WOLFASI 2004
Turku, Finland July 12-13, 2004

Towards an Algebraic Approach to Solve Policy Conflicts

Cataldo Basile
< cataldo.basile @ polito.it >

Antonio Lioy
< antonio.lioy @ polito.it >

Politecnico di Torino

Dip. Automatica e Informatica

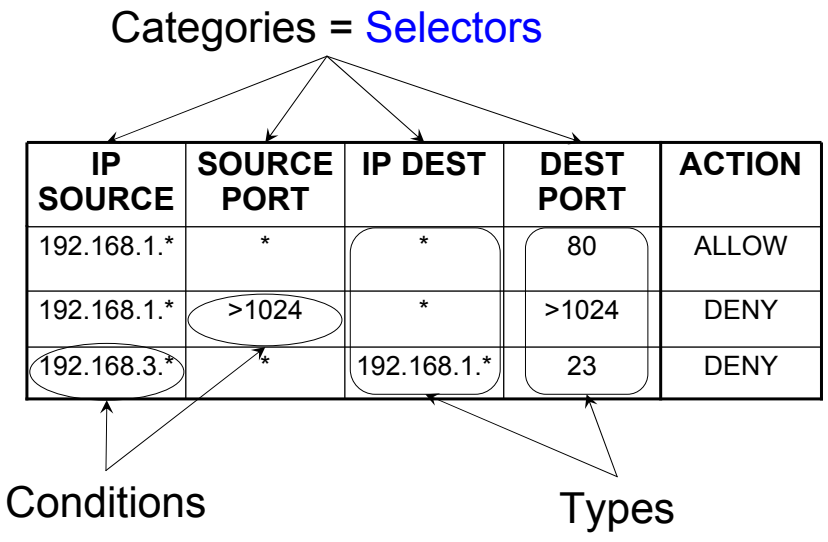
Motivations

- **policy must drive the behaviour of the systems**
  - inconsistencies lead to errors or unknown states
- **automatic decision relies on the conflict resolution**
  - Adaptive Security
  - Policy Analyzer always able to select a new action

Conflict Resolution

- **the problem of finding an action to be applied when rules conflict**
  - different strategies
    - Deny/Permit Take Precedence
    - Most/Least Specific Take Precedence
- **conflict detection**
  - run-time, off-line
  - models required

Conflict Detection: the model



Conditions

- **a condition defines the set of allowed values**
  - a subset c of a selector S

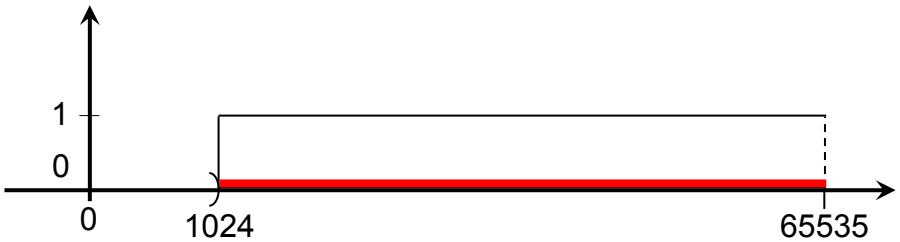
TCPDestPort > 1024

S = TCPDestPort
Type = [0,65535] $\subset \mathbb{N}$

Evaluation of a Condition

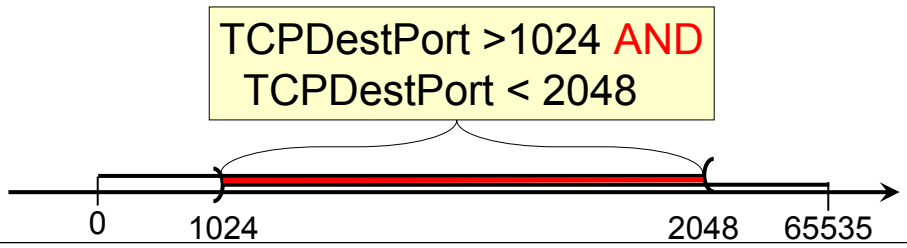
- **the evaluation is done by means of characteristic functions**

$$\chi_c : \quad S \rightarrow \{0,1\}$$
$$x \mapsto \begin{cases} 1 & \text{if } x \in c \\ 0 & \text{otherwise} \end{cases}$$



