A Network Access Control Tool

Joshua D. Guttman, Ph. D.

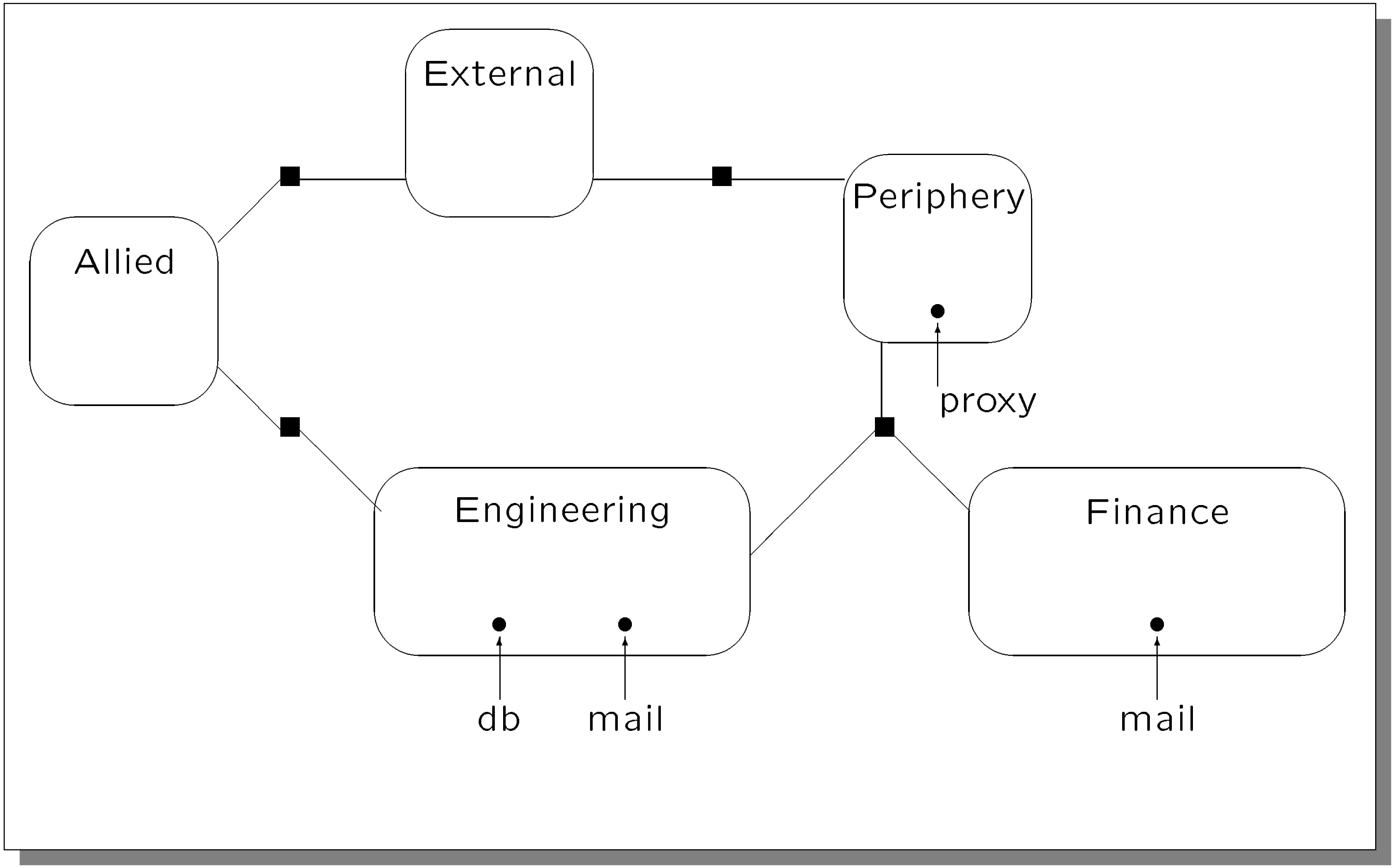
August 1996

MITRE

Network Access Control: The Problem

- Given:
  - The network resources of an enterprise
  - Security goals (external attacks to prevent)
- Goal:
  - Devise useful network access control policy
  - Implement it using filtering routers & firewalls
- Ingredients in solution:
  - Survey routers, hosts, services (using Tkined e.g.)
  - Develop/analyze security policy for filtering
  - Implement policy by configuring filters
  - Test implementation for compliance with policy

