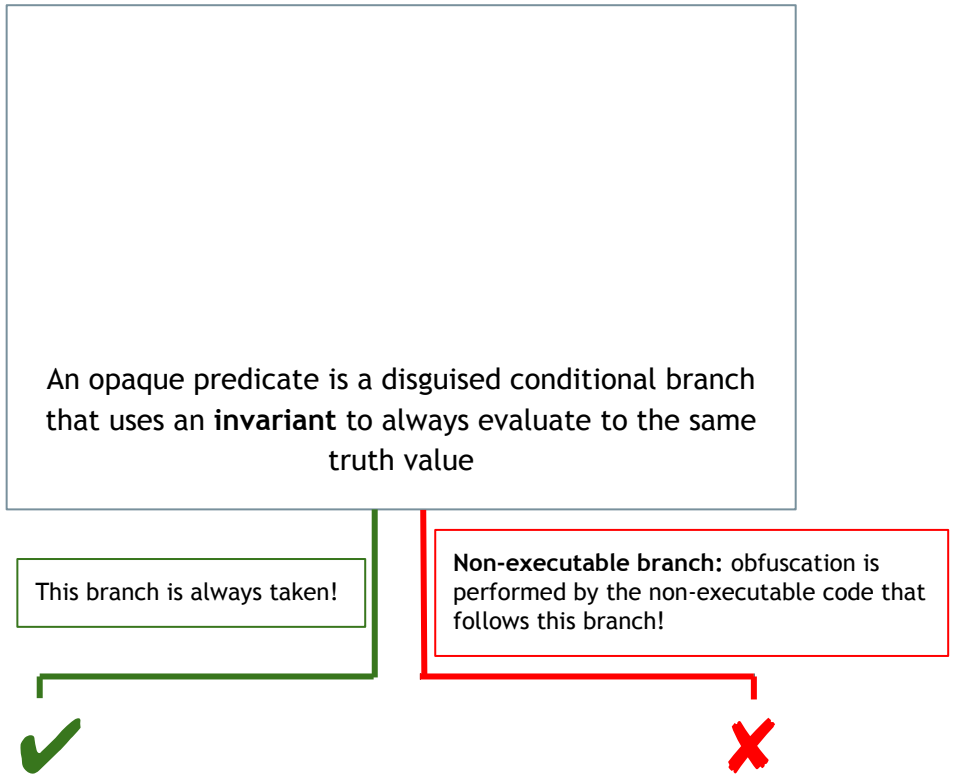
Zero Footprint Opaque Predicates:
Synthesizing Opaque Predicates From Naturally Occurring Invariants

Yu-Jye Tung, Ian G. Harris

What is Opaque Predicates?



OLLVM

algebraic opaque predicate

0x80484ed

```
                    ...
mov    edx, ds:x
mov    esi, ds:y
mov    edi, edx
sub    edi, 1
imul   edx, edi
and    edx, 1
cmp    edx, 0
setz   bl
cmp    esi, 0Ah
setl   bh
or     bl, bh
test   bl, 1
mov    [ebp+var_10], eax
mov    [ebp+var_14], ecx
jnz    loc_80484F8
```

This branch is always taken!

Non-executable branch: obfuscation is performed by the non-executable code that follows this branch!

