

# Re: How to develop a random number generation device

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg02024.html>

---

- *From:* Nobody <nobody@xxxxxxxxxxxx>
  - *Date:* Wed, 12 Sep 2007 00:04:30 +0100
- 

On Tue, 11 Sep 2007 13:11:18 -0700, John Larkin wrote:

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

Ancient computers, PDP-11 and VAX certainly, had memory management hardware that separated I and D space, where I space was read-only, and D space could not be executed. And the OS's enforced those rules. It was common to have many users running the exact same code, but mapped into different data spaces.

Problem is, neither Intel nor Microsoft was in the mainstream of computing when they kluged up x86 and Windows.

W^X (write or execute but not both) is available on current systems, but that doesn't necessarily cure all buffer overflow exploits. It prevents an attacker from injecting new code, but it doesn't stop them from calling existing code with their own data.

The latter may be just as good as the former if some form of "Swiss army knife" function (e.g. execute arbitrary VB/C#/JS/etc code) is present in the code space. Actually, system() is almost certain to be there, and there isn't much you can't do by passing it the right string.