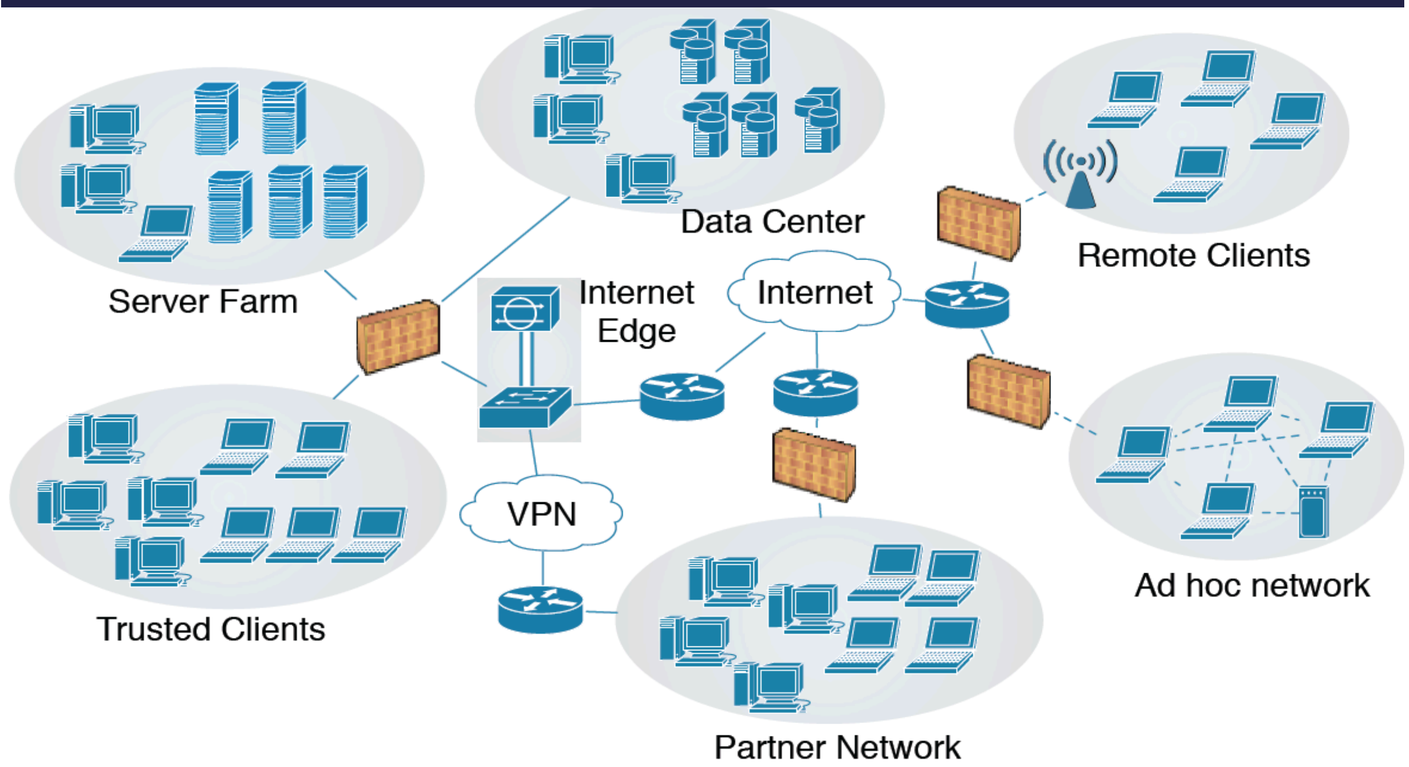
Automating Network Monitor Placement in Large-Scale Networks

Nirupama Talele, Jason Teutsch, Trent Jaeger and Robert F. Erbacher

Abstract

- The existing techniques for placing network monitoring and intrusion prevention apparatuses in a network do not account for host flows and fail to defend against vulnerabilities resulting from minor modifications to host configurations.
- We propose a method to compute network monitor placements that leverages commonality in available access control policies across hosts to compute network monitor placement for large-scale systems.

