Estimating Types in Binaries using Predictive Modeling

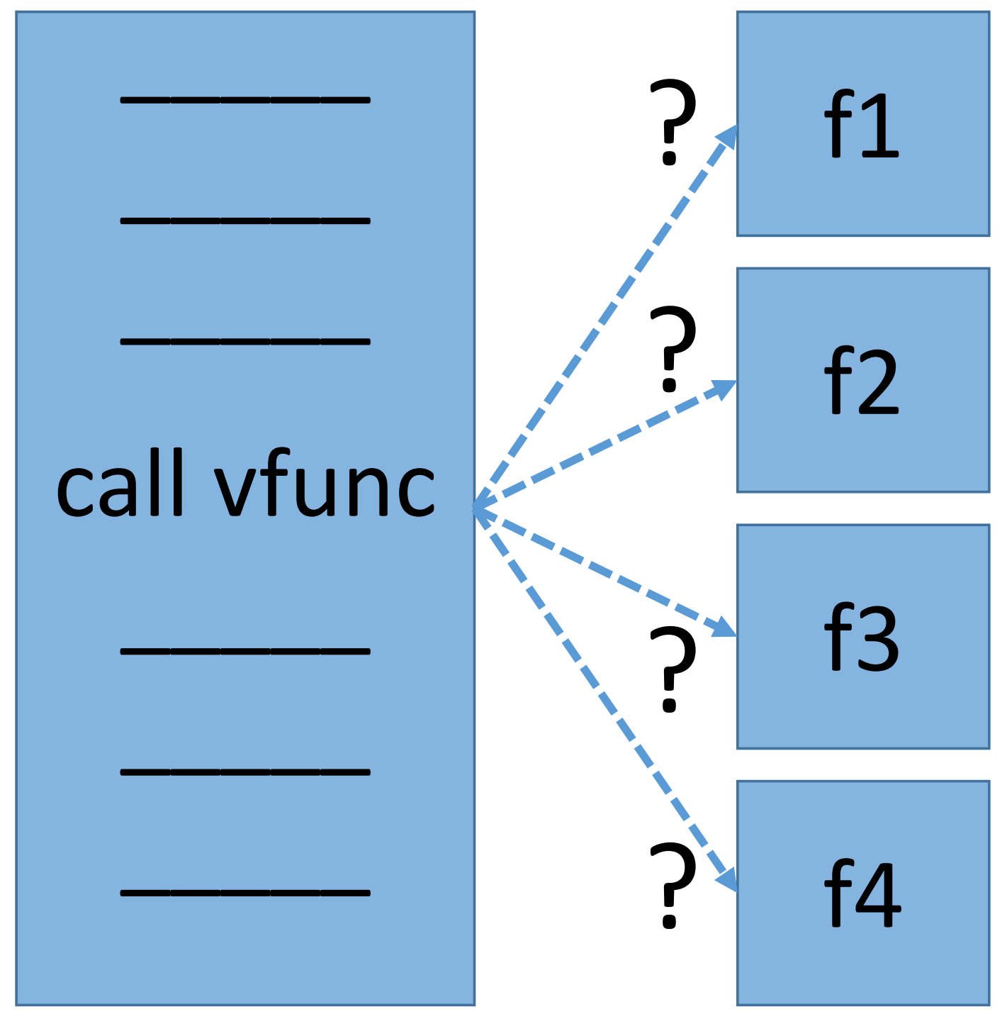
Omer Katz Ran El-Yaniv Eran Yahav

Problem: Calls to virtual functions break the control flow of programs and hinder reverse engineering (RE) efforts

Goal: Statically determine most likely targets of each virtual function call

Solution: Determine types of objects used in virtual function calls based on how they are used

Input



Many possible targets

2nd stage – Train a model for each type

Use objects for which a type can be statically determined (e.g. recently initialized)

N-gram based Statistical Language Models

- Computes sentence probability in a language $P_{English}(\text{I am giving a talk}) > P_{English}(\text{A talk giving I am})$
- Sentences of length n

