Loom

Weaving Instrumentation for Program Analysis

Brian Kidney (Presenter)
Jonathan Anderson
Memorial University

But Instrumentation is done, right?

Why another instrumentation tool

- There are lots of instrumentation tools
  - Intel Pin, XRay, CSI, DTrace…
- Most tools focus on performance
- We needed something different
  - No custom compiler frontend or backend
  - “Non-traditional” instrumentation
  - The ability to transform code when needed
- Our focus was security
  - We wanted something more general-purpose

What if you wanted to instrument every PAM Authentication?

Instrumenting PAM

And you could do it with this:

```
strategy: callout
dtrace: userspace
functions:
- callee: [entry]
  metadata:
    name: auth
    id: 1
  name: pam_authenticate
```

Instrumenting PAM

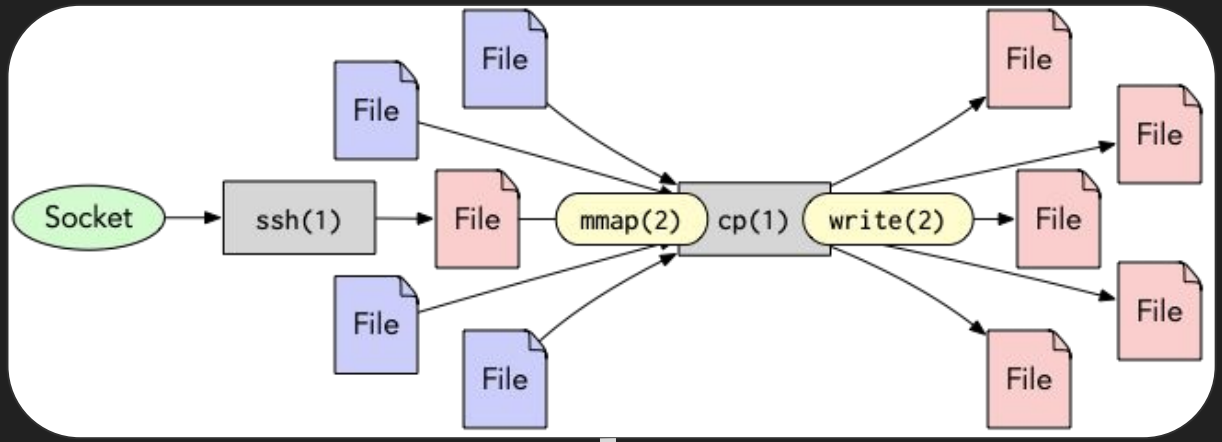
And you could do it with this:

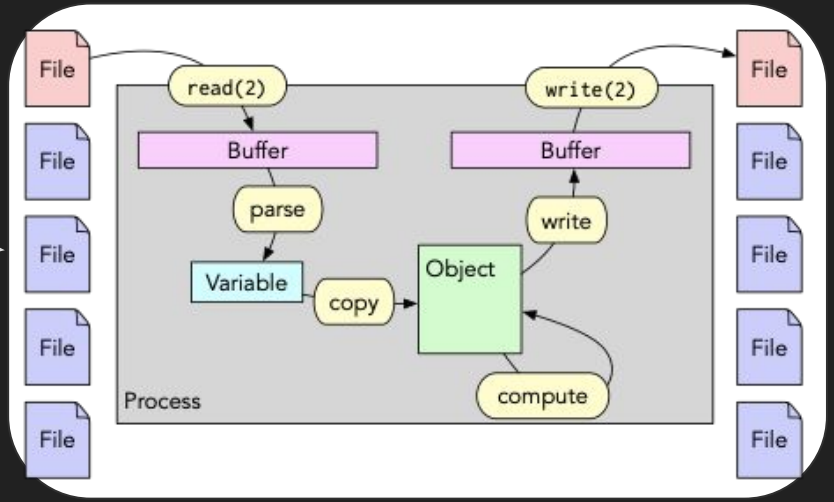
```
strategy: callout
dtrace: userspace
functions:
- callee: [entry]
  metadata:
    name: auth
    id: 1
  name: pam_authenticate
```

And you get this:

```
dtrace: script './pam.d' matched 6 probes
CPU     ID                    FUNCTION:NAME
  2  61725                    none:dt-probe
Pam Authentication - execname: sshd, user: bkidney
  0  61725                    none:dt-probe
Pam Authentication - execname: sshd, user: steve
  2  61725                    none:dt-probe
Pam Authentication - execname: sshd, user: badguy
```

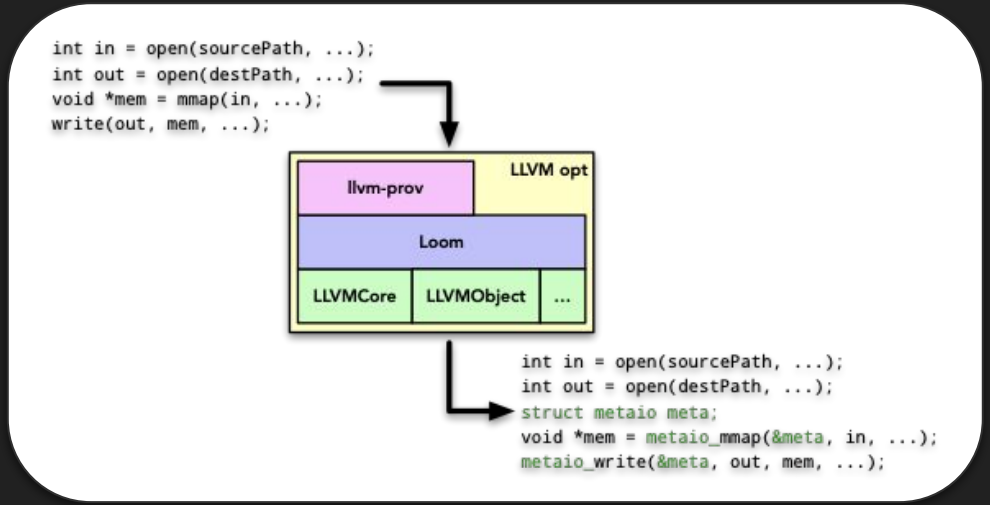
What if you wanted to transform system calls for provenance instrumentation?


File
File
File
File
File
File
File
File
File
File
Socket
ssh(1)
mmap(2)
cp(1)
write(2)


File
File
File
File
File
File
File
File
File
File
read(2)
write(2)
Buffer
Buffer
parse
write
Variable
copy
Object
compute
Process

llvm-prov

- Too complex an application for the policy file
- Transforms system call API to new API
- Domain specific logic to determine code of interest
- Loom emits code transformation



Loom

- Simple Policy files to apply instrumentation without code modification
- Framework to build custom tools when you need more

Loom

- Simple Policy files to apply instrumentation without code modification
- Framework to build custom tools when you need more

Currently supports instrumenting:

- Struct fields
- Functions
  - Callee / Caller
- Global Variables
- Pointer Instructions

Supported Outputs:

- Logging
  - KTrace, DTrace, Text, Json, XML
- Code transformation

Loom

- Simple Policy files to apply instrumentation without code modification
- Framework to build custom tools when you need more

- Work has started on instrumentation language
  - DAG matching for code transformation

Currently supports instrumenting:

- Struct fields
- Functions
  - Callee / Caller
- Global Variables
- Pointer Instructions

Supported Outputs:

- Logging
  - KTrace, DTrace, Text, Json, XML
- Code transformation

Come see more
during poster session!