A Verified Information Flow Type Checker

Tom Magrino & Matthew Milano
CS6113 Fall 2013

Motivation

- Always desireable!
- ..but hasn't been done (we think)
- How hard is it (scale to JIF?)

Approach

- Simplified Pottier and Simonet's CoreML[2]
  - Effectively STLC w/o parametric polymorphism
  - For now, no references or exceptions
- Coq proofs of:
  - Type checking
  - Label checking

Example Program

```
let (h:int<High>) = 0 in
let (l:int<Low>) = 1 in
if h == 0 then 1 else 0
```

- Simple labeled types (2 labels)
- This should not type check
  - Leaks h!

Example Program

```
let (h:int<High>) = <0 | 1> in
let (l:int<Low>) = 1 in
if h == 0 then 1 else 0
```

- Basic trick: prove evaluation does not result in <a | b> term
  - Reduction rules for <a | a> → a

Goals and Future

- Goal: learn what's difficult
- Next: Exceptions, references, richer types and labels.
- Eventually: verified mature IF language, Jif/FlowCaml