CS 5430

Information-Flow Control

Elisavet Kozyri
Spring 2017
Review: Labels represent IF policies

A label $\ell$ on some data represents an IF policy.

Possible labels for confidentiality:

• Classifications
  – Unclassified (U), Confidential (C), Secret (S), Top Secret (TS)
  – Low (L), high (H)

• Sets of principals
  – \{Alice, Bob\}, \{Alice\}, \{Bob\}, {}
Review: Noninterference

• Given a program, and
• given a mapping from variables to labels,
• it usually suffices to enforce noninterference.
Today: Information Flow Control

• **Goal**: Enforce IF policies that tag variables in a program.

• There is a mapping $\Gamma$ from variables to labels, which represent desired IF policies.

• The enforcement mechanism should ensure that a given program and a given $\Gamma$ satisfy noninterference.
Information Flow Control

![Diagram showing information flow control with inputs, outputs, variables, label, and program.]
Information Flow Control: fixed $\Gamma$

- $\Gamma$ remains the same during the analysis of the program.
- The mechanism checks that $\Gamma$ satisfies noninterference.
- The program is rejected, if at least one red arrow appears in the program.
Information flow control: flow-sensitive $\Gamma$

- $\Gamma$ may change during the analysis of the program.
- The mechanism deduces $\Gamma(x)$, $\Gamma(y)$, $\Gamma(z)$ such that noninterference is satisfied.
- The program is never rejected.
Enforcing IF policies

• Static mechanism
  – Checking and/or deduction of labels before execution.

• Dynamic mechanism
  – Checking and/or deduction of labels during execution.

• Hybrid mechanism
  – Combination of static and dynamic.
Enforcement mechanism for today:

• Static
• Fixed $\Gamma$
  – So, for a program, the mechanism only needs to check whether $\Gamma$ satisfies noninterference (NI).
Programs are written using this syntax:

\[
e ::= x \mid n \mid e_1 + e_2 \mid \ldots
\]

\[
c ::= x ::= e
\mid \text{if } e \text{ then } c_1 \text{ else } c_2
\mid \text{while } e \text{ do } c
\mid c_1; c_2
\]
Checking an assignment

\[ x := y \]

Examples for confidentiality

- \( \Gamma(x) \) is L.
  \( \Gamma(y) \) is L.
  Does this assignment satisfy NI?

- \( \Gamma(x) \) is H.
  \( \Gamma(y) \) is L.
  Does this assignment satisfy NI?

- \( \Gamma(x) \) is L.
  \( \Gamma(y) \) is H.
  Does this assignment satisfy NI?
Order relation on labels

• $\ell \sqsubseteq \ell'$ iff $\ell'$ is at least as restrictive as $\ell$.
• Values in variables tagged with $\ell$ may flow to variables tagged with $\ell'$.
• Examples (for confidentiality):
  – $L \sqsubseteq H$
  – $\{Alice\} \sqsubseteq \emptyset$
  – $\{Alice, Bob\} \sqsubseteq \{Alice\}$
• Relation $\sqsubseteq$ should be:
  – reflexive, transitive, and antisymmetric.
• There is a label $\bot$ (bottom) that is less restrictive than all other labels.
• There is a label $\top$ (top) that is more restrictive than all other labels.
Checking an assignment

Assignments cause explicit flows of values.

\[ x := y \]

It satisfies NI, if \( \Gamma(y) \subseteq \Gamma(x) \).
Checking an assignment: connection with MLS

\[ x := y \]

It satisfiesNI, if \( \Gamma(y) \sqsubseteq \Gamma(x) \).

**MLS for confidentiality**

“no read up”:

\[ S \text{ may read } O \text{ iff Label}(O) \sqsubseteq \text{Label}(S) \]

“no write down”:

\[ S \text{ may write } O' \text{ iff Label}(S) \sqsubseteq \text{Label}(O') \]
Checking an assignment: connection with MLS

\[ x := y \]

It satisfies NI, if \( \Gamma(y) \sqsubseteq \Gamma(x) \).

**MLS for confidentiality**

“no read up”:

CPU may read \( y \) iff \( \text{Label}(y) \sqsubseteq \text{Label}(\text{CPU}) \)

“no write down”:

CPU may write \( x \) iff \( \text{Label}(\text{CPU}) \sqsubseteq \text{Label}(x) \)
Checking an assignment

\[ x := y + z \]

It satisfies NI, if \( \Gamma(y) \sqsubseteq \Gamma(x) \) and \( \Gamma(z) \sqsubseteq \Gamma(x) \).

It satisfies NI, if \( \Gamma(y+z) \sqsubseteq \Gamma(x) \).

???
Operator for combining labels

- For each \( \ell \) and \( \ell' \), there should exist label \( \ell \sqcup \ell' \), such that:
  - \( \ell \sqsubseteq \ell \sqcup \ell' \), \( \ell' \sqsubseteq \ell \sqcup \ell' \), and
  - if \( \ell \sqsubseteq \ell'' \) and \( \ell' \sqsubseteq \ell'' \), then \( \ell \sqcup \ell' \sqsubseteq \ell'' \).