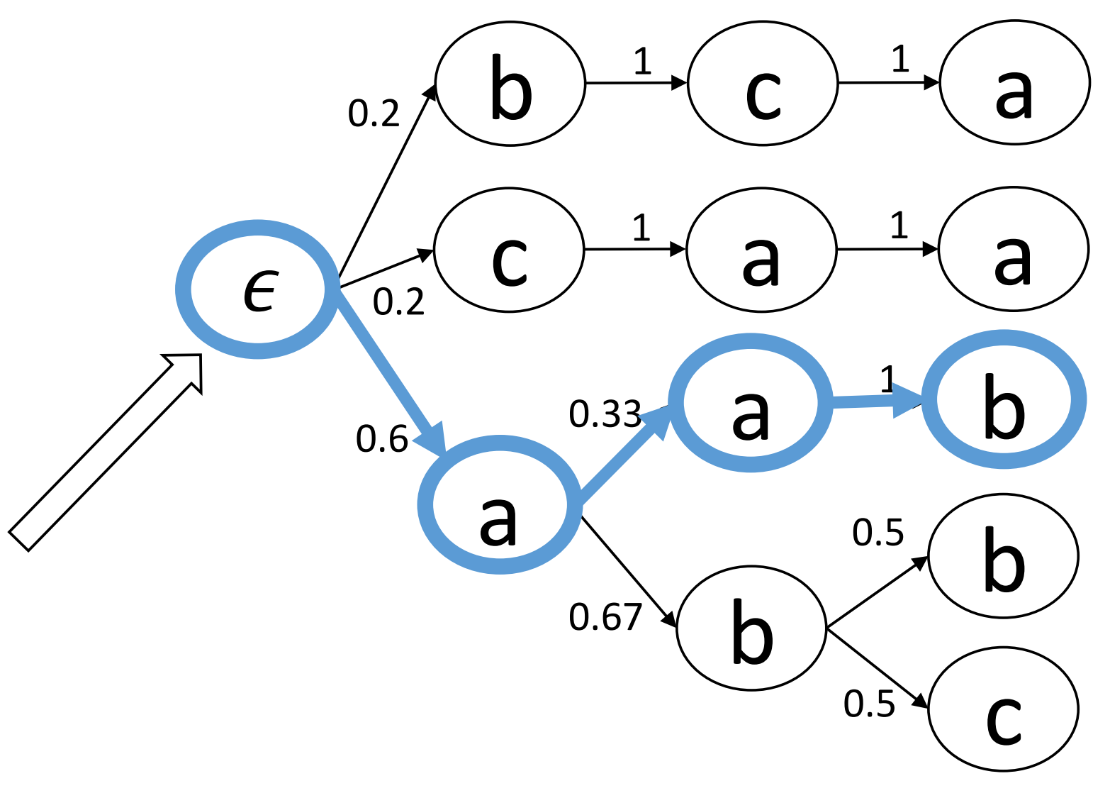
- Uses Prediction by Partial Match implementation
  - Originally a compression algorithm