Corporate Protection Problem



MITRE

To Develop a Policy

- Divide network into regions
- Use firewalls/routers
  - To separate regions
  - To control access to regions
- Select constraints on which datagrams may be transmitted between regions
- Express these constraints in the form:
  - If datagram was in region R_1 and later reaches R_2, then its source/destination address fields and its source/destination port fields are required to say ...
- Select a policy constraint for each pair of regions

MITRE

Network Regions

- Region: A contiguous set of physical networks, together with the hosts and router interfaces on them
- Filtering routers provide links between regions
- Regions are the basic unit of protection
  - No protection at smaller granularity
  - Policy describes what packets travel between regions
- Strictly speaking: A region is a set of interfaces
  - Distinct regions must be disjoint
  - Every machine (host or router) with interfaces on more than one region must be known

Network Access Policies: Our Proposal

- A policy statement:
  - If packet p was in region R_1 and later in R_2, then p satisfies ϕ
- Simple examples: ϕ may be
  - Unsatisfiable: p never gets from R_1 to R_2
  - Destination port is 25: p must be SMTP
  - Destination is h: p must be addressed to h (e.g. a bastion host)
- Policy: a policy statement for each pair of distinct regions
- Policy is global:
  - Describes what is permitted through using any path whatever

Policy and Filtering Posture

- A router has only local information:
  - Incoming interface
  - Outgoing interface
  - Datagram source/destination address fields and port fields
    - Fields subject to spoofing/interception
- Filtering posture:
  - A filter at each router interface
  - Each router must make its own local contribution to enforcing the global policy
- Posture enforces policy ϕ for R_1, R_n:
  - For every path $\langle R_1 \ldots R_n \rangle$, for every p violating ϕ, some filter on the path discards p

MITRE

Policy and Posture: What had to be done?

- A little language for expressing:
  - Network configuration:
    - Regions, routers, hosts, services
  - Rules: Policies, filters

  expressed in a logical notation
- Symbolic computation:
  - Efficient manipulation of rules
  - Apply rules symbolically to all packet types
- Graph algorithms traverse all paths
  - Apply symbolic manipulation of rules
  - Calculate global consequences of local postures

From Policy to Filtering Posture

Efficient graph algorithms:

- Create filtering posture certain to enforce a given policy
  - Resulting posture may be too restrictive
  - Manual improvements may be needed
- Check a filtering posture
  - Determine all policy violations, or report that none exist
  - Useful for manually modified postures
- Check "service specifications"
  - Ensure that needed services have a route through the filters
  - (Currently unimplemented)

MITRE

Representing Rules

- **Host set: Zero or more regions, omitting zero or more hosts from them, adding zero or more other hosts**
  - **Boolean operators easily defined**
- **Rule:** $\langle sources, dests, services \rangle$
  - $sources$, $dests$ **are host sets**
  - $services$ **defines network services (telnet, smtp, . . .)**
  - R **applies to** p **if** $p.src \in sources$ **and** $p.dst \in dests$
  - R **allows** p: R **applies to** p **and** $serv(p) \in services$
  - R **disallows** p: R **applies to** p **and** $serv(p) \notin services$
- **Domain of applicability of a rule is a "rectangle"**
- **Rules used to build policy statements, filter clauses**

