Paralegal: Practical Static Analysis for Privacy Bugs

Justus Adam, Carolyn Zech, Livia Zhu, Sreshtaa Rajesh, Nathan Harbison Mithi Jethwa,
Will Crichton, Shriram Krishnamurthi, Malte Schwarzkopf

Brown University

Problem

Existing tools are not practical enough to help developers find subtle privacy bugs.

Practical tools must meet these challenges:

1. Support for relevant privacy policies
2. No changes to code, behavior & performance
3. Policy robust against source code changes
4. Support libraries w/o needing complex models

Approach	1. Flexible	2. Non-invasive	3. Robust	4. Libraries
IFC	✗	✗	✓	✗
Specialized Bug Finder	✗	✓	✓	✓
Runtime Checks	✓	✗	✓	✓
Code Query Engine	✓	✓	✗	✗
Paralegal	✓	✓	✓	✓

Policy: expressive first-order logic policies with legal-document-like syntax

Succinct and auditable

```
Somewhere:
1. For each "data type" marked user_data:
  A. There is a "source"
      that produces "data type" where:
    a. There is a "delete query" marked deletes:
      i) "source" goes to "delete query"
```

Markers: Semantically meaningful terms that connect policy and application code

```
#[paralegal::marker(user_data)]
struct Post { ... }
#[paralegal::marker(user_data)]
struct Comment { ... }
```

Policy does not need to change when code does

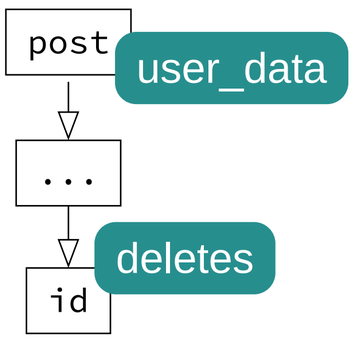
Real bug in Plume app: Devs forgot to delete `Comment`s

Link

Application Code

Receives

Representation: Program Dependence Graph with markers



Handles library modeling, scaling and FFI

Paralegal

Emits

Fixes

Application Developer

Paralegal

```
'Deletion Policy' not satisfied.
No entrypoints satisfied rule 1.A

Entrypoint `delete_user`
  Did not satisfy Rule 1.A
    There is a "source"
    that produces "data type"

  for the "data type" type: `Comment` (Rule 1)
```

Approximation: Rusts types let Paralegal approximate a PDG for any function.

```
fn HashMap::get_mut<'a, 'k, K, V>(&'a mut self,
        key: &'k K) -> Option<'a mut V>
```

- `mut` tells Paralegal only `self` is modified, not `key`
- lifetimes `'a` and `'k` tell Paralegal that `self` and `key` cannot alias

Paralegal finds subtle privacy bugs in real code bases by

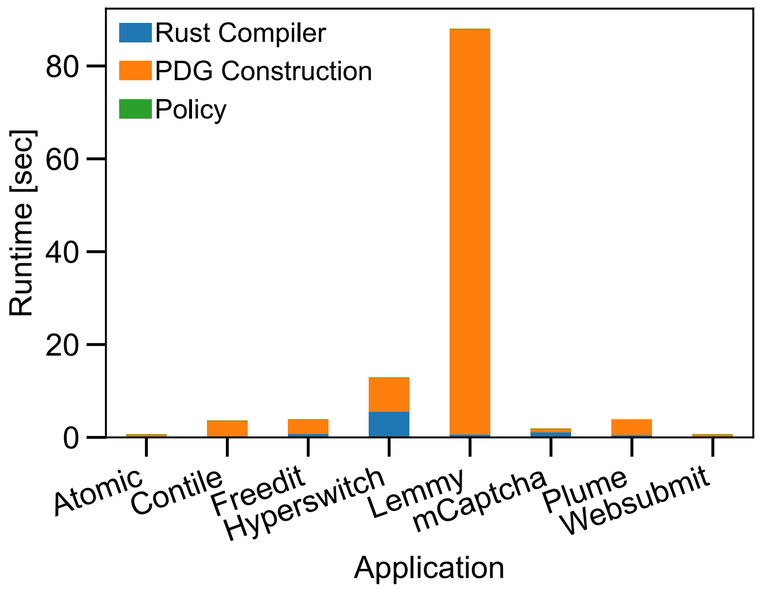
1. **separating concerns between policy writers and developers; and**
2. **leveraging Rust for precise approximations**

Evaluation

Setup: 8 Applications and **11 Policies** (Access Control, Purpose Limitation, Data Deletion, Credential Security, …)

Bugs: Found **7** subtle bugs. **2** new, **5** previously known

Related Work Comparison: IFC fails to express half of the policies. CodeQL only finds half the bugs



Performance

- Full codebase check: <90 seconds
  ⟹ fast enough for CI
- Incremental check: <5 seconds
  ⟹ fast enough for IDE

Paralegal is open source.

github.com/brownsys/paralegal