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

<table>
<thead>
<tr>
<th>CPU</th>
<th>ID</th>
<th>FUNCTION:NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>61725</td>
<td>none:dt-probe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pam Authentication - execname: sshd, user: bkidney</td>
</tr>
<tr>
<td>0</td>
<td>61725</td>
<td>none:dt-probe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pam Authentication - execname: sshd, user: steve</td>
</tr>
<tr>
<td>2</td>
<td>61725</td>
<td>none:dt-probe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pam Authentication - execname: sshd, user: badguy</td>
</tr>
</tbody>
</table>
What if you wanted to transform system calls for provenance instrumentation?
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!