Firewall Rules Discovery and Generation

Language Theory with Applications 2012

Matej Kollár

Faculty of Information Technology

December 12, 2012

Outline

Background

Policy discovery

Synthetic policy generation

Based on PhD thesis *Discovery, Generation and Analysis of Network Policy Configurations* by *Taghrid Samak*, 2010.

Outline

Background

Policy discovery

Synthetic policy generation

Firewall

Any barrier that is intended to thwart the spread of a unwanted (destructive, malicious, ...) agent.

First generation firewalls

- act by inspecting fields of packets,
  - source address
  - destination address
  - protocol
  - port
  - ...
- stateless (simple packet filters),
- packet can either pass or be dropped,
- described by “policy”.

We will not consider second and third generations.

General policy model

A general policy is modeled as a four-tuple structure

$$P = \langle \mathcal{C}, \mathcal{A}, \rho, \omega \rangle$$

- $\mathcal{C}$ n-dimensional domain specified by field values,
- $\mathcal{A}$ set of actions that can be taken when policy is applied,
- $\rho : C \to \mathcal{A}$, maps filter set conditions to actions ($C \subseteq 2^{\mathcal{C}}$),
- $\omega : C \times \mathcal{A} \to \mathbb{N}$ ordering function, maps rule set to priority level.

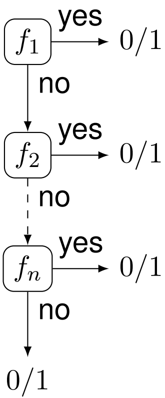
Decision list

$$L = [(f_1, v_1), (f_2, v_2), \ldots, (f_n, v_n)]$$

where

- f_i – boolean function over "packet",
- $v_i \in \{\text{pass}, \text{drop}\}$.

It is boolean function...
it is learnable.



Learnability—models of learning

Learning boolean functions.

Probably approximately correct (PAC) learning

Offline learning from examples. Aim is to find approximately correct hypothesis after seeing random sample of classified instances.

Mistake-bound (MB) learning

Adaptive learning. Sequence of trials. Learner enhances hypothesis. Learning terminates after specific number of mistakes. Better suited for interactive learning.

Outline

Background

Policy discovery

Synthetic policy generation

Policy discovery

Techniques:

- Exhaustive search
- Basic heuristics
  - Genetic algorithms
  - Region growing
  - Split-and-merge
- Hybrid heuristics

Region growing

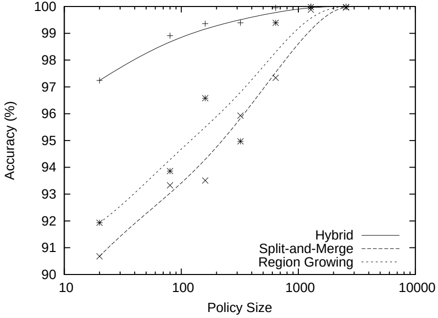
- “Deny all” assumed at the beginning.
- Sampling until positive match is found.
- Exponential search in every dimension.
- Binary search to pinpoint exact location.

Split-and-merge

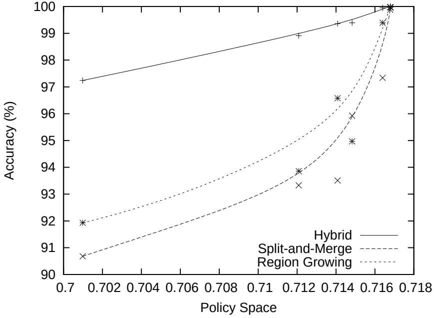
- Originally for image segmentation.
- Default "deny all".
- Split given space into n non-overlapping regions such that
  - $\bigcup_{i=1}^{n} R_i = R$,
  - R_i is a connected region,
  - R_i has rectangular shape (policy rule restriction),
  - $i \neq j \Rightarrow R_i \cap R_j = \emptyset$,
  - all "points" in R_i has same action,
  - no two regions can be joined so that previous conditions will be met.

Hybrid heuristic

- Split-and-merge with bound recursion depth.
- On Each resulting region run region growing to obtain exact boundaries.



Policy size vs. accuracy.



Policy space vs. accuracy.

Outline

Synthetic policy generation

Objective

Generate policies similar to those used in real world employing readily available data as much as possible and requiring minimal user intervention.

Applications

Packet classification algorithms, security devices testing, policy and configuration analysis algorithms.

Available data:

- grammar for policy description,
- example policy rules (e.g. provided by Cisco),
- vague description (using terms like policy size, rule complexity, . . .).

Policy grammar

Policy context free grammar defined by standard model:

$$G = (N, T, P, S)$$

- N – non-terminal symbols;
- T – terminal symbols;
- P – production rules;
- S – starting non-terminal.

Productions

$$\varrho : A \to \alpha; A \in N; \alpha \in (N \cup T)^+$$

Policy grammar

Policy context free grammar defined by standard model:

$$G = (N, T, P, S)$$

- N – non-terminal symbols;
- T – terminal symbols;
- P – production rules;
- S – starting non-terminal.

Productions

$$\varrho : A \to \alpha; A \in N; \alpha \in (N \cup T)^+$$

But this is not probabilistic at all...

Probabilistic grammar

To make grammar probabilistic, we need to add

$$p : P \rightarrow (0,1\rangle$$

such that

$$\forall A \in N : \sum_{A \rightarrow \alpha \in P} p(A \rightarrow \alpha) = 1$$

How do we obtain such probability function?

Probabilistic grammar

To make grammar probabilistic, we need to add

$$p : P \to (0{,}1\rangle$$

such that

$$\forall A \in N : \sum_{A \to \alpha \in P} p(A \to \alpha) = 1$$

How do we obtain such probability function?
Learn it from available data…

Estimation

Number of occurrences of rule in subtree $f(A \to \alpha, \tau)$.

Number of occurrences of nonterminal in subtree $f(A, \tau)$.

Given n parse trees $(\tau_1, \ldots, \tau_n)$, p can be approximated by

$$\hat{p}(A \to \alpha) = \frac{\sum_{i=1}^{n} f(A \to \alpha, \tau_i)}{\sum_{i=1}^{n} f(A, \tau_i)}$$

Better way

Let ω be subset of all possible parse trees such that every production rule appears in ω.

A positive weight $W(\tau)$ is assigned to each tree $\tau \in \omega$ such that $\sum_{\tau \in \omega} W(\tau) = 1$. The system production probabilities are then defined by:

$$p(A \to \alpha) = \frac{\sum_{\tau \in \omega} f(A \to \alpha, \tau) W(\tau)}{\sum_{\tau \in \omega} f(A, \tau) W(\tau)}$$

Even better way—Informed Mode

Observation

Position of a rule within the policy affects the rule structure.

Consider probability values within policy parts.

- Every rule single “part”,
- if there are two neighbouring parts similar up to threshold, merge them,
- if there is nothing to merge, terminate.

Conclusion

Make your policy as specific as possible.

Thank you for your attention.