- \( \ell \sqcup \ell' \) is called the \text{join} of \( \ell \) and \( \ell' \).

- Operator \( \sqcup \) is associative and commutative.
Checking an assignment

\[ x := y + z \]

It satisfies NI, if \( \Gamma(y) \cup \Gamma(z) \subseteq \Gamma(x) \).
Lattice of labels

• The set of labels and relation ⊑ define a lattice, with join operator ⊔.
• Examples (confidentiality):

```
H
⊔
{Alice} {Bob}
{Alice, Bob}
⊔
L
```
Checking an if-statement

if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3

Examples for confidentiality

<table>
<thead>
<tr>
<th>Γ(x) is L.</th>
<th>Γ(y), Γ(z) is L.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does this if-statement satisfy NI?</td>
<td>😊</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Γ(x), Γ(z) is L.</th>
<th>Γ(y) is H.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does this if-statement satisfy NI?</td>
<td>😞</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Γ(x), Γ(y) is L.</th>
<th>Γ(z) is H.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does this if-statement satisfy NI?</td>
<td>😞</td>
</tr>
</tbody>
</table>
Checking an if-statement

if z>0 then
    if y>0 then x:=1 else x:=2
else
    x:=3

Conditional commands (e.g., if-statements and while-statements) cause \textit{implicit} flows of values.
Context

if \( z > 0 \) then
  if \( y > 0 \) then \( x := 1 \) else \( x := 2 \)
else
  \( x := 3 \)

They reveal information about \( z > 0 \) and \( y > 0 \).
Context label $ctx$

if $z > 0$ then
  if $y > 0$ then $x := 1$ else $x := 2$
else
  $x := 3$

Its $ctx$ is $\Gamma(z)$.

Their $ctx'$ is $\Gamma(z) \cup \Gamma(y)$.
Context label $ctx$

if $z > 0$ then
  if $y > 0$ then $x := 1$ else $x := 2$
else
  $x := 3$

Check if $ctx' \sqsubseteq \Gamma(x)$, where $ctx' = \Gamma(z) \cup \Gamma(y)$.
Context label $ctx$

if $z > 0$ then
  if $y > 0$ then $x := e$ else $x := 2$
else
  $x := 3$

Check if $ctx' \sqcup \Gamma(e) \subseteq \Gamma(x)$.

Implicit flow

Explicit flow
Typing system for IF control

• Static
• Fixed $\Gamma$
• Labels as types
  – Label $\Gamma(x)$ is the type of $x$.
• Typing rules for all possible commands.
• **Goal:** type-correctness $\implies$ noninterference
We are already familiar with typing systems!

Example of typing rule from Java or OCaml:

\[
x + y : \text{int} \\
\text{if } x : \text{int} \\
\text{and } y : \text{int}
\]
Typing rules for expressions

Judgement $\Gamma \vdash e : \ell$

- According to mapping $\Gamma$, expression $e$ has type (i.e., label) $\ell$.

Constant: $\Gamma \vdash n : \bot$

Variable: $\Gamma \vdash x : \Gamma(x)$

Expression: $\Gamma \vdash e + e' : \ell \sqcup \ell'$

  if $\Gamma \vdash e : \ell$

  and $\Gamma \vdash e' : \ell'$
Typing rules for expressions

Expression: $\Gamma \vdash e + e' : \ell \sqcup \ell'$
  
  if $\Gamma \vdash e : \ell$
  and $\Gamma \vdash e' : \ell'$

Inference rule:

Premises $\Gamma \vdash e : \ell \quad \Gamma \vdash e' : \ell'$

Conclusion $\Gamma \vdash e + e' : \ell \sqcup \ell'$
Example

- Let $\Gamma(x) = L$ and $\Gamma(y) = H$.
- What is the type of $x + y + 1$?
- *Proof tree:*

\[
\begin{align*}
\Gamma(x) &= L & \Gamma(y) &= H \\
\Gamma \vdash x : L & \quad \Gamma \vdash y : H & \quad \Gamma \vdash 1 : L \\
\hline
\Gamma \vdash x + y + 1 : H
\end{align*}
\]
Typing rules for commands