Train model by parsing to n-grams and building probability tree

Query model by traversing path to sentence

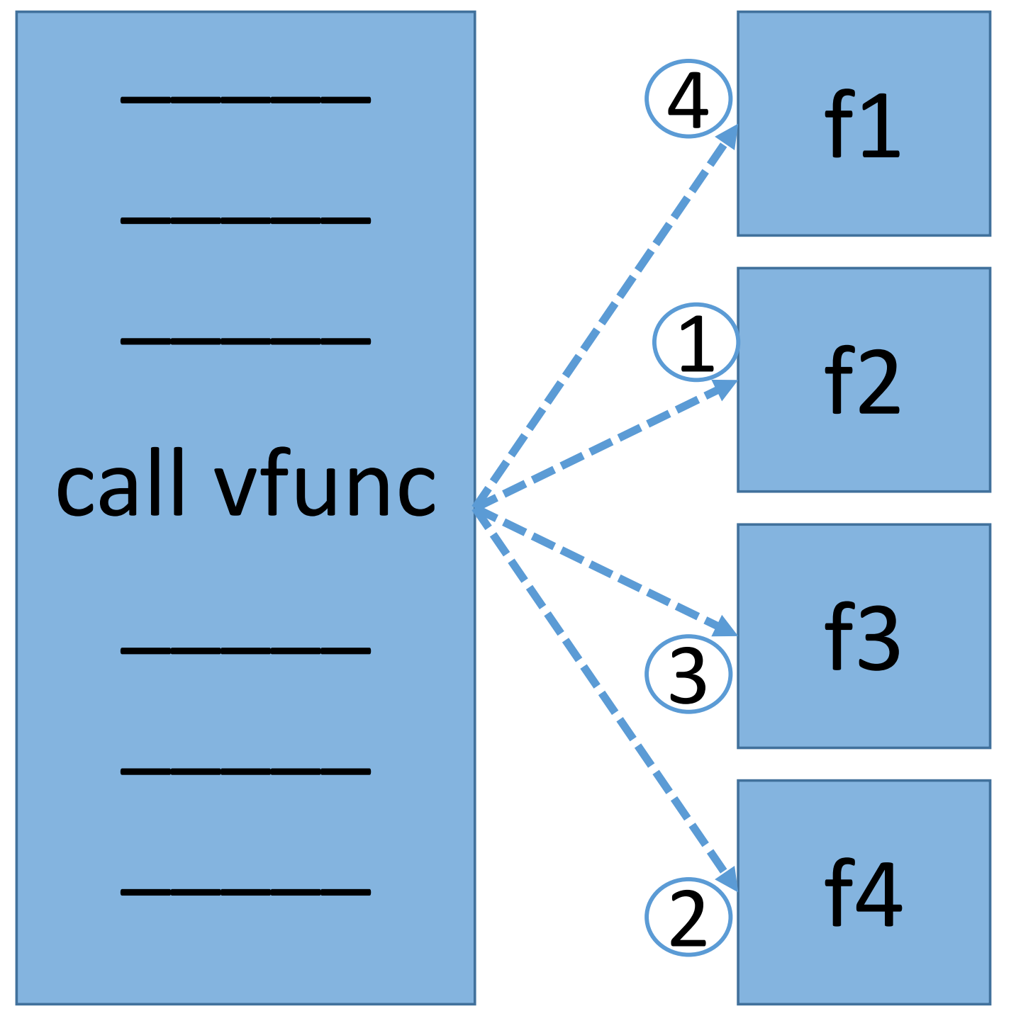
abcaabb

3-grams:
abc
bca
caa
aab
abb



$P(\text{aab}) = 0.2$

Output



Focus on most likely targets

1st stage – Binary Analysis and Tracelet Extraction

eax points to object

```
mov    [eax+10h],1
mov    ebx,[eax]
mov    edx,[ebx]
call   edx
mov    ebx,[eax+08h]
```

$W(16) - Write\ Field\ at\ offset\ 16$

$R(0) - Read\ Field\ at\ offset\ 0$

$C(0) - Call\ 1^{st}\ Virtual\ Function$

$R(8) - Read\ Field\ at\ offset\ 8$

$W(16) \rightarrow R(0) \rightarrow C(0) \rightarrow R(8)$

Tracelets:

- $W(16) \rightarrow R(0) \rightarrow C(0) \rightarrow R(8) \rightarrow W(16)$
- $R(0) \rightarrow C(0) \rightarrow R(8) \rightarrow W(16) \rightarrow R(0)$
- $C(0) \rightarrow R(8) \rightarrow W(16) \rightarrow R(0) \rightarrow C(0)$
- $R(8) \rightarrow W(16) \rightarrow R(0) \rightarrow C(0) \rightarrow R(8)$

3rd stage – Compute ranking of types for objects

Use objects for which a type can't be statically determined

- Model can compute probability of tracelet
- For object o and type t:

