

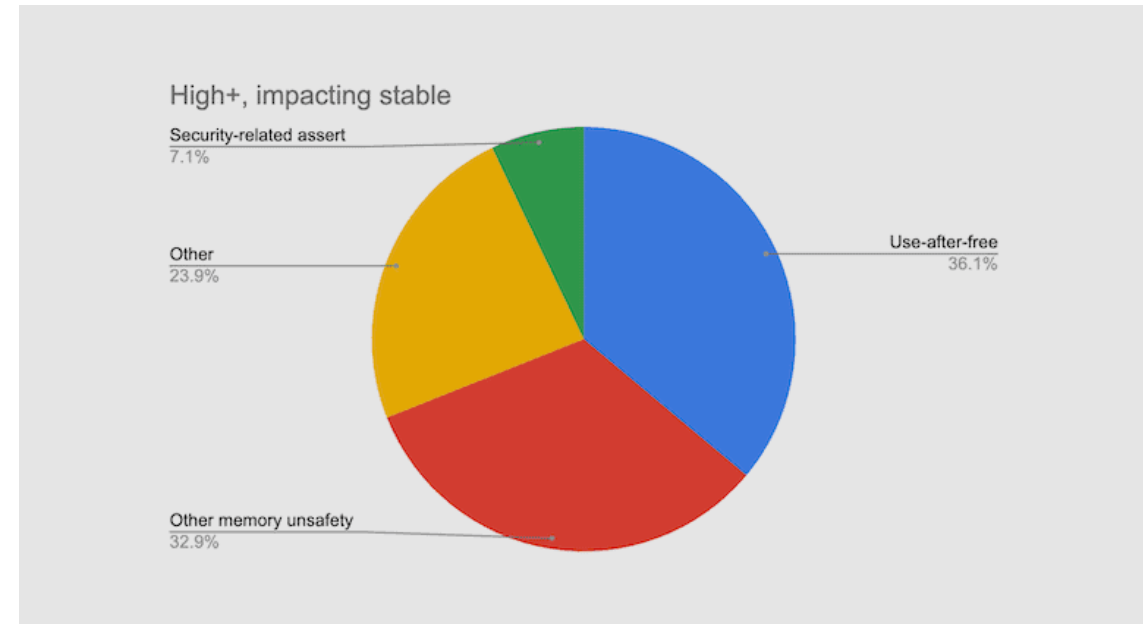
BENEFITS AND DRAWBACKS OF ADOPTING A SECURE PROGRAMMING LANGUAGE: RUST AS A CASE STUDY

Kelsey Fulton, Anna Chan, Daniel Votipka,
Michelle Mazurek, Michael Hicks



Memory safety vulnerabilities remain a problem

- 70% of vulnerabilities in Chrome are memory safety problems (May 2020)
- 70% of vulnerabilities in Microsoft products are memory safety problems (2002 - 2019)
- **C/C++ source of most of these bugs**



Google / <https://www.zdnet.com/article/chrome-70-of-all-security-bugs-are-memory-safety-issues/>

Fix or replace (or both)

Case Study: Rust



- Attempt safety and performance (e.g., no GC)
 - Useful where C/C++ are hardest to replace
- What does the adoption of secure programming languages look like?
- What benefits (if any) accrue after the adoption of a secure programming language?

Case Study: Rust

- Semi-structured interview with senior developers (I = 16)
- Survey with Rust community (S = 178)

Learning Rust

- Rust is hard to learn.

Rust has “a near-vertical learning curve.”

Learning Rust

- Rust is hard to learn.
 - Rust is more difficult to learn than other languages

Learning Rust

- Rust is hard to learn.
 - Rust is more difficult to learn than other languages
- Easy to find solutions to problems

Learning Rust

- Rust is hard to learn.
 - Rust is more difficult to learn than other languages
- Easy to find solutions to problems
 - Good compiler error messages

“Most of the time the compiler is very,
very good at telling you exactly what the
problem is”

Learning Rust

- Rust is hard to learn.
 - Rust is more difficult to learn than other languages
- Easy to find solutions to problems
 - Good compiler error messages
 - Good official documentation

Learning Rust

- Rust is hard to learn.
 - Rust is more difficult to learn than other languages
- Easy to find solutions to problems
 - Good compiler error messages
 - Good official documentation
 - Helpfulness of community

Positive Impact on Development

- Improves confidence in code correctness
- Improves long-term productivity
- Improves safe development in other languages
 - By adjusting developer mindset

“Once you learn Rust, you are one with the borrow checker — it never leaves you. I now see many of the unsafe things I have been doing in other languages for years, (but probably not all of them, as I am human and not a compiler).”

Employer Concerns

- Specific to Rust:
 - Steep learning curve

Employer Concerns

- Specific to Rust:
 - Steep learning curve
 - Difficulty hiring Rust developers

“Do we really want to keep this thing in Rust? It’s hard to find a new person for the team. . . because we don’t have . . . a huge pool of Rust programmers.”

Participants' Advice

- Demonstrate value of Rust
 - Offers measurable improvement
- Be helpful and have a good support system
 - Willing and able to help new developers
 - Support system for new Rust developers

Takeaways

- Documentation, community, and feedback matter a lot!
- Steep learning curve can inhibit adoption
 - Pay now, but (maybe) benefit later
 - Flatten the learning curve?
 - Reduce the risk of investment

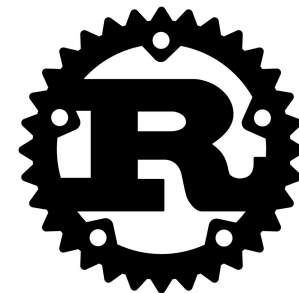
Questions:

kfulton@umd.edu

<http://www.cs.umd.edu/~kfulton/>

<https://sec-professionals.cs.umd.edu/>

Twitter - @kfulton121



**The Rust
Programming
Language**