Example Problem



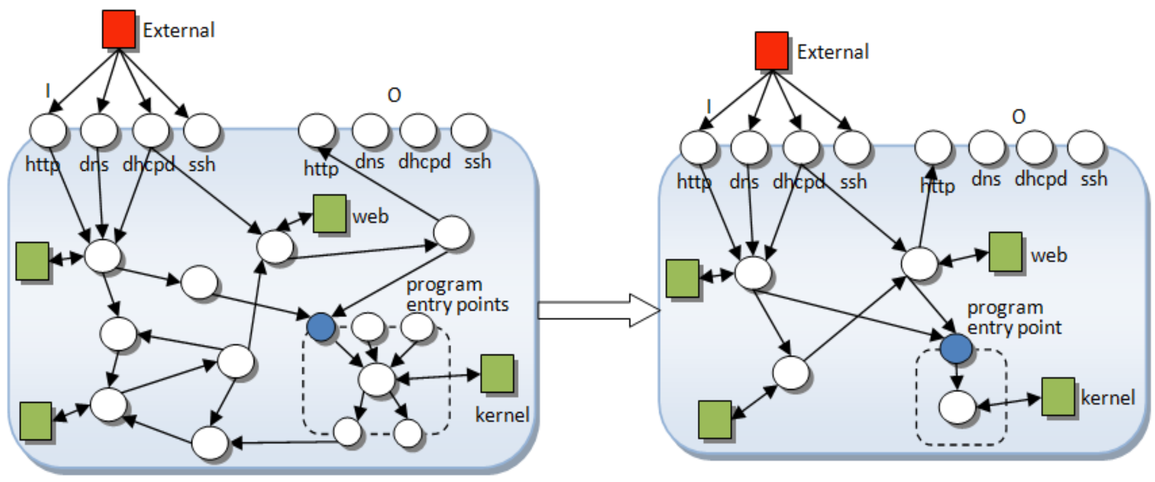
System

- The example organization network deploys a set of web applications across six networks, two server networks and four client networks.
- Deployment consist 3600 hosts of following configurations:
  - Web Server
  - Database Server
  - Web Client regular and protected
  - Privileged VM
- The system wide data flow graph for the above system generates
  - 9 million nodes
  - 19 million edges

Contributions

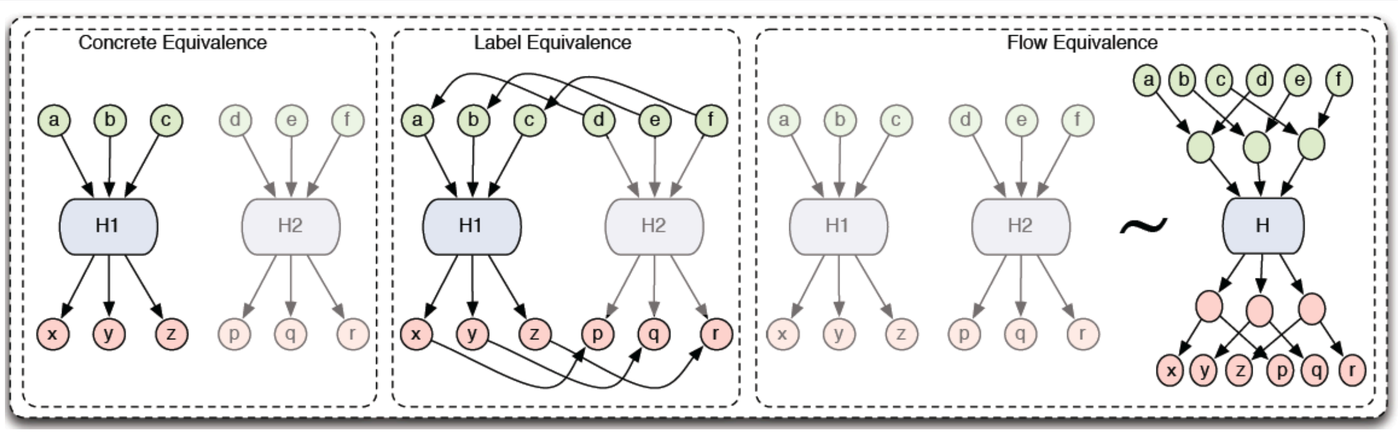
- **Key Observations:**
  - We find that, not all host internal data flow contributes in security relevant flows.
  - Also there is a significant redundancy among individual hosts in conventional systems because many hosts are usually deployed from the same image
- **Contributions:**
  - We generate the system wide data flow graph using the individual component policies and network policies.
  - We show how to build the information flow problem given the security goal.
  - We generate the host summaries by eliminating the irrelevant nodes that don't affect the information flow.
  - We merge the hosts that have similar information flow models, using three iterative approach.
  - We solve to compute mediations and network monitor placements using a greedy graph mincut approach..

Host Summary



- We identifying all the nodes mapped to any security level and the flows between them and the possible mediation nodes.
- We remove all other nodes that do not affect the flow between the above identified nodes to summarize the hosts.

Host Merge



- **Concrete Equivalence:** Hosts that have the dame data flow, same label mapping and same network connections.
- **Label Equivalence:** Hosts that have same data flow, same label mapping, but may have different network connections.
- **Flow Equivalence:** Hosts that have same data flow, might have different label mappings and network connections, but needs to model same information flow problem.

Results

- By summarizing we see about 80% and 60% reduction in nodes and edges in a host.
- Merging is independent of number of hosts and is dependent on the number of unique host configurations in a system.

Host count	Unique Mapping	Unique host Config	Concrete Equivalent	Label Equivalent	Flow Equivalent
3600	2	11	14	13	9
6000	4	15	26	18	13
9500	30	120	130	121	112

*Each system has 5 unique host data flows and 5 sub networks. The following table give magnitude of reduction obtained.

	Whole Network	Summarized	Merged Hosts
Nodes	9 million	1.5 million	3540
Edges	19 million	3.8 million	20819

Sponsored by Army Research Lab and Hewlett-Packard Innovation Research Program