

Re: How to develop a random number generation device

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg02083.html>

- *From:* MooseFET <kensmith@xxxxxxxxxx>
 - *Date:* Tue, 11 Sep 2007 19:08:16 -0700
-

On Sep 11, 4:58 pm, John Larkin
<jjar...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:
[... buffer overflow ...]

It sounds to me like C compilers/linkers tend to allocate memory to code, buffers, and stack sort of anywhere they like.

No the problem isn't really with code mixed with data. It is data mixed with data. The return addresses etc are on the stack along with the local arrays. This means that a routine can overwrite the return address with data by walking off the end of an array. Once that happens the return instruction jumps you to the bad code.

Why can't at least the compilers be fixed so that they put all the stacks first, then the code, then all the buffers?

With an x86's MMU, you can make segments for code and stack and the like that have limits on their sizes. The problems can be partly overcome by this.

On most programs, the stack is just above the data segment in physical memory and the malloc() obtained memory is beyond that.