
- [20 lines of C code can crash Linux](#) (tomshardware.com)
- [New Linux security hole found](#) (net-security.org)
- [Linux bug discovered](#) (theinquirer.net)

Germany

- [Norweger entdeckt Sicherheitsloch im Linux-Kernel 2.4x und 2.6x](#) (silicon.de)
 - ◆ x86-Systeme können zum Absturz gebracht werden

Sweedden (Söta bror)

- Norrman varnar för lucka i Linux (idg.se)

4.2. Norwegian

- Alvorlig kjernefeil rettet i Linux (digi.no)
- Feil får Linux til å krasje (pcworld.no)

5. Why was/is the original article changed over and over?

To keep it up to date. It is that simple.

I try to make as much of Linuxreviews content as possible licensed under the GNU GPL. What this means is that anyone is free to download the page source (available at the bottom of each article) and publish it pretty much without restrictions. You may not remove my name completely and claim it is solely your work, but you are totally free to make improvements and tell anyone you did. And I do make corrections to any article I have created if someone point out that I got the facts wrong or provides additional information. Embarrassing spelling errors are also silently corrected..

> [Linux Reviews](#) > [Features](#) >

Feature: The Kernel Exploit Time-line