Building More Secure Software with Memory-Safe Programming Languages

SHANE MILLER (SHE/HER)
NONRESIDENT SENIOR FELLOW, ATLANTIC COUNCIL
DISTINGUISHED ADVISOR, RUST FOUNDATION
Agenda

- What is “memory safety”?
- Success stories
- Memory safe choices and tradeoffs
Home Expectations
Regulations

- Encryption
- Least Privilege
- Logging
Secure by Design
Secure by Design Tradeoffs
Flexibility -> Complexity -> Bugs
“Memory safety” means making it very difficult for developers to create memory bugs.
The Cost of Safety
Safer "unsafe"?
Definitions

Vulnerability

Exploit
Trends

- 35 million active software developers in 2023
- 26,448 security vulnerabilities in 2022
- 59% more critical vulnerabilities in 2022 than 2021
- 4,135 critical vulnerabilities in 2022
Critical Vulnerabilities
Microsoft Security Research Center

https://msrc.microsoft.com/blog/2019/07/a-proactive-approach-to-more-secure-code/
Google Android

- 86% of critical severity vulnerabilities were memory safety bugs in 2022
- 89% of remotely exploitable vulnerabilities were memory safety bugs in 2022
- 78% of confirmed exploited vulnerabilities were memory safety bugs over the last several years

Strategic Security

Google Android Languages

Android 13 Languages

Android Native Code Languages

New Code By Language in Android 13

New Native Code

Google Android Vulnerabilities

Memory Safety

Critical Severity

Amazon Prime Video Languages

https://youtu.be/erdHTxghyM0
“Because we use Rust, we have a crash rate that is ten times smaller for the WebAssembly systems compared to the C++ systems. [...] On some devices, actually, the crash rate is most days zero with WebAssembly [Rust].”

Alexandru Ene
Principal Engineer, Amazon Prime Video
https://discord.com/blog/why-discord-is-switching-from-go-to-rust
### Size of programming language communities in Q1 2023

**Active software developers, globally, in millions**

<table>
<thead>
<tr>
<th>Language</th>
<th>Most popular in</th>
<th>Least popular in</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript*</td>
<td>Web, Apps for 3rd-party ecosystems</td>
<td>DS/ML/AI, Embedded</td>
</tr>
<tr>
<td>Java</td>
<td>Cloud, IoT devices</td>
<td>Web, DS/ML/AI</td>
</tr>
<tr>
<td>Python</td>
<td>DS/ML/AI, IoT apps</td>
<td>Web, Mobile</td>
</tr>
<tr>
<td>C/C++</td>
<td>Embedded, IoT apps</td>
<td>Cloud, Web</td>
</tr>
<tr>
<td>C#</td>
<td>Desktop, Games</td>
<td>IoT devices, DS/ML/AI</td>
</tr>
<tr>
<td>PHP</td>
<td>Web, Cloud</td>
<td>Mobile, DS/ML/AI</td>
</tr>
<tr>
<td>Visual   development tools</td>
<td>AR/VR, Games</td>
<td>Embedded, Cloud</td>
</tr>
<tr>
<td>Kotlin</td>
<td>Mobile, AR/VR</td>
<td>Desktop, DS/ML/AI</td>
</tr>
<tr>
<td>Swift</td>
<td>AR/VR, Mobile</td>
<td>Embedded, Cloud</td>
</tr>
<tr>
<td>Go</td>
<td>Cloud, AR/VR</td>
<td>Web, Mobile</td>
</tr>
<tr>
<td>Rust</td>
<td>AR/VR, Games</td>
<td>Mobile, Web</td>
</tr>
<tr>
<td>Objective C</td>
<td>AR/VR, IoT devices</td>
<td>Embedded, Desktop</td>
</tr>
<tr>
<td>Ruby</td>
<td>IoT devices, IoT apps</td>
<td>DS/ML/AI, Web</td>
</tr>
<tr>
<td>Lua</td>
<td>IoT devices, AR/VR</td>
<td>Mobile, Desktop</td>
</tr>
<tr>
<td>Dart</td>
<td>Mobile, Apps for 3rd-party ecosystems</td>
<td>Web</td>
</tr>
</tbody>
</table>

Rust

- Ownership
- Borrow Checker
fn admire(gift: &Gift) {
    println!("wow, this {} looks nice!", gift);
}

let gift = Gift::new();

admire(&gift);
Discord

Go

Rust
Discord

Go

Rust
Amazon Prime Video

https://www.amazon.science/blog/how-prime-video-updates-its-app-for-more-than-8-000-device-types
## Programming Language Tradeoffs

<table>
<thead>
<tr>
<th></th>
<th>Language Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Systems (C/C++)</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Garbage Collector (Java, Python, etc)</td>
</tr>
<tr>
<td>Usability</td>
<td>Compile-time Verification (Rust)</td>
</tr>
<tr>
<td>Cost to Run</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
</tr>
<tr>
<td>Memory Safe</td>
<td></td>
</tr>
</tbody>
</table>
Open Source Entropy
This is Not the End
Acknowledgements

Graydon Hoare
Creator of Rust
Safe programming researcher
Community advocate

Josh Aas
Leader of the memory safety movement
Founder and leader of the Internet Security Research Group (ISRG)
Thank you!

SHANE MILLER
HTTPS://SHANE-ONE.COM/
TWITTER: @SKIPPERSWIF
LINKEDIN: SHANEMILLERITMANAGER
MAIL: SHANE@SHANE-ONE.COM