Judgement $\Gamma,ctx \vdash c$

- According to mapping $\Gamma$, and context label $ctx$, command $c$ is type correct.
Assignment rule

\[ \Gamma, ctx \vdash x := e \]

if \( \Gamma \vdash e : \ell \)
and \( \ell \sqcup ctx \sqsubseteq \Gamma(x) \)

\[ \Gamma \vdash e : \ell \quad \ell \sqcup ctx \sqsubseteq \Gamma(x) \]

\[ \Gamma, ctx \vdash x := e \]
If-rule

\[ \Gamma \vdash e : \ell \quad \Gamma, \ell \cup \text{ctx} \vdash c_1 \quad \Gamma, \ell \cup \text{ctx} \vdash c_2 \]

\[ \Gamma, \text{ctx} \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \]
If-rule (example)

\[ \Gamma, \Gamma(z) \cup L \vdash \begin{cases} \text{if } y > 0 \text{ then } x := 1 \\ \text{else } x := 2 \end{cases} \]

\[ \Gamma \vdash z > 0 : \Gamma(z) \]

\[ \Gamma, \Gamma(z) \cup L \vdash x := 3 \]

\[ \Gamma, L \vdash \begin{cases} \text{if } z > 0 \text{ then } \{ \text{if } y > 0 \text{ then } x := 1 \text{ else } x := 2 \} \\ \text{else } \{ x := 3 \} \end{cases} \]

What is the relation between \( \Gamma(x) \), \( \Gamma(y) \), and \( \Gamma(z) \), such that the above judgement can be proved?
If-rule (example)

$$\Gamma \vdash y > 0 : \Gamma(y) \quad \Gamma, \Gamma(z) \vdash x := 1$$

$$\Gamma, \Gamma(z) \cup \Gamma(y) \vdash x := 2$$

$$\Gamma \vdash z > 0 : \Gamma(z) \quad \Gamma, \Gamma(z) \vdash \text{if } y > 0 \text{ then } x := 1 \quad \Gamma, \Gamma(z) \vdash x := 3$$

else x := 2

$$\Gamma, \Gamma(z) \cup \Gamma(y) \vdash \text{if } y > 0 \text{ then } x := 1 \text{ else } x := 2$$

else {x := 3}

$$\Gamma, L \vdash \text{if } z > 0 \text{ then } \{\text{if } y > 0 \text{ then } x := 1 \text{ else } x := 2\}$$

else {x := 3}
If-rule (example)

\[
\begin{align*}
\Gamma, \Gamma(z) \cup \Gamma(y) & \vdash x := 1 \\
\Gamma & \vdash y > 0 : \Gamma(y) & \Gamma, \Gamma(z) \cup \Gamma(y) & \vdash x := 2 \\
\Gamma & \vdash z > 0 : \Gamma(z) & \Gamma, \Gamma(z) & \vdash \text{if } y > 0 \text{ then } x := 1 \text{ else } x := 2 \\
\Gamma, \Gamma(z) & \vdash x := 3
\end{align*}
\]
If-rule (example)

\[ \Gamma(z) \cup \Gamma(y) \subseteq \Gamma(x) \quad \Gamma(z) \cup \Gamma(y) \nvdash x:=1 \]
\[ \Gamma(z) \cup \Gamma(y) \subseteq \Gamma(x) \quad \Gamma(z) \nvdash x:=2 \]
\[ \Gamma(z) \subseteq \Gamma(x) \quad \Gamma(z) \vdash x:=3 \]

\[ \Gamma, \Gamma(z) \cup \Gamma(y) \vdash \text{if } z>0 \text{ then } \{ \text{if } y>0 \text{ then } x:=1 \text{ else } x:=2 \} \]
\[ \quad \text{else } \{ x:=3 \} \]
If-rule (example)

\[ \Gamma(z) \cup \Gamma(y) \subseteq \Gamma(x) \]
\[ \Gamma, \Gamma(z) \cup \Gamma(y) \vdash x:=1 \]

What is the relation between \( \Gamma(x), \Gamma(y), \) and \( \Gamma(z) \), such that the above judgement can be proved?
If-rule (example)

\[ \Gamma(z) \sqcup \Gamma(y) \subseteq \Gamma(x) \]

\[ \Gamma, L \vdash \text{if } z > 0 \text{ then } \{ \text{if } y > 0 \text{ then } x := 1 \text{ else } x := 2 \} \text{ else } \{ x := 3 \} \]
while-rule

\[
\begin{align*}
& \Gamma \vdash e : \ell \\
\hline
& \Gamma, \ell \cup \text{ctx} \vdash c
\end{align*}
\]

\[
\Gamma, \text{ctx} \vdash \text{while } e \text{ do } c
\]
Sequence rule

\[ \Gamma,ctx \vdash \text{c1} \quad \Gamma,ctx \vdash \text{c2} \]

\[ \Gamma,ctx \vdash \text{c1;}\text{c2} \]
Sequence rule (example)

\[ \Gamma, \ell \sqcup \Gamma(e) \vdash x := 1 \quad \Gamma, \ell \sqcup \Gamma(e) \vdash x := 2 \]

\[ \Gamma, \ell \vdash \text{if } e \text{ then } \{x := 1\} \]
\[ \quad \text{else } \{x := 2\} \]

\[ \Gamma, \ell \vdash \text{if } e \text{ then } \{x := 1\} \text{ else } \{x := 2\}; \quad x := 3 \]
Theorem

Type correctness $\Rightarrow$ Noninterference
Upcoming events

• [May 10] A6 due
• [May 18] Final exam

A type system is the most cost effective unit test you’ll ever have. – Peter Hallam