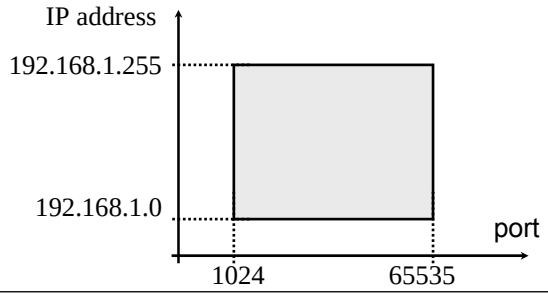
Boolean Algebras

- **the set of all the subsets of a given set forms a boolean algebra**
- **in a selector S:**
  - intersection ↔ AND
  - union ↔ OR
  - set minus ↔ NOT



Conditions Belonging to Different Selectors

- **if two selectors are non overlapping ($S_1 \neq S_2$)**
  - impossible to use above boolean algebras
- **need of extension to cartesian products**
  - if $c_1 \subseteq S_1$ AND $c_2 \subseteq S_2 \Longrightarrow c_1 x\, c_2 \subseteq S_1 x\, S_2$

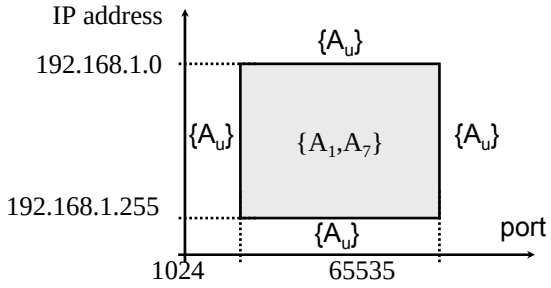


Selection Condition

- **if the number of Selectors is finite and the order is defined**
  - policy rules domain: $S_1 x S_2 x \ldots x S_n = \mathcal{S}$
- **conditions in DNF/CNF are subsets C of $\mathcal{S}$**
  - C is called selection condition
- **the subsets of $\mathcal{S}$ form a boolean algebra**
  - AND ↔ intersection of selection conditions
  - OR ↔ union of selection conditions
  - negative selection condition $\neg C$ is the subset $\mathcal{S} \backslash C$

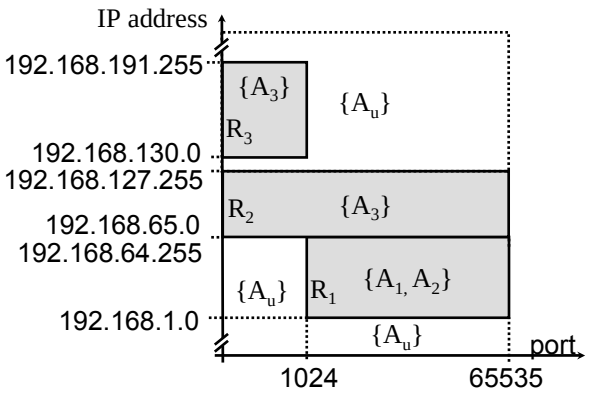
Policy Rules

- **actions may be collected in an actions set $\mathcal{A}$**
- **a policy rule defines a mapping**
  - from $\mathcal{S}$ to a subset of the actions set $\mathcal{A}$
  - the action is undefined (A_u) out of the selection condition



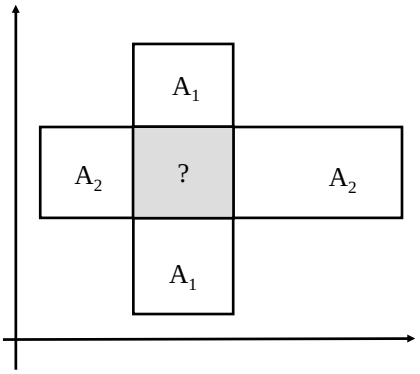
Policy Function

- **a policy is applied by means of a set of rules**
  - superposition of many rules



Policy Conflicts

- **a conflict between two rules occurs if**
  - their selection conditions intersect, and
  - they specify different actions



Conflict Resolution: hypothesis

- **conflict resolution and management is a difficult task**
- **select a concrete case**
  - rules in "**if** *conditions* **then** *actions*" form
  - finite set of actions
  - resolution methods based on the action clause
- **the action to be applied where rules conflict**
  - function of the action clauses:

$$f(\{A_{j_1}\}_{j_1 \in J_1}, \{A_{j_2}\}_{j_2 \in J_2})$$

Conflict Resolution: simple case

- **if the action clauses contain only one action**
  - the problem is reduced to

$$f(A_1, A_2)$$

- **it is possible to define a binary operation in the actions set $\mathcal{A}$**

$$A_1 \circ A_2 = A_{new}$$

- **look at the structure that $\mathcal{A}$ assumes when the resolution strategy "$\circ$" is defined**

