

## Potential bug in fs/binfmt\_elf.c?

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-03/1231.html>

---

**From:** Mike Hearn ([mike\\_at\\_navi.cx](mailto:mike_at_navi.cx))

**Date:** 03/05/04

To: [linux-kernel@vger.kernel.org](mailto:linux-kernel@vger.kernel.org)

Date: Fri, 05 Mar 2004 17:38:01 +0000

Hi,

I believe there is a problem in fs/binfmt\_elf.c, around line 700 (kernel 2.6.1)

When mapping a nobits PT\_LOAD segment with a memsize > filesize, the kernel calls set\_brk (which in turns calls do\_brk) to map and clear the area, but this discards access permissions on the mapping leading to rwx protection. This causes a load failure on systems where the VM cannot reserve swap space for the segment, unless overcommit is active (on many systems it's not on by default).

I don't know this code well, but it seems that this discarding of access permissions on the unlikely codepath is incorrect. I filed bug #2255 [1] on it.

Could somebody who understands the ELF loading code please check to see if this is a bug, and if so produce a patch?

The ability to define a new (large) ELF section which isn't backed by swap space nor disk space and that will be mapped to a specific VMA range is needed by Wine to reserve the PE load area.

Currently the fact that the section is always mapped rwx despite being marked read-only in the binary prevents us from using this as a solution to the problems caused by exec-shield/prelink, meaning the only solution is to bootstrap the ELF interpreter ourselves from a statically linked binary. Clearly we'd rather not do that.

Thanks to pageexec@freemail.hu for bringing the matter to my attention.

Your assistance is appreciated,  
thanks -mike

[1] [http://bugzilla.kernel.org/show\\_bug.cgi?id=2255](http://bugzilla.kernel.org/show_bug.cgi?id=2255)

-

## Linux-Kernel: Potential bug in fs/binfmt\_elf.c?

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@vger.kernel.org](mailto: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/>