OLLVM

algebraic opaque predicate

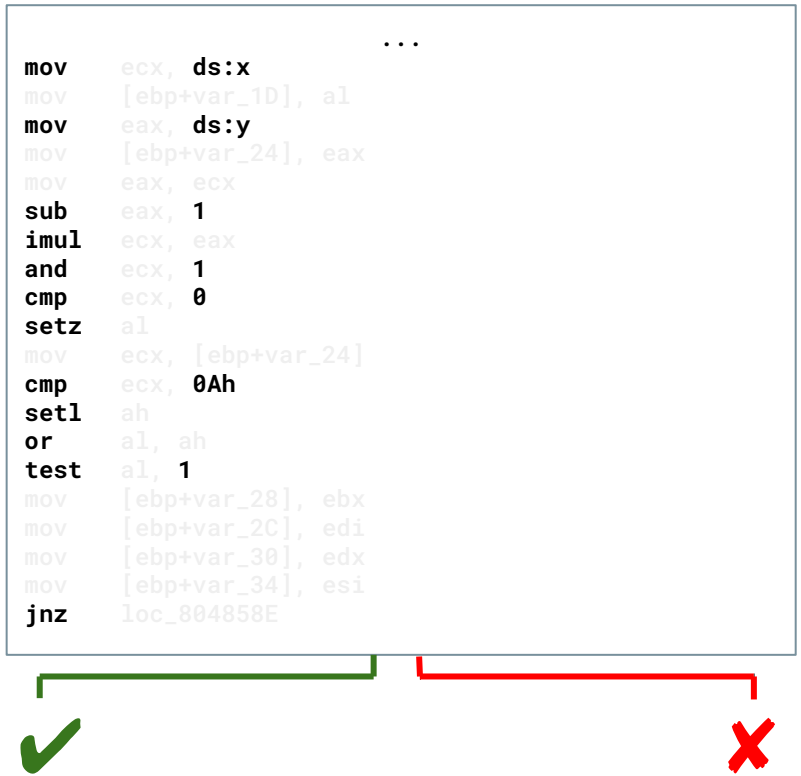
0x8048583

```
                    ...
mov    ecx, ds:x
mov    [ebp+var_1D], al
mov    eax, ds:y
mov    [ebp+var_24], eax
mov    eax, ecx
sub    eax, 1
imul   ecx, eax
and    ecx, 1
cmp    ecx, 0
setz   al
mov    ecx, [ebp+var_24]
cmp    ecx, 0Ah
setl   ah
or     al, ah
test   al, 1
mov    [ebp+var_28], ebx
mov    [ebp+var_2C], edi
mov    [ebp+var_30], edx
mov    [ebp+var_34], esi
jnz    loc_804858E
```

Heuristic Attack

OLLVM

algebraic opaque predicate



0x80484ed ∩ 0x8048583

Shared features can be used to detect other algebraic opaque predicates in the same binary!

Tigress

array-based opaque predicate

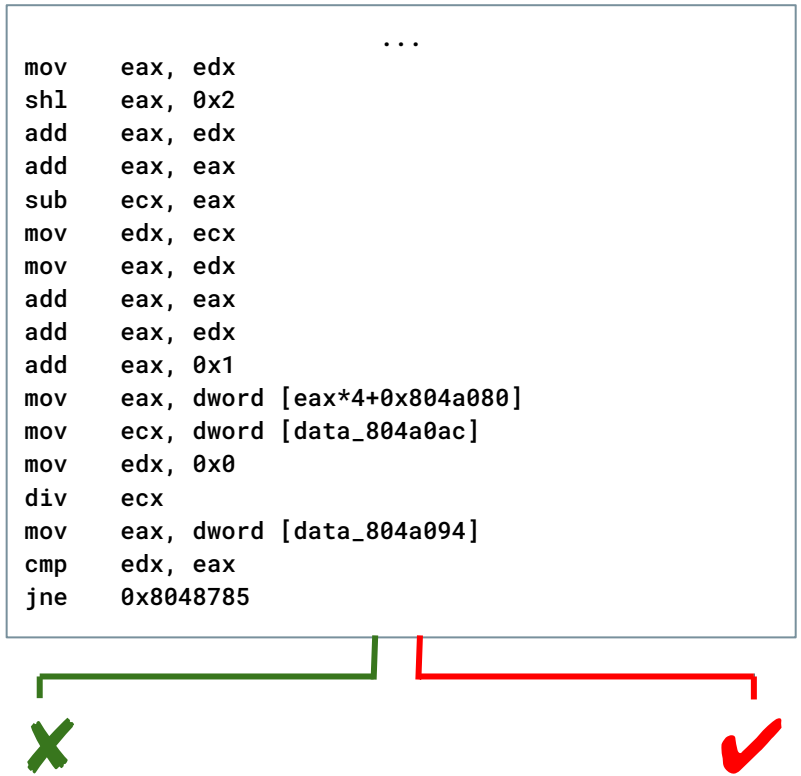
```
                    ...
mov    eax, edx
shl    eax, 0x2
add    eax, edx
add    eax, eax
sub    ecx, eax
mov    edx, ecx
mov    eax, edx
add    eax, eax
add    eax, edx
mov    eax, dword [eax*4+0x804a080]
mov    ecx, dword [data_804a094]
mov    edx, 0x0
div    ecx
mov    eax, dword [data_804a088]
cmp    edx, eax
jne    0x804880a
```

