Rex: Usable Operating System Kernel Extensions with Safe Rust

Jinghao Jia, Ruowen Qin, Michael V. Le, Hani Jamjoon, Hubertus Franke, Dan Williams, Tianyin Xu

Safe kernel extensions are taking over the OS

eBPF is the *de facto* safe kernel extension on Linux

Many advanced use cases beyond "packet filtering"

- Security, storage, scheduling, memory management, etc

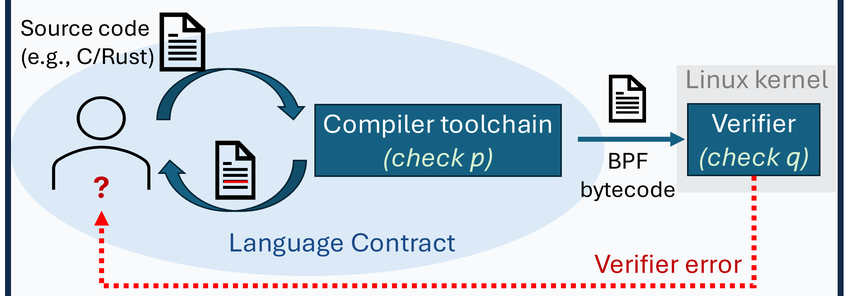
The Language-Verifier Gap

eBPF use static verification to ensure safety

- Symbolic-execution-based analysis on all code paths
- Checked at load time by an in-kernel verifier

Static verification creates **the language verifier gap!**

- Developers' expectation of safety is based on the language
- Verifier is not part of the language contract
  - Rejects safe programs and force user to take workarounds



Research Statement

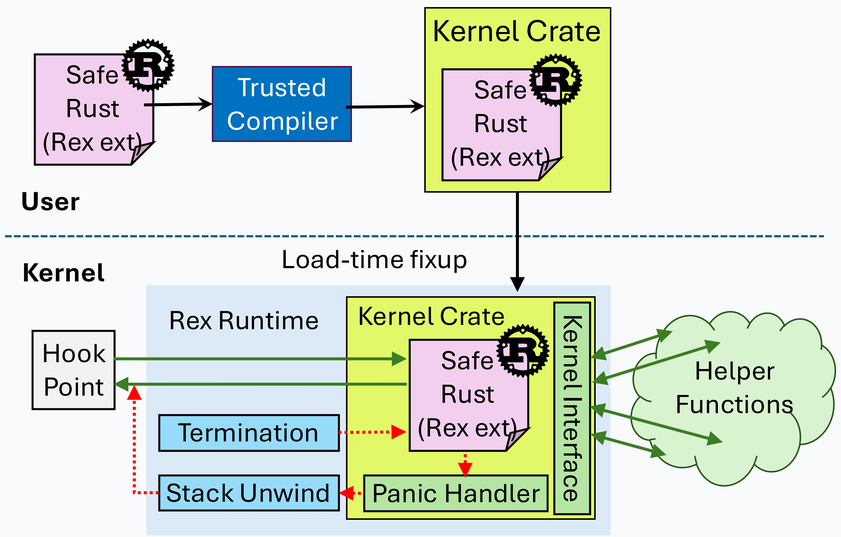
Building a **safe** and **usable** kernel extension framework

- build on language-based safety and runtime protection
- no language-verifier gap (with the same safety guarantee)
- clean, maintainable code for advanced extension programs

The Rex Extension Framework

Principle: Language-based safety + runtime protection

- Rust as the **safe language** (safe Rust only in Rex code)
- Runtime safety checks for other safety properties, e.g.,
  - Proper program termination
  - Safe Kernel stack usage



Rex provides the same safety guarantee as eBPF

- **Memory safety**
  - Access memory with correct lifetime and size
- **Extended type safety**
  - Allow safe extraction typed data from packet payload bytes
- **Safe resource management**
  - Correctly release acquired kernel resources through RAII
- **Safe exception handling**
  - Clean up resources and gracefully exit upon Rust panics
- **Kernel stack safety**
  - Avoid overflowing of the limited, fix-sized kernel stack
- **Safe termination**
  - Prevent long-executing programs from holding the CPU

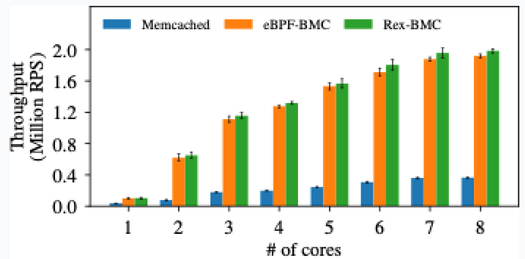
Evaluation

Usability

- Eliminated existing eBPF verifier workarounds
  - No language-verifier gap anymore
- Implemented the BPF Memcached Cache (BMC)
  - **Much cleaner, simpler code**
    - 326 lines of Rust code vs. 513 lines of C code

Performance

- Rex stack check is **3x faster** than eBPF tail-calls
- Map access in Rex incurs **small overhead** (<7 ns)
- Rex-BMC brings **5.4x speedup** for Memcached



Broader Impacts

- Rex as an open-source project https://github.com/rex-rs/rex
- Presentation at **Open Source Summit (OSS)**
- Used in undergrad OS courses (CS 423)
- Used as a project to engage CDS undergrads

Anticipated Benefits to IBM

- IBM used eBPF and kernel extensions in many products (including our collaboration).
- Our goal is to make Rex be the **next-generation kernel extension mechanism** of emerging use cases for real-world industry products.