

Re: [Lit.] Buffer overruns

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2005-02/0285.html>

From: BRG (brg_at_nowhere.org)

Date: 02/01/05

Date: Tue, 01 Feb 2005 14:59:17 +0000

Trevor L. Jackson, III wrote:

> *Fair warning: I disagree with a lot of this, but none of my comments
> should be interpreted as criticism, more like a different interpretation
> of the same facts. --tj3 (and most of this is OT or only tangential for
> sci.crypt)*

Further warning : this is a long rant about the links between language and library design, systems engineering, the cultural influences associated with programming languages some of the factors that(in my view) contribute to the emergence of these cultures. Leave now if this doesn't interest you!

[snip]

>>>

>>> *I dispute that. There are C users who are not part of
>>> such a "culture". Therefore other factors are (also?)
>>> at work. It would behoove us to identify them.*

>>

>> *Yes there are many good people working with C. There are many doing
>> good work using C. There are superb designs that I have seen in C.*

>>

>> *But what I am considering is the average standard of systems
>> engineering that we see when we look across the totality of fielded
>> systems written in C.*

>

> *In C or in programming an general?*

Both to some extent but definitely worse within the C community than in, for example, the Ada community.

In my experience the vast majority

> *of software should be rejected unusable and unfixable. I've seen teams
> produce utter crap in lots of languages. And the kinds of problems in
> the code I've reviewed mostly have nothing to do with the language. Sure
> C is easy to misuse, but most procedural languages have weak points.
> Those aren't the real problem with most of the software I've had to review.*

sci.crypt: Re: [Lit.] Buffer overruns

It isn't the language that's the problem in any direct sense, its the cultural baggage that comes with the language that creates the problem.

- > *My experience includes consulting for medium and large size*
- > *organizations to evaluate their software development efforts. Since I*
- > *get hired when there are problems I may be committing a sampling error*
- > *when I use my experience as a basis. But the literature on software*
- > *development methodology is not dissimilar to my experience.*

I am a 'troubleshooter' too so I have the same problem :-)

- > *Point is that forcing all of those organizations to another language,*
- > *even ADA, would not improve the quality of their output.*

I don't think that forcing people to do things they don't want to do solves any problems so we can agree quickly on that.

- >> *There are a huge number of companies putting together and fielding*
- >> *abysmally designed systems in C simply because it is so easy and cheap*
- >> *to do.*
- >
- > *And they would do that in any language. Evan Ada.*

This is where I disagree since there really isn't any easy or cheap way of throwing a system together using an Ada tool base. So in my experience it doesn't happen this way.

This is because the cultural predisposition within the Ada community is to take a systems engineering approach so it is much easier to go down this path and much harder to go in any other direction.

- > > *In contrast I have _never_ seen a bucket shop approach used with*
- >
- >> *Ada because the culture that comes with Ada _is_ a systems engineering*
- >> *culture (I am contrasting Ada and C only because I have a lot of first*
- >> *hand experience of managing projects using these two languages).*
- >
- > *I think you have committed a sampling error here in that the*
- > *self-selected shops that pick Ada as one of the main implementation*
- > *languages clearly have an interest in a high quality software rather*
- > *than lower cost-to-produce software.*

I may well have done. But at a microscopic level it really doesn't matter whether the language caused the culture or the culture caused the language – in my experience the two are correlated irrespective of what caused this or whether this was 'chicken before egg' or 'egg before chicken'.

- > *If your force everyone to use Ada you would still see crappy code coming*
- > *from people who don't know or care how to do better.*

Yes but that is not my point since we all know that it is possible to write crappy code in any language.

But my point is not what *_can_* be done but what *_is_* being done. That is, what proportion of all systems written in Ada are crappy compared with what proportion of all systems written in C?.

And if you truly don't believe that this proportion is very different for the two communities, I think you are in denial. Again it not the microscopic analysis that concerns me but what is happening at a macroscopic level.

If we take the totality of the systems built using C there is plenty of evidence to show that a significant proportion of them are bug ridden, unreliable, unrobust and insecure. You can wash your hands and say that this has nothing whatsoever to do with the evolution of C and the culture of low level design that it has unwittingly fostered. But I don't accept that this is a correlation is purely random and devoid of cause.

Of course there are masses of influences at work – cultures develop in a largely (but not totally) uncontrolled way through a series of small steps and through the slow evolution of attitudes and through their progressive diffusion through the community. Its slow and subtle, so much so, that we often don't consciously realise that it is happening until it is well entrenched.

I don't think there can be much doubt that there is a very significant amount of poor systems development going on in C and this alone ought to spawn the question "why are so many systems that are developed using C poorly designed in systems engineering terms?". Even if you don't think that C carries any balme for this, it is still a question that deserves attention. But I don't see much evidence that this question is even being asked, let alone being answered.