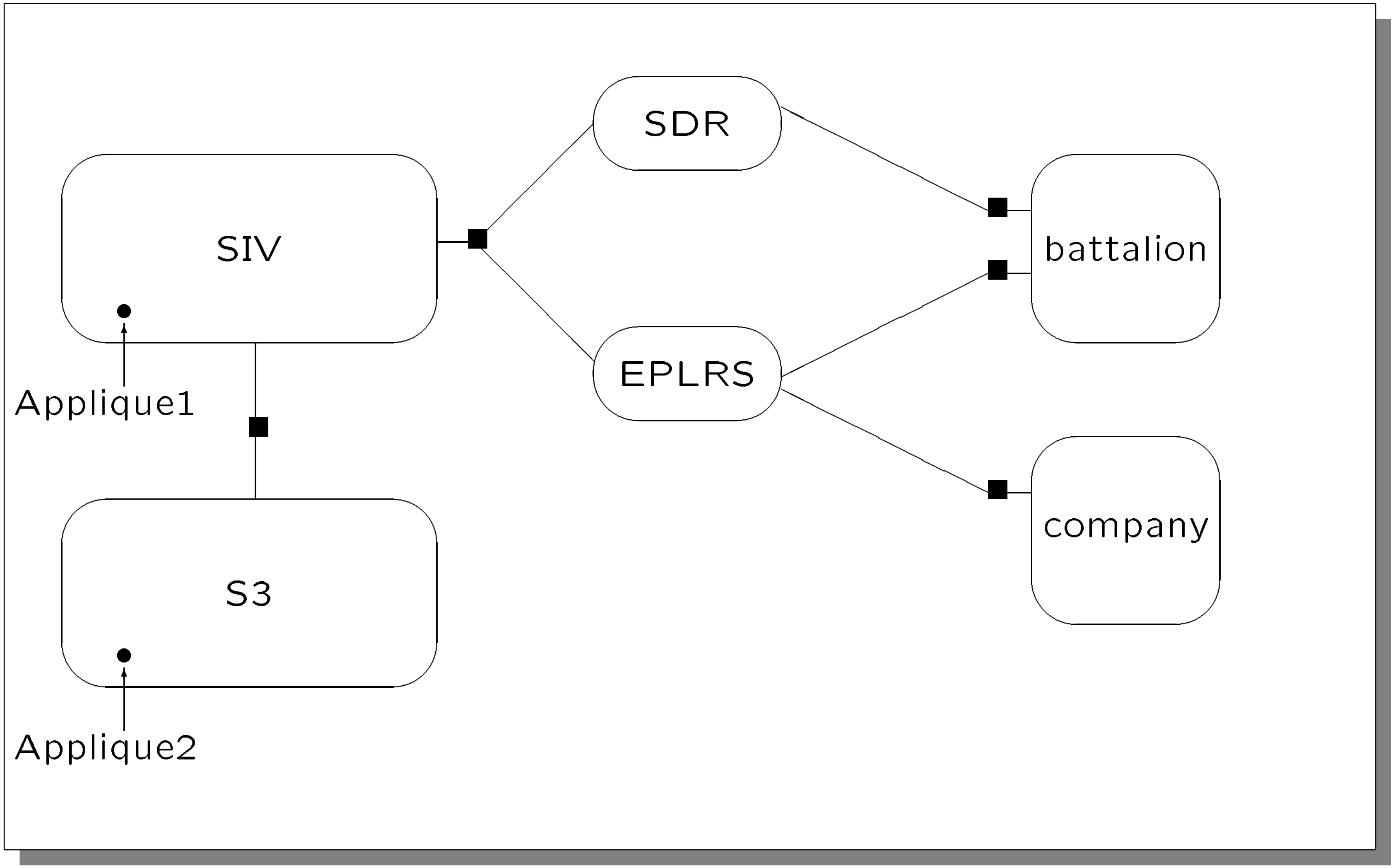
Rule-sets

- Rule-set $\mathcal{R}$: At most one $R \in \mathcal{R}$ applies to any packet
  - Rectangles of applicability disjoint
  - $\mathcal{R}$ applies to p if any $R \in \mathcal{R}$ does
- Operators on rule sets $\mathcal{R}_1, \mathcal{R}_2$:
  - Domain of result: Union of domains
  - Packet p allowed if
    - Loosen: either rule-set applies and allows p
    - Tighten: any rule-sets that apply allow p
    - Conjoin: both rule-sets do apply and allow p

    etc.
- Implementation: Keeping the rectangles of applicability disjoint was the hard part

MITRE

Tactical Communications Problem



MITRE

Test Suite Development

- Case-by-case simulation
  - Use Tkined as enhanced by MITRE
  - Simulate paths of selected datagrams
  - Discover hosts a datagram can reach
- Test suite for unit testing
  - Derive comprehensive set of test datagrams from region-based policy
  - Predict intended results from verified filtering posture
  - Validate vendor-specific filter configuration files on live router

Policy and Filtering Posture: Conclusion

- Express access control policy explicitly
- Use network topology in graph-based algorithms
- Have the right level of abstraction, to allow informative and efficient symbolic computation
- Must still:
  - Integrate with network survey tools
  - Integrate with work to generate router configuration files
  - Design unit test suites