$$P_{M(t)}(o) = \Pi_{tr \in TR(o)} P_{M(t)}(tr)$$

$TR(o)$ ≔ set of tracelets of o

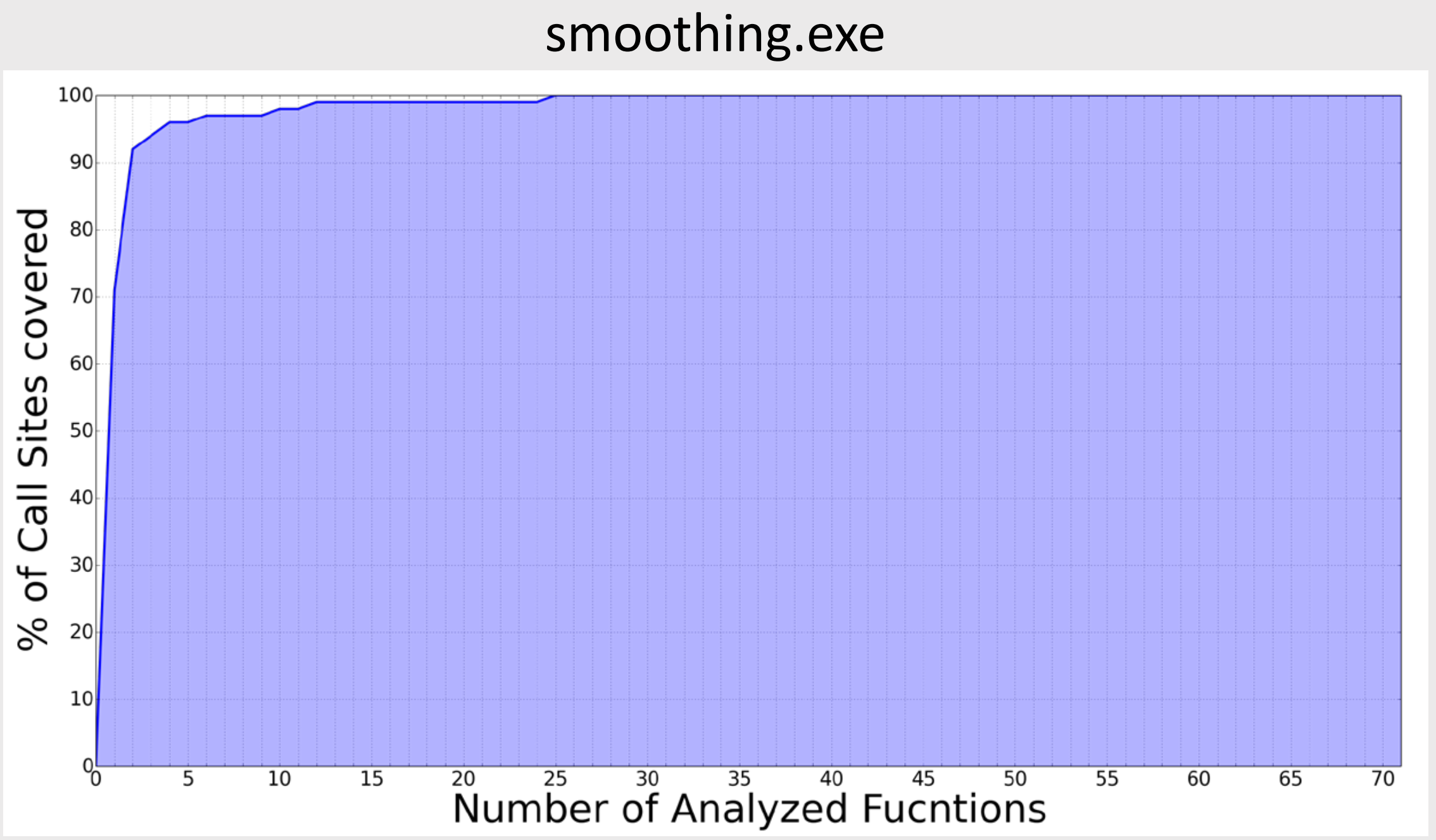
- Rank types for object o according to $P_{M(t)}(o)$

Evaluation:

- Evaluated over 20 benchmarks
- Compared to ground truth from manual RE
- Objective: rank expected target highest
- Across all benchmarks, for over 80% of calls to virtual functions, expected target ranked in top 3

Result of *Smoothing.exe:*

- ***X*** axis – maximum rank
- ***Y*** axis – percentage of calls to virtual functions for which the expected target was ranked below the maximum rank



The research leading to these results has received funding from the European Union's - Seventh Framework Programme (FP7) under grant agreement n° 615688 – ERC- COG-PRIME.