0x80487fc

Tigress

array-based opaque predicate

0x8048752



Heuristic Attack

Tigress

array-based opaque predicate

```
                                ...
mov     eax, edx
shl     eax, 0x2
add     eax, edx
add     eax, eax
sub     ecx, eax
mov     edx, ecx
mov     eax, edx
add     eax, eax
add     eax, edx
add     eax, 0x1
mov     eax, dword [eax*4+0x804a080]
mov     ecx, dword [data_804a0ac]
mov     edx, 0x0
div     ecx
mov     eax, dword [data_804a094]
cmp     edx, eax
jne     0x8048785
```

0x80487fc ∩ 0x8048752

Shared features can be used to detect other array-based opaque predicates in the same binary!

Key Idea

Syntactically and *Semantically* Resemble Real Predicates

Key Idea

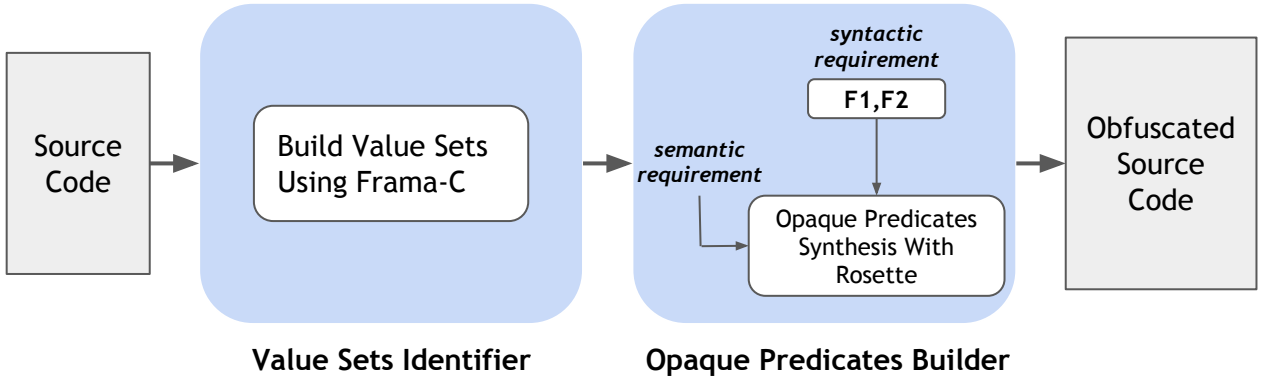
Syntactically and *Semantically* Resemble Real Predicates

