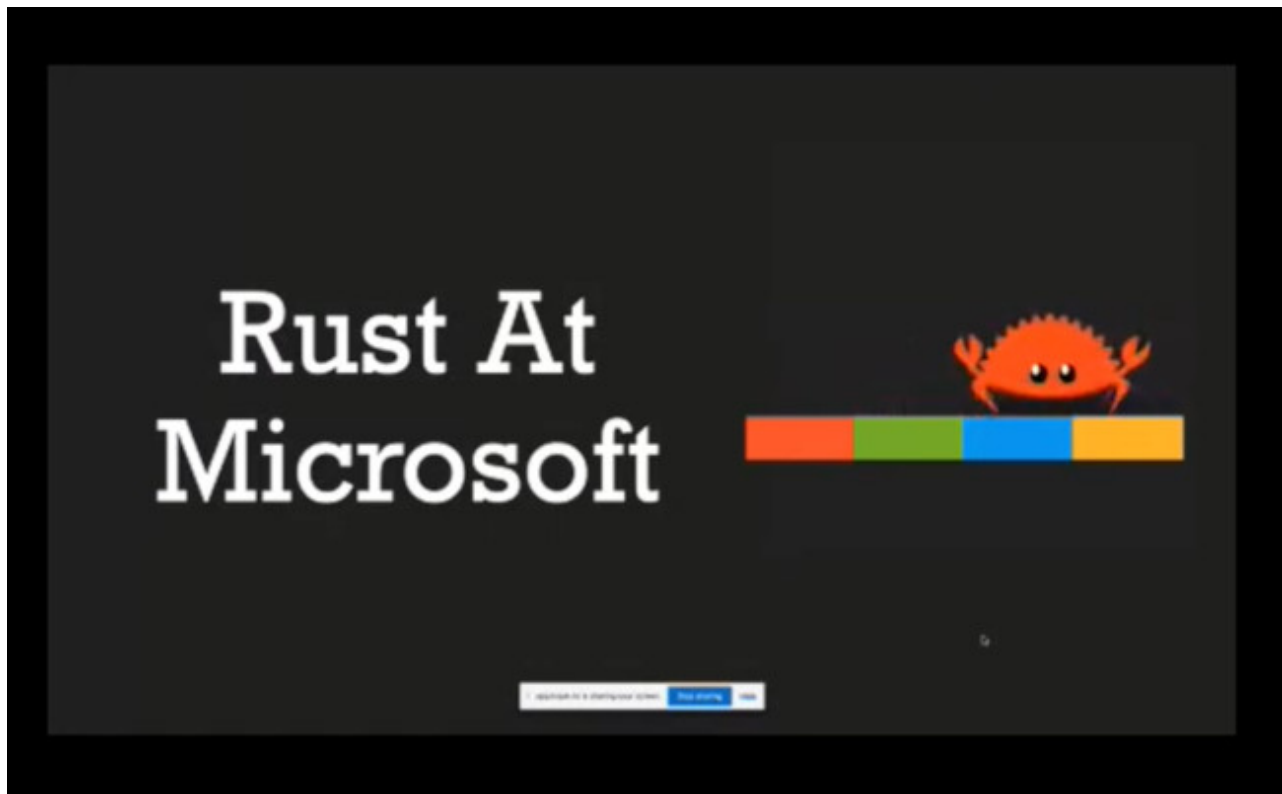


DEVELOPMENT / SECURITY

Microsoft: Rust Is the Industry's 'Best Chance' at Safe Systems Programming

10 Jun 2020 10:10am, by [Joab Jackson](#)

No matter how much investment software companies may put into tooling and training their developers, "C++, at its core, is not a safe language," said [Ryan Levick](#), [Microsoft](#) cloud developer advocate, during the AllThingsOpen virtual conference last month, explaining, [in a virtual talk](#), why [Microsoft is gradually switching to Rust](#) to build its infrastructure software, away from C/C++. And it is encouraging other software industry giants to consider the same.

"We're using languages that are, because they are quite old and come from a different era, do not provide us the ability to protect ourselves from these kinds of vulnerabilities, he [said](#). "C++ is not a memory safe language and no one would really pretend that it is," he said.

