Beyond Labels: Permissiveness for Dynamic Information Flow Enforcement

Elisavet Kozyri[*], Fred B. Schneider[*], Andrew Bedford[†], Josée Desharnais[†] and Nadia Tawbi[†]

Cornell University [*], Laval University[†]

32nd IEEE Computer Security Foundations Symposium

An observation

The designer of a

- dynamic,
- flow-sensitive,
- permissive, and
- sound

information flow mechanism
is forced to think about:

label
label
label
label
label

This paper

- Enforcement mechanisms for label chains.
  - Block executions that are deemed unsafe.
  - No leak through enforcement actions (label chains deduction, blocking).
    - Strong threat model: *observation* produced on updates to variables and labels during normally terminated and blocked executions.
- Theorems that relate length of label chains to permissiveness.

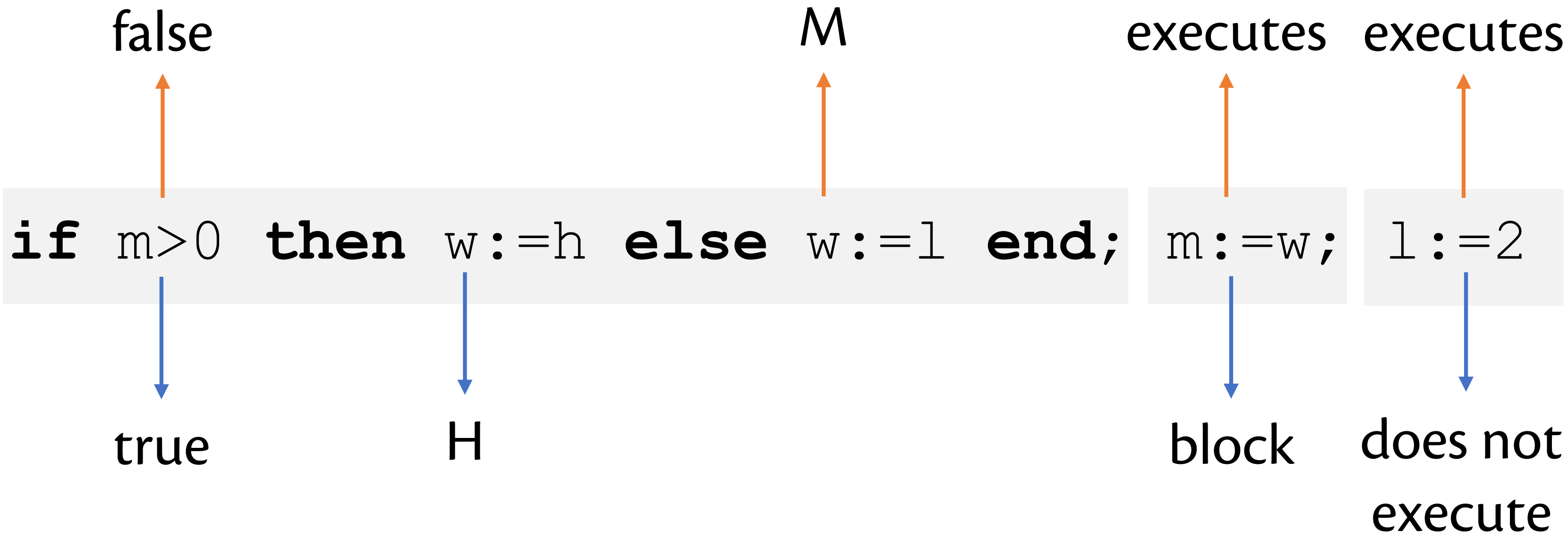
More observations on variables and labels in chains $\Rightarrow$ Increased permissiveness

Example

```
if m>0 then w:=h else w:=l end; m:=w; l:=2
```

- Lattice of labels: $L \sqsubset M \sqsubset H$
- Constants are tagged with L.
- *Anchor* variable `l` is tagged with *fixed* L; `m` with M; `h` with H.
- *Flexible* variable `w` is tagged with a *flow-sensitive* label.

