

# Re: How to develop a random number generation device

---

*Source:* <http://sci.tech-archive.net/Archive/sci.electronics.design/2007-09/msg02171.html>

---

- *From:* Nobody <nobody@xxxxxxxxxxxx>
  - *Date:* Wed, 12 Sep 2007 15:54:15 +0100
- 

On Wed, 12 Sep 2007 13:42:56 +0000, No Spam wrote:

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

That's not true. Many operating systems are by design, immune to buffer over-runs modifying unrelated code.

The issue isn't about modifying code, related or otherwise. It's about either injecting new code or executing existing code with attacker-supplied data.

This isn't about protecting one process from another, but about protecting a process from itself. Most of the existing mechanisms for mitigating buffer overruns are implemented in either the compiler or libraries. The only OS-level mechanisms (things that work on any executable, however it was built) involve making it harder to exploit an overrun (e.g. randomising memory locations) rather than actually preventing the overrun.

Given that:

- a) this would make Windows totally incompatible with most existing software, and

Did you mean to write "nothing the \*Windows\* OS does can prevent machine code from overrunning a buffer?"

No, the issues apply to any OS. But binary compatibility is much more important for Windows (and Mac) than for Linux.

If you try to run a 5-year old Linux binary on a current distribution, you'll probably find that a lot of the interfaces on which it depends have either disappeared or have changed in an incompatible manner. Lack of a

Re: How to develop a random number generation device

stable ABI is a simple fact of life on Linux.