[In fact, Microsoft has deemed C++ no longer acceptable for writing mission-critical software.](#) The industry sorely needs to move to a [performant, memory-safe language](#)

C/C++ Can't Be Fixed

Today, C and C++ are the go-to languages for writing core system software. It is fast, with the only assembly between the code and the machine itself.

But the industry is being crippled by all the memory-related bugs — many of which are security hazards — caused by these languages. Now, 70% of the CVEs originating at Microsoft are memory safety issues, Levick said. “There is no real trend it’s just staying the exact same,” he said. “Despite massive efforts on our part to fix this issue it still seems to be a common thing.”

From a financial perspective, it makes sense, given the soaring cost of remedying this never-ending stream of memory-related errors. Back in 2004, each memory-related error cost industry about \$250,000 each, and that Microsoft estimation is probably on the lower-end, Levick said.

Of course, there are a number of efforts to boost C++ security, but while each is effective in the way it does, none entirely solves the problem.

One approach that has been long floated is to do more programmer training in how to write safer code. But, “there is zero evidence that doing holistic training of C and C++ developers will actually fix this issue in any significant way,” Levick said, citing Microsoft’s own heaps of dev internal training.

Static analysis is cited as another possible solution. But static analysis comes with too much overhead: It needs to be wired into the build system. “So there’s a lot of incentive not to use static analysis,” Levick said. “If it’s not on by default it won’t it won’t help.”

The same goes for runtime checks: “It’s impossible or it’s very least extremely hard to know when runtime checking contracts are used and when they’re not,” he said, adding that they also come with an operational overhead.

The Industry’s Best Chance

In response to this problem of memory-related errors, the [Microsoft Security Response Center](#) launched the [Safe Systems Programming Language](#) initiative. There, some work was dedicated for shoring up C/C++. [Verona](#), a new programming language being created for safe low-level programming, was also created here. But the third prong of

And we believe that to be Rust," he said.

Performance-wise, Rust is on par with C/C++, and maybe even slightly faster. Rust brings developer productivity, with package management, modern testing frameworks and the like. And programmers love Rust for it.

But the main reason Microsoft is so enamored with Rust is that it is a memory-safe language, one with minimal runtime checking. Rust excels in creating correct programs. **Correctness** means, roughly, that a program is checked by the compiler for unsafe operations, resulting in fewer runtime errors. Unsafe keyword is an option, but not the default. Unsafe Rust code is almost always a subset of a larger body of safe code. Unsafe mode is necessary for memory-assigning jobs like writing device drivers. But even here the unsafe portions of memory are encapsulated behind an API.

This ability to program safely is not one that should be taken lightly, Levick said. In fact, it can provide more than a 10x improvement, making it worthwhile for investment. This is largely because pretty much all C/C++ code needs to security audits for unsafe behavior, whereas unsafe code written in Rust that would need to be checked is only a small subset of most code bases.

While Microsoft is bullish on Rust, Levick admits that Microsoft core developers won't stop using C/C++ anytime soon.

"We have a lot of C++ at Microsoft and that code is not going anywhere," he said. "In fact, Microsoft c++ continues to be written and will continue to be written for a while."

A lot of tooling is built around C/C++. In particular, Microsoft binaries are now almost completely built on the Microsoft Visual C++ compiler which produces MSVC binaries, whereas Rust relies on LLVM.

Perhaps the biggest challenge, though, is cultural. "There is just some people that just want to get their job done in the language that they already know," Levick admitted.

Still, the industry seems to be moving towards Rust. Amazon Web Services uses it, in part for deploying the Lambda serverless runtime, as well as for some parts of EC2. Facebook has started using Rust, as has Apple, Google, Dropbox and Cloudflare.

Architecture

Development

Operations



The dates All Things Open 2020 have been announced: [Oct. 20–22](#).

The New Stack does not allow comments directly on this website. We invite all readers who wish to discuss a story or leave a comment to visit us on Twitter or Facebook. We also welcome your news tips and feedback via email: feedback@thenewstack.io.

Amazon Web Services is a sponsor of The New Stack.

NEWS

THE NEW STACK UPDATE

A newsletter digest of the week's most important stories & analyses.

Email Address

Subscribe

We don't sell or share your email. By continuing, you agree to our [Terms of Use](#) and [Privacy Policy](#).