Prevent Heuristic Attacks

Maintain Resilience Against Automated Attacks

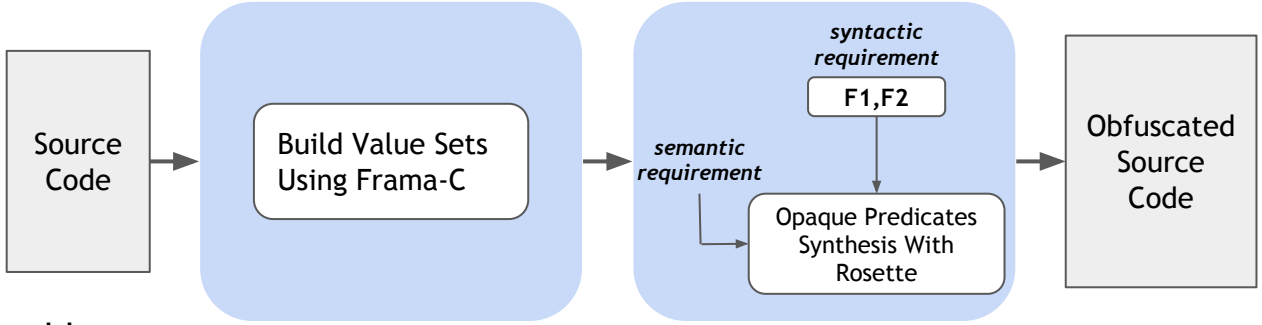
Zero Footprint Opaque Predicates

Our Approach!



Zero Footprint Opaque Predicates

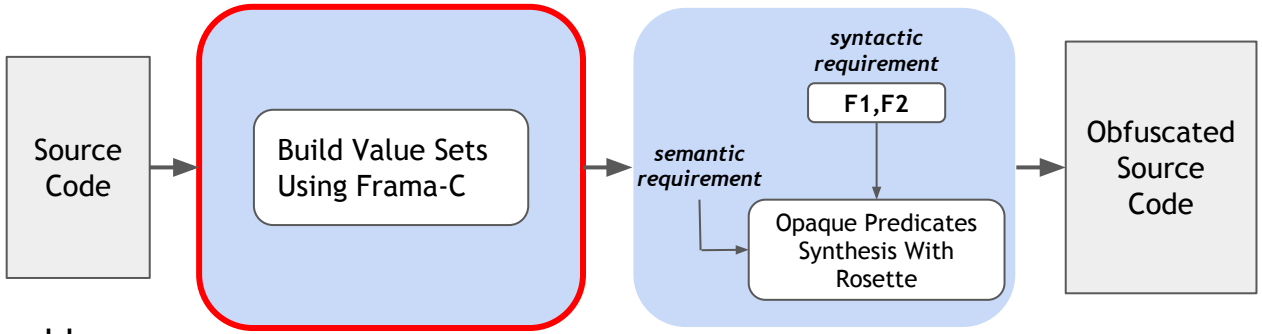
Our Approach!



How To Resemble Real Predicates?

Zero Footprint Opaque Predicates

Our Approach!



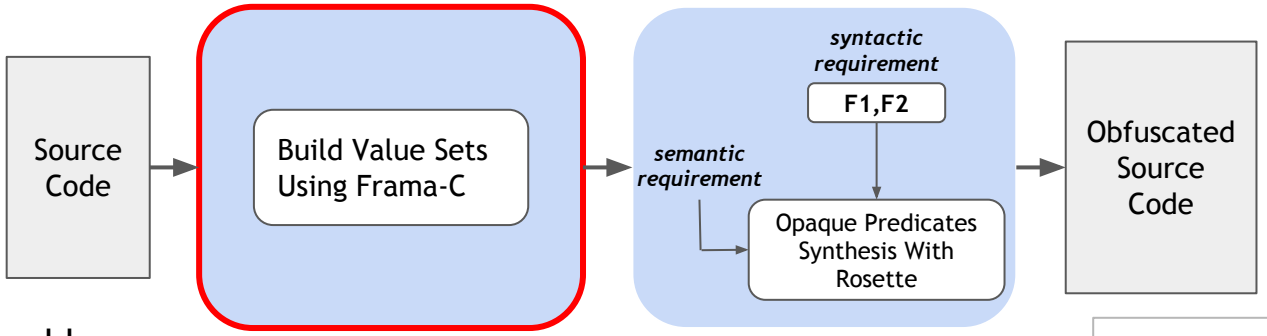
How To Resemble Real Predicates?

