Nondeterminism and Information Flow

ca. 1986-1997

Michael Clarkson
CS 711
10/29/2003

Limited Chronology

- 1986 – Nondeducibility (ND)
- 1987 – Generalized noninterference (GNI)
- 1988 – Forward correctability (FC)
- 1990 – Restrictiveness (RES),
  Flow model (FM),
  Nondeducibility on strategies (NDS)
- 1994 – Separability (SEP)
- 1997 – Perfect security property (PSP)

Relative Power

- Depends on exact definitions and assumptions
  - These vary widely, especially for GNI
- Under input totality:
  SEP ⇒ PSP ⇒ RES ⇒ FC ⇒ GNI
  [Zakinthinos & Lee 97]
- Without input totality:

[Mantel 02]

Nondeducibility
[Sutherland 86]

Low-security users should not be able to deduce *with certainty* anything about the activities of high-security users

- Cannot rule out any high activities
- S is secure when:
  - For all $l \in$ l-traces(S), $h \in$ h-traces(S), interleavings(l,h) $\subseteq$ traces(S)

Nondeducibility

Can also formulate in terms of statistical independence:

$Pr(l) > 0$ and $Pr(h) > 0 \Rightarrow Pr(h \mid l) > 0$

$\equiv$

$Pr(l) > 0$ and $Pr(h) > 0 \Rightarrow Pr(l \mid h) > 0$

Nondeducibility

Problems with nondeducibility:

- Disallows some safe flows
  - e.g., auditing: flows from L to H [McLean 90]
- High events can still affect low events [McCullough 87]
- Not preserved under feedback/composition
  - Based on single traces, not sets

Generalized Noninterference
[McCullough 87]

- Changes in high level input events do not cause changes in low level events
  - Fixes bug in ND
  - If all inputs happen before all outputs, generalizes:

$$\sigma_1 \approx_L \sigma_2 \Rightarrow [\![S]\!]\ \sigma_1 \approx_L [\![S]\!]\ \sigma_2$$

    - Domain of $[\![S]\!]$ becomes sets of states
    - $\approx_L$ becomes low-view set equality

Generalized Noninterference

If inputs and outputs are interleaved:

For all $t \in traces(S)$, s = change-high-input($t1$),
There exists $t' \in traces(S)$ s.t.
$t = uw$, $s = uw'$, $t' = uw''$ & $w' \approx_{\sim HO} w''$

Generalized Noninterference

- Not preserved under composition:
  - Machine A:
    - on receive H input: echo to H output
    - on receive "reset" (L input):
      - echo to L output
      - cancel all pending H outputs
      - if there were none, nondeterministically choose
        - » output "nothing to reset"
        - » no output
  - Machine B:
    - Same as A except no echo of "reset" to L output

Generalized Noninterference

- A and B satisfy GNI
  - Because of ND choice, can't tell whether change in H inputs has caused any change in L outputs

- **Machine A:**
  - **on receive "reset":**
    - **echo to L output**
    - **cancel all pending H outputs**
    - **if there were none, nondeterministically choose**
      - » **output "nothing to reset"**
      - » **no output**
- **Machine B:**
  - **Same as A except no echo of "reset" to L output**

Generalized Noninterference

- Composition of A and B doesn't satisfy GNI

Odd number of H inputs makes it impossible for both machines to output "nothing"

- **Machine A:**
  - **on receive "reset":**
    - **echo to L output**
    - **cancel all pending H outputs**
    - **if there were none, ND choose**
      - » **output "nothing to reset"**
      - » **no output**
  - **on receive "nothing", echo to L output**
- **Machine B:**
  - **Same as A except no echo of "reset" to L output**

Composition Paradox

- Individual systems secure; composition insecure
- Composition of safety, liveness properties well-understood [Alpern & Schneider 85]
- Why not security?
  - Many security properties outside of safety/liveness domain [McLean 94]

Composition Paradox

- Can we develop a theory of composition?
  - [McLean 94], [Zakinthinos & Lee 97], [Mantel 00, 02]
- Before these, there were various *ad hoc* compositional properties
- Not the only paradox…

Refinement Paradox

- A ND program is secure, but a refinement of it is not:

```
l := h [] 0 [] 1;   // secure
l := h [] 0;        // insecure
```