A dynamic analysis



`m` leaks to:

- Principals reading variable `l`.
- Principals reading the flow-sensitive label of `w`.

Strong Threat Model

Metalabels represent sensitivity of labels

false M

```
if m>0 then w:=h else w:=l end; m:=w; l:=2
```

true H

But, what is the sensitivity of the metalabel of `w`?

Label chains

- A variable x is associated with label chain

$$\langle \ell_1, \ \ell_2, \ldots, \ \ell_i, \ldots \rangle$$

$$T(\mathrm{x}) \quad T^2(\mathrm{x}) \quad\quad T^i(\mathrm{x})$$

$T^{i+1}(\mathrm{x})$ is the sensitivity of $T^i(\mathrm{x})$.

- Flexible variable: the entire label chain is updated at every assignment
- Anchor variable: $T(\mathrm{x})$ is fixed

$$\langle \ell_1, \bot, \ldots, \bot, \ldots \rangle$$

- Monotonically decreasing: $\ell_1 \sqsupseteq \ell_2 \sqsupseteq \cdots \sqsupseteq \ell_i \sqsupseteq \cdots$

Why monotonically decreasing label chains?

Consider, instead, a non-monotonically decreasing label chain for $\mathtt{x}$:

$$\langle \mathsf{L}, \mathsf{H}, \ldots \rangle$$

- Principals assigned label L are authorized to read the value in $\mathtt{x}$.
- When read access to $\mathtt{x}$ succeeds, these principals conclude that $T(\mathtt{x})$=L.
- So, principals assigned L learn the value of $T(\mathtt{x})$, even though the sensitivity of $T(\mathtt{x})$ is H.

Enforcer ∞-Enf: assignment to flexible variable

ctx of C is set $\{\mathtt{b},\ \mathtt{b'}\}$:

```
if b then
  if b' then
    C
```

$\mathtt{w:=e}$ in ctx

$T(\mathtt{w}) \coloneqq \underbrace{T(\mathtt{e}) \sqcup T(ctx)}$ in ctx

$T^2(\mathtt{w}) \coloneqq \underbrace{T^2(\mathtt{e}) \sqcup T^2(ctx) \sqcup T(ctx)}$ in ctx

$T^i(\mathtt{w}) \coloneqq T^i(\mathtt{e}) \sqcup T^i(ctx) \sqcup \cdots \sqcup T^2(ctx) \sqcup T(ctx)$ in ctx

Enforcer ∞-Enf: assignment to flexible variable

$$\forall i\colon\ T^i(\mathtt{w}) \coloneqq T^i(\mathtt{e})\ \sqcup\ T^i(ctx) \sqcup \cdots \sqcup T^2(ctx) \sqcup T(ctx)$$

But, due to monotonically decreasing label chains, simplifies to:

$$\forall i\colon\ T^i(\mathtt{w}) \coloneqq T^i(\mathtt{e}) \sqcup T(ctx)$$

Enforcer ∞-Enf: assignment to anchor variable

block condition G

```
…
C;
a:=e;
C';
…
```

ctx

Enforcer ∞-Enf: assignment to anchor variable

```
...
C;
a:=e;
C';
...
```

block condition G → $ctx \cup \{G\}$

$$\boldsymbol{G} \quad \Rightarrow \quad (\,T(\mathtt{e}) \sqcup T(ctx) \sqcup T(\boldsymbol{G}) \sqsubseteq T(\mathtt{a}))$$

Given monotonically decreasing label chains, a solution for G is :

- $T(\mathtt{e}) \sqcup T(ctx) \sqsubseteq T(\mathtt{a})$

Enforcer ∞-Enf: assignment to anchor variable

...
`C;`
Block Unless $\overbrace{\boldsymbol{T}(\mathbf{e}) \sqcup \boldsymbol{T}(\boldsymbol{ctx}) \sqsubseteq \boldsymbol{T}(\mathbf{a})}^{G}$
`a:=e;`
$ctx \cup \{G\}$ → `C';`
...

It prevents leaks though blocking.

