



Presenting

Developing Device Drivers in Rust

School of Computing, Engineering & Physical Sciences

BSc (Honours) Computing Science

Supervisor: Paul Keir

Moderator: Stephen Devine



device drivers
the fragile

Image: David Carson

device drivers

- Control peripheral devices – interact with underlying hardware.
- Provide extensions to the Operating System.
- A necessity that suffers from a range of issues with dangerous consequences.

problems

- Continue to be written in C.
 - Originally developed 1969-1973.
 - Suffers from issues with memory safety.
- Memory safety can lead to critical vulnerabilities and is mostly present in C, C++ and Assembly

project

aim

overcome previously described
issues by developing a Linux
device driver in rust

rust for Linux

2019, Miguel Ojeda

introduce a new system programming language into
Linux kernel

memory safe language

- strong compiler
- borrow system
- variable lifetimes

rust

Stroustrup's criticism

"every safe language, including rust, has
loopholes allowing unsafe code"

memory safety

Accessing memory typically outside the bounds of a data structure which then provides a vector to attack from to conduct further exploitation/attack.

| | |
|---|---|
| Android | >65% of High & Critical security bugs |
| Android (bluetooth & media components) | 90% of vulnerabilities |
| iOS 12 | 66.3% of all vulnerabilities |
| MacOS Mojave | 71.5% of all vulnerabilities |
| Chrome | ~70% of serious security bugs |
| Microsoft | ~70% of CVE vulnerabilities |
| Firefox (css subsystem) | 73.9% of bugs |
| Ubuntu kernel | 65% of CVEs (In security updates between November and May 2020) |

(Alex Gaynor, 2020)

Each statistic is that of a large code base containing millions of lines of code.

All are written in C or C++.

Includes;

- Use-after-free
- Double-free
- Heap Buffer overflow
- Integer overflow
- Out-of-bounds read
- Out-of-bounds write

the great below

(progress)

Industry input has informed development and other aspects of project.

- Alex Gaynor
- Miguel Ojeda
- Jonathan Blow

(development)

- Have enabled Rust support on Linux 6.1 Machine.
- USB support is under development within RFL.
- Not all kernel subsystems are implemented.
- Conclusion seems to be that the project is still quite young but has a bright future.
- Demo: compilation & execution of 'Hello, World' in a Rust driver.

(findings)

Google & Android 13

Significant drop in memory safety vulns & severity.

Now 35% of total Android vulns (previously 76%).

Rust adopted in Android 12 as alt to C/C++.

0 memory safety vulnerabilities in Android Rust code.

Exo-kernel

Remove as many hardware abstractions as possible.

Allow the application to control its own memory resources.

Paging, Scheduling, Context, Faults.

Improve performance, efficiency, development, testing.

Enabling Rust support on Linux (6.1 and beyond)

Recompile
your existing
kernel to
Linux 6.1+.



Install and
configure Rust

Recompile this
new 6.1 kernel
with Rust
enabled.



Enjoy your new
Rust-y kernel!





underneath it all

these slides inspired by 'The Fragile' by Nine Inch Nails

images credited to David Carson/Nine Inch Nails

Kyle Christie

University of the West of Scotland

[2023]