- Recent work [Zdancewic & Myers 03] shows how to solve this problem
- More on this later in the course

Restrictiveness
[McCullough 90]

- Fix GNI so that it is composable
- Recall GNI is:

  For all $t \in$ traces(S), s = change-high-input(t1),
  There exists $t' \in$ traces(S) s.t.
  $t = uw$, $s = uw'$, $t' = uw''$ & $w' \approx_{\sim HO} w''$
- Changing H inputs may require introducing new H outputs
- After composition, new outputs from A become secret inputs to B
- Result: cascade of changes

Restrictiveness

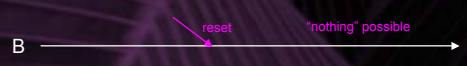
- McCullough's solution:
  - Restrict exchange of messages
  - Results in composability

For all $t \in$ traces(S), s = change-high-input(t1),
There exists $t' \in$ traces(S) s.t.
$t = uw$, $s = uw'$, $t' = uw''$ & $w' \approx_{\sim HO} w''$
} GNI

& $w = xy$, $w' = x'y'$, $w'' = x'y''$
& x,x' contain only inputs
} RES

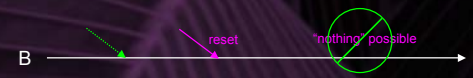
Restrictiveness

- Reject systems that require:
  - Insertion of new H output
  - Before/during the sequence of inputs after 1st change in H input
- Machine B from GNI was not restrictive:



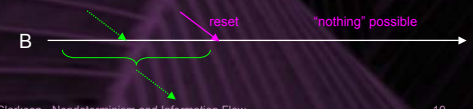
Restrictiveness

- Reject systems that require:
  - Insertion of new H output
  - Before/during the sequence of inputs after 1st change in H input
- Machine B from GNI was not restrictive:



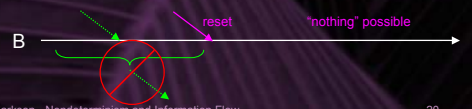
Restrictiveness

- Reject systems that require:
  - Insertion of new H output
  - Before/during the sequence of inputs after 1st change in H input
- Machine B from GNI was not restrictive:



Restrictiveness

- Reject systems that require:
  - Insertion of new H output
  - Before/during the sequence of inputs after 1st change in H input
- Machine B from GNI was not restrictive:



Flow Model
[McLean 90]

- Recall another bug in ND:
  - ND disallows some safe flows
    - e.g., auditing: flows from L to H
  - Results from symmetry of independence:
    $Pr(h \mid l) = Pr(h) \equiv Pr(l \mid h) = Pr(l)$

Flow Model

- Solution: break the symmetry using time
  - Not ok for h_t to affect l_{t+1}
  - But ok for l_t to affect h_{t+1}
- "Affect": statistical and causal dependency

Flow Model

- FM requires that l_t be independent of $h_{0..t-1}$:
  $Pr(l_t \mid l_{0..t-1} \,\&\, h_{0..t-1}) = Pr(l_t \mid l_{o..t-1})$
- High events can be correlated with low events if caused by previous low event

Flow Model

- GNI and its extensions ignore causal dependencies
  - Overly restrictive
- FM allows more useful programs than GNI, ND, etc.
- But still ensures high level of security
- FM later extended to quantitative information flow [Gray 91]

Nondeducibility on Strategies
[Wittbold and Johnson 90]

- Recall problem with ND
  - Nondeducible on a single run
  - Leaks information every *n* runs
  - Strategy exists to leak information
- Can require system to be nondeducible on any strategy
  - Formulated using information theory
  - System is NDS iff there exist no noiseless communication channels

Separability
[McLean 94]

- Absolutely no possibilistic information flow
- Like running the system as two separate, non-communicating processes
  - One process for each security level
- Few (useful) systems can satisfy this property

PSP
[Zakinthinos & Lee 97]

- Weaken SEP to allow high outputs to depend on low events
  - But all high inputs still possible
- Flow occurs when some high trace is not possible
  - Construction guarantees that low user can't tell what that trace is
- Provably weakest such property
- Also composable

Conclusion

- Not too hard to generalize information flow to nondeterministic systems
- Hard to find balance between security and utility

- Coming up: restricting, quantifying nondeterminism