Properties of the Binary Operation

- **associative**

$$\forall a,b,c \in \mathcal{A} : a \circ (b \circ c) = (a \circ b) \circ c$$

- **commutative**

$$\forall a,b \in \mathcal{A} : a \circ b = b \circ a$$

- **idempotent**

$$\forall a \in \mathcal{A} : a \circ a = a$$

- **an algebraic structure ($\mathcal{A}$, $\circ$) satisfying these axioms is a semi-lattice**
  - a partially ordered set in which any subset has the least upper bound

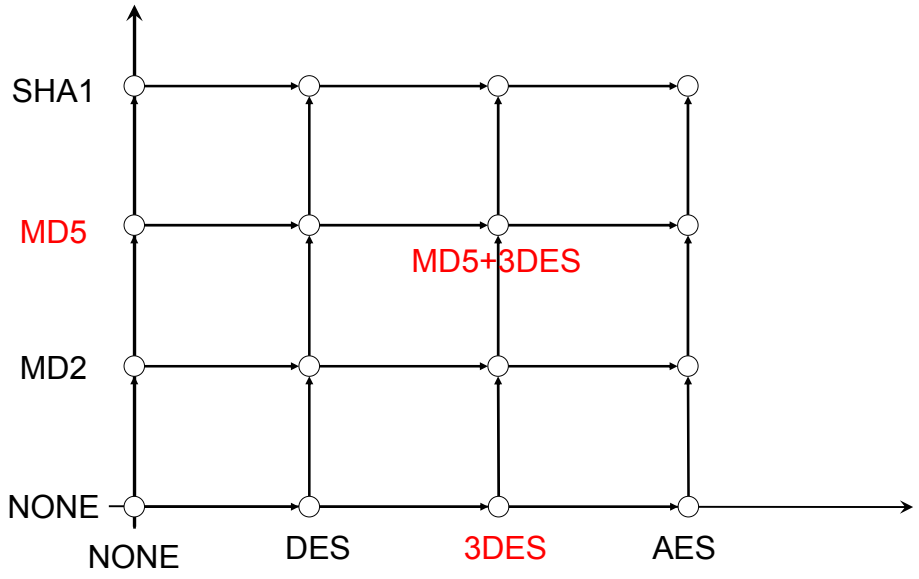
Conflict Resolution: the role of semi-lattices

- **if an action set is a semi-lattice**
  - it is always possible to find a new action to solve conflicts
- **if an automatic decision is required**
  - the actions set must be a semi-lattice
- **the property of being a semi-lattice is necessary and sufficient**
  - the actions set must be completed to a semi-lattice

More Complex Examples: AND of actions

- **if the action clause is composed by a set of actions**
  - semi-lattice considerations hold
- **AND of actions generates a bigger actions set**
  - A_1 AND A_2 more restrictive than A_1 OR A_2
  - from $(\mathcal{A}, \circ)$ to $(\mathcal{A}^{\wedge}, \circ)$
    - A_1 AND A_2 as a new atomic action
    - complete this new set $\mathcal{A}^{\wedge}$ to a semi-lattice

An Example



More Complex Examples: OR of actions

- **allow for the selection of an action from a set**
  - definition of a new operation * in $2^{\mathcal{A}^\wedge}$
    - union of all the compositions (by means of "$\circ$") of elements in the first subset with elements of second subset
  - not all the subsets can be taken into account
    - because of the idempotence
  - only the sub-semi-lattices of $(2^{\mathcal{A}^\wedge}, \circ)$
- **further work is needed**
  - to characterize composition of complex forms of aggregation

Conclusions

- **model of policy useful in conflict detection**
- **analysis of resolution strategy based on the action clause**
  - semi-lattice property of the actions sets
  - application to AND-ed and OR-ed sets of actions
- **further work**
  - a tool for conflict resolution simulations
  - application to practical cases
  - extension to general resolution methods
  - map semi-lattices to ordered sets of rules

Additional Considerations

- **intersections**
  - a general concept
- **cartesian products**
  - to combine different domains
- **use of algebras**
  - obtain an algebraic structure for each set of rules + conflict resolution method
    - obtain a semi-lattice of rules for each device
  - more convenient methods to store and process rules databases
    - semi-lattices representable as graphs