Given monotonically decreasing label chains, a solution for G is :

- $T(\mathtt{e}) \sqcup T(ctx) \sqsubseteq T(\mathtt{a})$

∞-Enf satisfies Block-safe Noninterference (BNI)

No leak through variables, label chains, and blocking.

Observations Enforcer Lattice Command

k-BNI(E, L, C)

$$\forall \ell \in L : \forall M, M' :$$

$$M|_\ell = M'|_\ell$$

$$\wedge\ \tau = trace_E(C, M) \text{ is finite}$$

$$\wedge\ \tau' = trace_E(C, M') \text{ is finite}$$

$$\Rightarrow \tau|_\ell^{\boxed{k}} =_{obs} \tau'|_\ell^{\boxed{k}}$$

Termination Insensitive NonInterference

$\tau = trace_E(C, M)$ terminates normally
$\tau' = trace_E(C, M')$ terminates normally

Termination Sensitive NonInterference

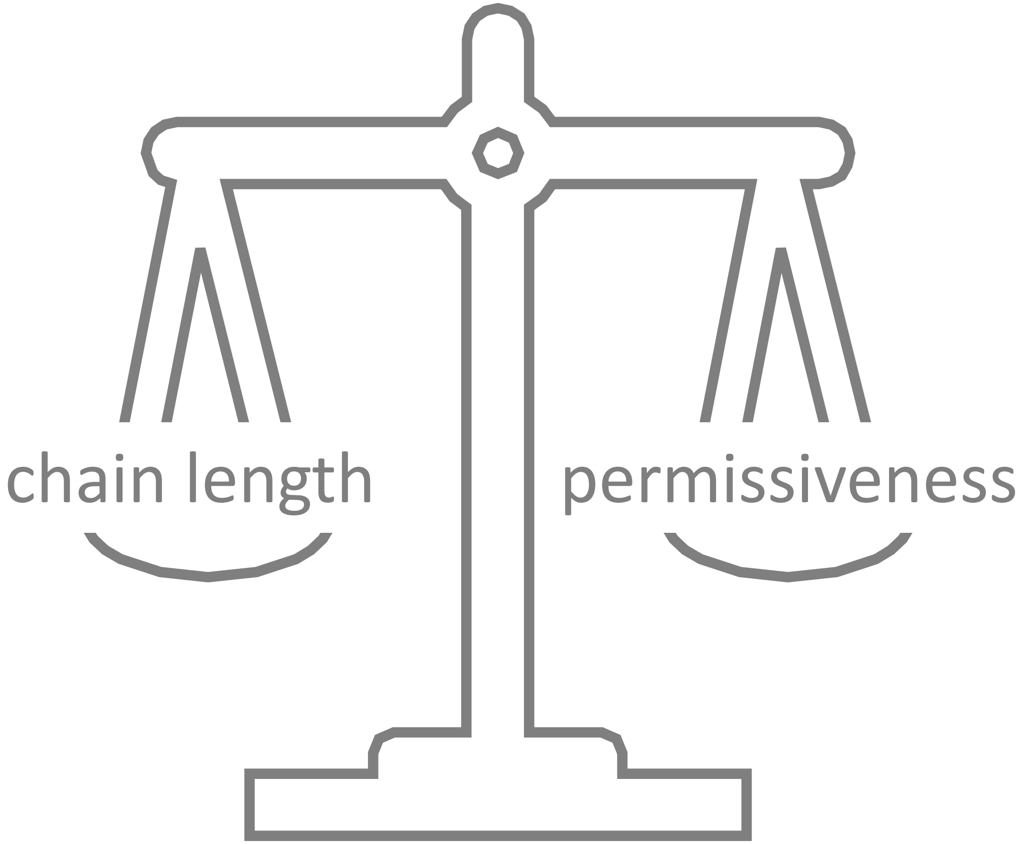
$\tau = trace_E(C, M)$
$\tau' = trace_E(C, M')$

Enforcer k-Enf

For $k \geq 2$:

- k-Enf is based on ∞-Enf to compute the first k labels of chains.
- k-Enf conservatively approximates the sensitivity of $T^k(x)$ to be itself:
  - $\boxed{T^{k+1}(x) = T^k(x).}$
- k-Enf generates observations for the first k labels.
- [Thm] k-Enf satisfies BNI.
- k-Enf conservatively approximates
  - $\langle \ell_1, \ell_2, \dots, \ell_k, \ell_{k+1}, \dots \rangle$ with
  - $\langle \ell_1, \ell_2, \dots, \ell_k, \ell_k \rangle$.