DATA / DEVELOPMENT / TOOLS

MongoDB 4.4 Promises Less Work for Database Developers

9 Jun 2020 11:53am, by [B. Cameron Gain](#)

MACHINE LEARNING

The Moral Choice Machine: An AI that Learns 'Right' from 'Wrong'

8 Jun 2020 11:35am, by [Kimberley Mok](#)

SPONSORED FEED



Statement from CNCF General Manager
Priyanka Sharma on the Black Lives Matter
Movement

JUNE 11, 2020



MongoDB Partner of the Year Award Winners
JUNE 11, 2020

Architecture

Development

Operations

JUNE 11, 2020

'SALTSTACK

The Death Star... Who's To Blame

JUNE 11, 2020



Vote for the Best App in the Redis 'Beyond Cache' Hackathon

JUNE 11, 2020



How Palo Alto Networks and Okta Are Helping Enable Secure Remote Work

JUNE 11, 2020

CLOUDFOUNDRY

Summit Talk: Cloud Foundry for Everyone!

JUNE 11, 2020



Getting Started With NerdGraph—The New Relic GraphQL API Explorer

JUNE 11, 2020



Fighting Ransomware: Part Four – UBA and ONTAP with FPolicy External Mode

JUNE 11, 2020



TL;DR Tech Tips – Flux Timestamps

JUNE 11, 2020



NIST: Adopt a Secure Software Development Framework (SSDF) to Mitigate Risk of

Architecture

Development

Operations

Deliver Better Customer Experiences With
Real-Time Business Orchestration by Ariel
Russo

JUNE 11, 2020



All Tier 1 Plugins Support Configuration as
Code

JUNE 10, 2020



Get a jump on errors with Sentry and
Atlassian

JUNE 10, 2020



Postmortem of Root Certificate Expiration
from 30 May 2020

JUNE 10, 2020



Software Package Management with AWS
CodeArtifact

JUNE 10, 2020



The Option Value of Hybrid Cloud

JUNE 10, 2020



IT Modernization Powers Resilient
Organizations

JUNE 10, 2020



Optimize your marketing campaign
investment by leveraging BizDevOps

JUNE 10, 2020

Architecture

Development

Operations

Research Paper on CockroachDB

JUNE 10, 2020



Medium Article – Deploying AWS Lambda Functions On GKE using Knative and Triggernesh KLR

JUNE 10, 2020



Run Apache Cassandra on Kubernetes 15x Faster with Arrikto and DataStax

JUNE 10, 2020



Announcing the Terraform Visual Studio Code Extension v2.0.0

JUNE 10, 2020



GitLab Critical Security Release: 13.0.6, 12.10.11, 12.9.10

JUNE 09, 2020



Continuous deployment for progressive web applications

JUNE 08, 2020



Linux Foundation Support for the Black Community

JUNE 08, 2020



Our privilege is our responsibility

JUNE 05, 2020

Architecture

Development

Operations

2

JUNE 05, 2020



BHIM Breach: Policy Guardrails No Longer Optional

JUNE 02, 2020



Our Plans with ThousandEyes: Expanding Monitoring Across the Public Internet for All Applications at the Heart of Today's Digital Businesses

JUNE 01, 2020



Lightbend Telemetry 2.14 Released

MAY 28, 2020



Autoscaling Citrix ADC CPX using Kubernetes HPA custom metrics

MAY 28, 2020



HAProxy Go Packages Ecosystem

MAY 27, 2020



Announcing Aspen Mesh 1.5

MAY 20, 2020



Introducing the Edge Metal Private Beta

MAY 04, 2020



Architecture

Development

Operations

ARCHITECTURE[Cloud Native](#)[Containers](#)[Edge/IoT](#)[Microservices](#)[Networking](#)[Serverless](#)[Storage](#)**DEVELOPMENT**[Cloud Services](#)[Data](#)[Development](#)[Machine Learning](#)[Security](#)**OPERATIONS**[CI/CD](#)[Culture](#)[DevOps](#)[Kubernetes](#)[Monitoring](#)[Service Mesh](#)[Tools](#)**THE NEW STACK**[Ebooks](#)[Podcasts](#)[Events](#)[Newsletter](#)

