

Re: How to develop a random number generation device

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg02167.html>

- *From:* Nobody <nobody@xxxxxxxxxxxx>
 - *Date:* Wed, 12 Sep 2007 15:41:18 +0100
-

On Tue, 11 Sep 2007 19:16:34 -0700, MooseFET wrote:

Nothing the OS does can prevent machine code from overrunning a buffer.

Absolute nonsense. Perhaps buffer overruns can't be prevented using C++, but they *can* be prevented.

Not by the *OS*.

[Using the traditional definition (the kernel, i.e. the part that the process can't bypass), not Microsoft's definition (the OS plus a bunch of user-level applications which are bundled in order to kill the market for competing applications).]

A C++ compiler could be created that inserted checking code in every operation that may overrun. Every buffer would have to have its length recorded somewhere.

The language can prevent overruns; the OS just sees machine code.

The OS can let your program single step and check what every instruction does.

The OS doesn't know where one "buffer" ends and the next one begins; it's all just memory. And single-stepping programs would be out of the question from an efficiency perspective.

The OS can always leave a dead page after every malloc() block so you get a segment fault on stepping off the end.

Re: How to develop a random number generation device

That prevents overflowing malloc()ed b