

Re: [patch] remove MNT_NOEXEC check for PROT_EXEC mmmaps

Re: [patch] remove MNT_NOEXEC check for PROT_EXEC mmmaps

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-10/msg00906.html>

- *From:* Stas Sergeev <stsp@xxxxxxxx>
 - *Date:* Tue, 03 Oct 2006 21:15:05 +0400
-

Hello.

Arjan van de Ven wrote:

no what bothers me that on the one hand you want no execute from the partition, and AT THE SAME TIME want stuff to execute from there (being libraries or binaries, same thing to me).

The original problem came from "noexec" on /dev/shm mount. There is no library and no binary there, but the programs do shm_open(), ftruncate() and mmap(MAP_SHARED, PROT_EXEC) to get some shared memory with an exec perm. That fails.

That duality feels strange to me,

IMHO there should be some policy that can be achieved. If the policy is: "noexec should fail execve()", then this can be achieved, and that's what it was in the past. What is the policy now? The things like a possibility to mprotect() that memory to PROT_EXEC, or in case of a MAP_PRIVATE, to simply use MAP_ANONYMOUS then read(), suggests that there is no strict policy at all any more.

I could understand if you wanted noexec to be MORE strict; I fail to understand why you want it LESS strict!

My point is that it is neither more not less strict with such a change. If the workaround is trivial anyway (either mprotect or use MAP_ANONYMOUS and read()), then there is no point in such a strictness. On the other hand, the programs break.

What was pointed out by Hugh is that the current behaviour is needed to solve one particular problem, which is when the user invokes ld.so directly and you want it to fail on a noexec partition. I accept that argument, but I have to

Re: [patch] remove MNT_NOEXEC check for PROT_EXEC mmmaps

Re: [patch] remove MNT_NOEXEC check for PROT_EXEC mmap

add that the mmap change doesn't solve the similar problem when the user uses ld.so directly to execute the binaries he doesn't have an exec permissions for.

So I think another solution is needed: the one, preferably, not breaking an existing apps; solving both of the above problems, not just one of them; allowing an admin to control that behaviour in a convenient way.

My idea is to execute the loader with the fsuid=0. Then you can do simply "chmod 'go-x' ld.so", and the problem solved. I'd like any opinions on that idea, although nothing positive is expected at that point. :)

What breaks?

You missed the beginning of the discussion, but briefly: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=386945> ... breaks UML and dosemu.

Also I speculate that it makes Wine slower causing it to fallback to read() if the windows partition is mounted with "noexec" (which I think is/was common). In that case people will never figure out why Wine suddenly became slower and more memory-consuming than before.

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>