> *So neither of the above observations about the systems engineering culture are suitable for generalization.*

>>

>> *And of course you are right that there are many cultural influences at work. But for whatever reason I believe that their is a link between the language and this culture, and that this link has led to the fielding of many systems that have been 'put together' rather than designed simply because this is so easy to do with C. And my view is that this phenomenon partly explains why so many of the software based systems we see today lack robustness, reliability and security.*

>

> *There's a cart/horse issue here. I read the above as suggesting that because C is so "easy to use" (a description I've never see used before) it tempts development organizations that would otherwise be diligent and careful to be indolent and sloppy.*

>

- > *I find that far fetched. I suspect the indolent and sloppy*
- > *organizations are the problem and their use of C is irrelevant.*

Well we will have to disagree on this. I do think it is very easy to throw a system together with C. You buy a couple of PCs, hire a few programmers out of college, tell them roughly what you want, feed them for a few months with beer and sandwiches and a few months later you will have a product to sell. It won't be a robust or reliable product (except by good fortune) but it will be on the market and be one more contribution to the systems that we see around us that seem to spend more time being rebooted than they ever do working (well not quite, but you know what I mean).

I really do believe it is both easy and cheap to throw a system together in C whereas doing it right is, at least initially, hard and costly. In Ada there is no 'cheap and easy' way to do the same thing so it just doesn't happen.

- > *For*
- > *example, if Modula had become the prevalent language rather than C and*
- > *you made the same observations about how really awful the average*
- > *practitioner's output was, then you would be blaming Modula, which has a*
- > *quite distinct set of strengths and weaknesses from C, for all of the*
- > *crappy code that is being written.*

How do I know? My observations are about what has happened and what is happening, not what might of happened if ...

- > *Please state why you believe the tools are influencing the people rather*
- > *than the people just doing their lazy, half-assed enough-to-get-by jobs.*

My view is that if we focus much of our training and education and early experience on a language that is intended for low level design, then we should not be in the least surprised to find that people design systems at this level and lack the higher level systems engineering driven design that is so important in achieving system characteristics such as security, safety, robustness and reliability.

In short this is the Dijkstra view of the influence of language design on the way people think and the ensuing impact that this has on the character of the systems that they produce.

- >> *I don't think it would be credible for anyone within the C community*
- >> *with an understanding of systems engineering to claim not to have*
- >> *noticed that this was happening.*
- >
- > *While I agree this is happening I do not agree that this is new. It has*
- > *been happening for over 30 years that I can attest to personally. I'm*
- > *sure people with greater experience have seen the same pattern repeated*
- > *even longer than that.*
- >

> *So that recent events lead you to believe there is something new under
> the sun?*

NO, NO, NO and NO! The very last thing I have claimed is that this is new! The culture that is correlated (irrespective of cause) with C at a macroscopic level and which has created a situation in which a significant proportion of the systems written in C are of low quality in a systems engineering sense is NOT a new phenomenon. The negative cultural influences that determine how some within the C community go about building systems have been slow and stealthy, so much so that we have not really noticed (or reacted to them) them as they have been taking place.

>> *And yet I don't see much evidence of concern about this. The community
>> may not have consciously fostered this culture but I don't see that it
>> has done much to counter its development either – by and large the
>> community 'has let it happen' as if it is a phenomenon that has
>> nothing to do with them.*

> *What duty do you believe the community has to counter the tendency you
> describe? If, in fact, the problem is not the language but the people
> and the organizations, then what duty does the community (or its
> leaders) have to change people?*

I don't frankly care whether you think the 'problem' is caused by the language or the language is the caused by the problem (i.e the people who had the problem designed the language). If you wish to deny any correlation between the use of C on the one hand and poor systems engineering on the other (irrespective of why this correlation is there) then I don't think there is any more that we can sensibly discuss.

I am used to 'denial' as a response to this charge and I just have to accept that crappy systems will continue to be foisted on us by the C community because this community doesn't care that this is happening, washes it hands of any responsibility for it and sees it as no part of its duty to society to work out why this is happening or whether there is anything that can be done to counter it.

>> *All of this is what I mean when I say that a programming language has
>> a powerful influence on the way people approach the process of
>> producing solutions – if we put so much emphasis during people's
>> education and training on low level design, we should not then be
>> surprised when this is the level at which they design things.*

> *That's an educational issue rather than an language issue. At what
> point can a student or trainee even conceive of the issues in a
> high-level design? I suspect the problem is lack of sufficient
> education and experience rather than the wrong kind.*

Of course it is. But that doesn't make it a non issue. Nor does it make it one that is of no concern to C community.

sci.crypt: Re: [Lit.] Buffer overruns

>>>> *I totally agree with you that this is primarily an educational
>>>> issue. But its not new and yet we don't seem to have been able to do
>>>> anything about it.*

>>>

> *It's a motivation issue. Most educational issues are. if the student
> is not interested in learning there is nothing that _anyone_ can do to
> make them. But a student interested in learning will do whatever it
> takes to gain what they want.*

[snip]

I have cut the rest Trevor because you have moved into the 'its everyone else's fault but ours, the C community has played no part in this' mode.

Yes it did. At the very least it let it happen.

Brian Gladman