Smashing Stacks like it's 1996

An introduction to exploit development for new C programmers

Jo Hallett — 2023-10-23

OpenBSD 7.4 | .:: Phrack Magazine ::. | c++ - Why do people

https://softwareengineering.stackexchange.com/questions/321547

StackExchange | Search on Software Engineering... | Log in | Sign up

SOFTWARE ENGINEERING

Home

PUBLIC

Questions

Tags

Users

Companies

Unanswered

Why do people use C if it is so dangerous?

Ask Question

Asked 7 years, 4 months ago Modified 1 year ago Viewed 48k times

136

I am considering learning C.

But why do people use C (or C++) if it can be used 'dangerously'?

By dangerous, I mean with pointers and other similar stuff.

Like the Stack Overflow question *Why is the gets function so dangerous that it should not be used?*. Why do programmers not just use Java or Python or another compiled language like Visual Basic?

c++ c

Share
Improve this question
Follow

edited May 23, 2017 at 12:40
Community Bot
1

asked Jun 7, 2016 at 18:50
Tristan
1,295 ● 2 ● 9 ● 11

172 Why do chefs use knives, if they can be used 'dangerously'? – oerkelens Jun 9, 2016 at 8:24

82 With Great Power Comes Great Responsibility. – Pieter B Jun 9, 2016 at 11:15

10 Joe Blow, pontificate much? – Matthew James Briggs Jun 9, 2016 at 19:45

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

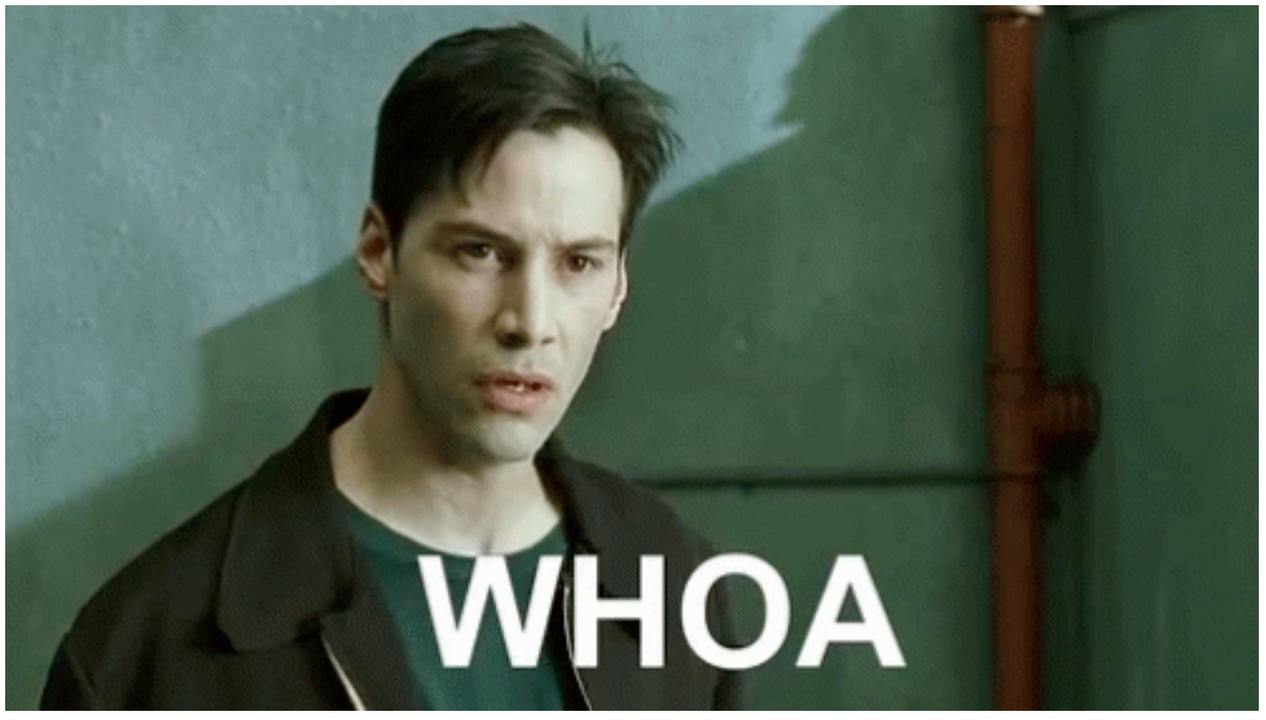
`smash the stack` [C programming] n. On many C implementations it is possible to corrupt the execution stack by writing past the end of an array declared auto in a routine. Code that does this is said to smash the stack, and can cause return from the routine to jump to a random address. This can produce some of the most insidious data-dependent bugs known to mankind. Variants include trash the stack, scribble the stack, mangle the stack; the term mung the stack is not used, as this is never done intentionally. See spam; see also alias bug, fandango on core, memory leak, precedence lossage, overrun screw.

Introduction

Over the last few months there has been a large increase of buffer overflow vulnerabilities being both discovered and exploited. Examples of these are syslog, splitvt, sendmail 8.7.5, Linux/FreeBSD mount, Xt library, at, etc. This paper attempts to explain what buffer overflows are, and how their exploits work.

Basic knowledge of assembly is required. An understanding of virtual memory concepts, and experience with gdb are very helpful but not necessary. We also assume we are working with an Intel x86 CPU, and that the operating system is Linux.

Some basic definitions before we begin: A buffer is simply a contiguous block of computer memory that holds multiple instances of the same data type. C programmers normally associate with the word buffer arrays. Most commonly, character arrays. Arrays, like all variables in C, can be declared either static or dynamic. Static variables are allocated at load time on the data segment. Dynamic variables are allocated at run time on the stack. To overflow is to flow, or fill over the t
We will concern ourselves only


WHOA

You get taught C because it teaches you to think like the machine...

Most programmers will not program in it day to day...

Modern OSs/languages prevent an awful lot of what I'm gonna show you...

(but not all of it ;-))

So lets go back to '96!

Lets do some hacking together and work through an exploit together!

You are not expected to understand this!

https://git.sr.ht/~sherbert/talks/tree/main/item/Tutorials/Smashing-the-Stack-Like-Its-1996


Whoa...