Re: Thou shalt have no other gods before the ANSI C standard

Source: http://www.derkeiler.com/Newsgroups/sci.crypt/2005−02/4928.html

---

From: Brian Inglis (*Brian.Inglis_at_SystematicSW.Invalid*)
Date: 02/27/05

```
Date: Sun, 27 Feb 2005 21:10:52 GMT
```

On Sun, 27 Feb 2005 14:33:06 −0500 in alt.folklore.computers, "Trevor L. Jackson, III" <tlj3@comcast.net> wrote:

>David Wagner wrote:
>> Trevor L. Jackson, III wrote:
>>
>>>We need to distinguish reads from writes. I'm assuming you are
>>>interested in errant writes rather than errant reads. Are you aware of
>>>any exploits based upon errant buffer reads?
>>
>>
>> Nothing really significant. I know of a few examples of exploits based
>> upon errant reads, but they were were "information disclosure" or "crash
>> the machine" vulnerabilities, rather than something that immediately
>> leads to the ability to take control of the machine. However, if your
>> application isn't handling any important secrets, errant reads seem like
>> a pretty low−priority item, I would think −− it seems to be pretty rare
>> to encounter a buffer overrun that leads to some kind of serious information
>> disclosure vulnerability but does not allow attackers to take control of
>> the application.
>
>Even if errant reads are a real problem, focusing on errant writes is
>probably not a mistake.
>>
>>>There's a similar idiom, NCHECK, that can be used to control
>>>the compilation of parallel implementations, one for production (speed,
>>>size, etc) and one for correctness (big & slow). The NCHECK symbol is
>>>most useful when it controls not only comparison of the results, but
>>>also comparison of the process.
>>
>> I take it this is another way of describing 2−version programming,
>> excep