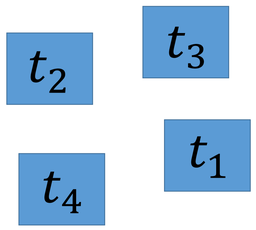
Statistical Reconstruction of Class Hierarchies in Binaries

Omer Katz Noam Rinetzky Eran Yahav

Problem: Existing techniques fail to reconstruct an actual class hierarchy resulting in gaps and partial hierarchies
Goal: Identify and output the most likely class hierarchy for a stripped binary
Solution: Reconstruct hierarchy based on behavioral similarity between types using tracelets

Input



set of types

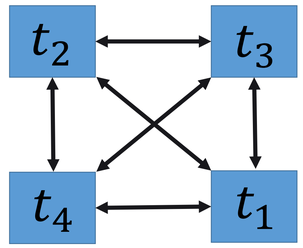
Measure similarity between types

- Use *Kullback-Leibler divergence*
- Measures similarity between SLMs
- Asymmetric measure

$$D_{KL}(A \parallel B) = \Sigma_{w \in W}\left(Pr_A(w) \cdot \ln\left(\frac{Pr_A(w)}{Pr_B(w)}\right)\right)$$

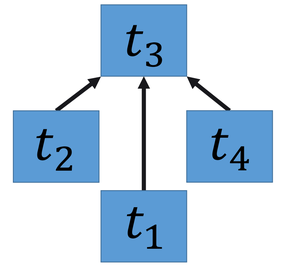
$$A = M_{t_1}, B = M_{t_2}, W = Tr(t_1)$$

Convert to a weighted graph



$$W(t_1 \rightarrow t_2) = D_{KL}(M_{t_1} \parallel M_{t_2})$$

Output



Minimum-weight directed spanning tree

Type representation

Collect a set of tracelets for each type

- Statically analyze the binary
- Extract usage sequences for types
- Sequence of operations applied to instances of that type
  - Filed accesses, function calls, etc…

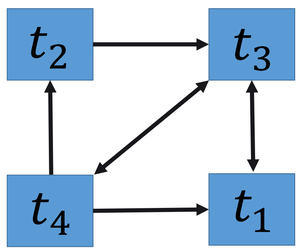
$$Tr(t) = \{W(16) \rightarrow R(0) \rightarrow C(0) \rightarrow R(8), \dots\}$$

Train model for each type

- Statistical language model (SLM), M_t, using Prediction by Partial Match algorithm
- Train using $Tr(t)$
  - Builds suffix tree from tracelets
- Given query tracelet q, model computes probability of tracelet, $\Pr_{M_t}(q)$

Simplify graph

- Using structural cues used by existing techniques
- Split to smaller subgraphs
  - Shared virtual table entries
- Eliminate impossible edges
  - Call to parent constructor
  - Virtual table size
- etc…



Evaluation Scenario:

Q: which types inherit from type t?

- Useful for Control Flow Integrity
  - Relevant to virtual function calls
  - Generate policy from hierarchy
  - Additional types → false positives
  - Missing types → false negatives

Benchmark	size (Kb)	num of types	Without SLMs Missing	Without SLMs Added	With SLMs Missing	With SLMs Added
Analyzer	419	24	0.21	6.79	0.25	1.38
CGridListCtrlEx	151	28	0.0	0.46	0.07	0.07
echoparams	58	4	0.0	2.25	0.0	0.0
gperf	84	10	0.0	3.8	0.0	0.5
libctemplate	1233	36	0.25	0.33	0.25	0.11
ShowTraf	137	25	0.04	0.4	0.04	0.08
Smoothing	453	31	0.19	7.9	0.23	1.1
td_unittest	101	2	0.0	1.0	0.0	0.5
tinyserver	46	4	0.0	2.25	0.0	0.25

Highlights:

- **echoparams**: reconstruct exact hierarchy
- **Smoothing**: from 7.9 false positives to 1.1
- **Analyzer**: from 6.79 false positives to 1.38

- Tradeoff between false negatives and false positives
  - Can use more than a single hierarchy to generate policy
  - Reduces missing types but increases added types

The research leading to the results presented in this paper is partially supported by the European Union's Seventh Framework Programme (FP7) under grant agreement no. 615688 (PRIME) and the Israel Science Foundation grant no.1319/16.