What is it lost when shorter chains approximate longer chains?



Permissiveness

- An enforcer E is *at least as permissive as* an enforcer E', iff
  - Traces of E are at least as long as E', and
  - E produces at most as restrictive label chains as E'.
- So, E generates at least as many observations on variables and labels as E'.

Permissiveness is lost when shorter chains approximate longer chains

- Assume enforcers E and E' satisfy BNI.
- E produces label chain Ω for flexible variable `w` at a particular program point.
- E' produces label chain Ω' for `w` at that program point.

$$\begin{array}{ll} \Omega: & \langle \ell_1, \ell_2, \dots, \ell_i \overset{\sqsupset}{,} \ell_{i+1}, \dots, \ell_k \rangle \\ & \qquad\qquad\qquad\qquad\quad \sqcap \qquad\quad \sqcap \\ \Omega': & \langle \ell_1, \ell_2, \dots, \ell_i, \quad \ell_i, \dots, \ell_i \rangle \end{array}$$

Loss of permissiveness

k-precise with $\ell_i \sqsupset \ell_{i+1}$

i-dependent

Does such an Ω arise?

Ω: $\langle \ell_1, \ell_2, \dots, \ell_i, \ell_{i+1}, \dots, \ell_k \rangle$

k- *precise* with
$\ell_i \sqsupset \ell_{i+1}$

- Arbitrary initialization.
  - Ω can be associated with flexible variable $\mathtt{w}$ at initialization.
- Common initialization.
  - All flexible variables are initially associated with $\langle \bot, \bot, \dots, \bot \rangle$.
  - [Thm] We have designed an enforcer that can associate Ω with $\mathtt{w}$ during execution of a command.
    - Optimization of k-Enf.

So, Ω can arise!

- Assume enforcers E and E' satisfy BNI.
- E produces label chain Ω for flexible variable `w` at a particular program point.
- E' produces label chain Ω' for `w` at that program point.

$$\begin{array}{ll} \Omega: & \langle \ell_1, \ell_2, \ldots, \ell_i, \ell_{i+1}, \ldots, \ell_k \rangle \\ & \qquad\qquad\qquad\quad\;\; \sqcap \qquad\;\; \sqcap \\ \Omega': & \langle \ell_1, \ell_2, \ldots, \ell_i, \quad \ell_i, \ldots, \ell_i \rangle \end{array}$$

k-precise with $\ell_i \sqsupset \ell_{i+1}$

i-dependent

Loss of permissiveness

[Thm] E' cannot be as permissive as E.

Changing threat model

- Strong threat model:
  - Principals observe updates to variables and labels in chains.
- Weakened threat model:
  - Principals only observe updates to variables.

How does the relation between permissiveness and label chain length change?

Weakened threat model

- [Thm] Enforcers that use label chains of length one are not at least as permissive as 2-Enf for lattice $\langle \{L, M, H\}, \sqsubseteq \rangle$.
  - With 2-Enf, the second label in a label chain enables the decision to block assignments to be more permissive.
- Open question: Are label chains with more than two elements useful under the weakened threat model?

Two-level lattice

- For the weakened threat model, one label is enough:
  - [Thm] Permissiveness is not lost comparing to 2-Enf.

From 30,000 feet…

- To increase permissiveness, we add metadata.
- But metadata might encode sensitive information.
- To prevent leaks without harming permissiveness, add more metadata.
- But storage is finite.
- So, there are storage VS permissiveness trade-offs.

Summary: longer label chains provide increased permissiveness

		Chain length > 1	Chain length > 2
Strong	Arbitrary Initialization	✓	✓
	Common Initialization	✓	✓
Weakened (common initialization)	3-level lattice	✓	?
	2-level lattice	✗	✗