Syntactically

- Using naturally occurring invariants as our opaque predicate's invariant

Zero Footprint Opaque Predicates

Our Approach!



How To Resemble Real Predicates?

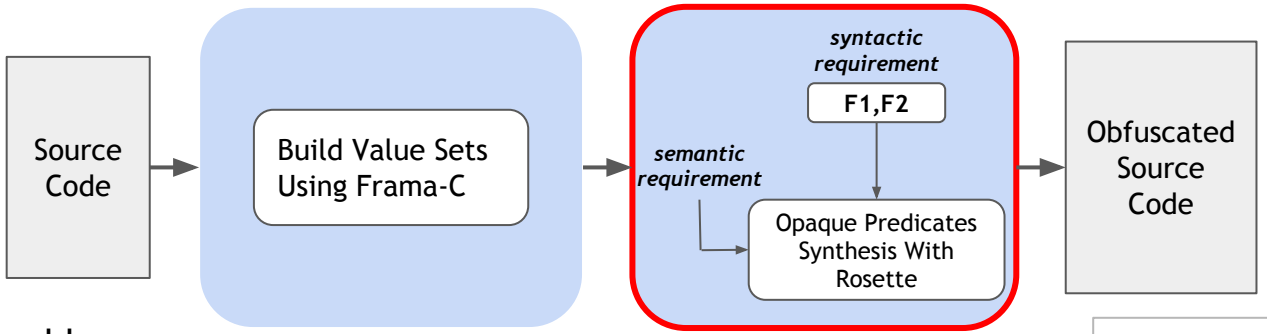
Syntactically

- Using naturally occurring invariants as our opaque predicate's invariant

Previous works' opaque predicates use synthetic invariants which are prone to heuristic attacks!!

Zero Footprint Opaque Predicates

Our Approach!



How To Resemble Real Predicates?

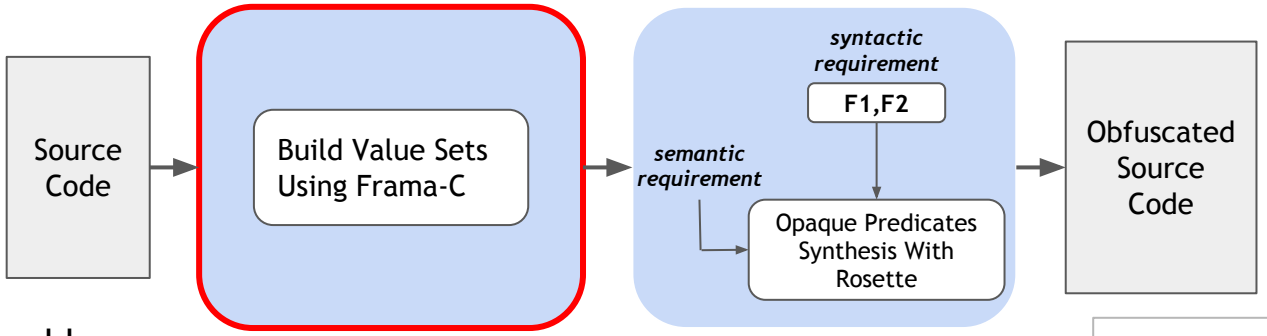
Syntactically

- Using naturally occurring invariants as our opaque predicate's invariant
- Using synthesis to apply syntactic biases of real predicates to the search space

Previous works' opaque predicates use synthetic invariants which are prone to heuristic attacks!!

Zero Footprint Opaque Predicates

Our Approach!



How To Resemble Real Predicates?

Value Sets Identifier

Opaque Predicates Builder

Previous works' opaque predicates use synthetic invariants which are prone to heuristic attacks!!

Syntactically

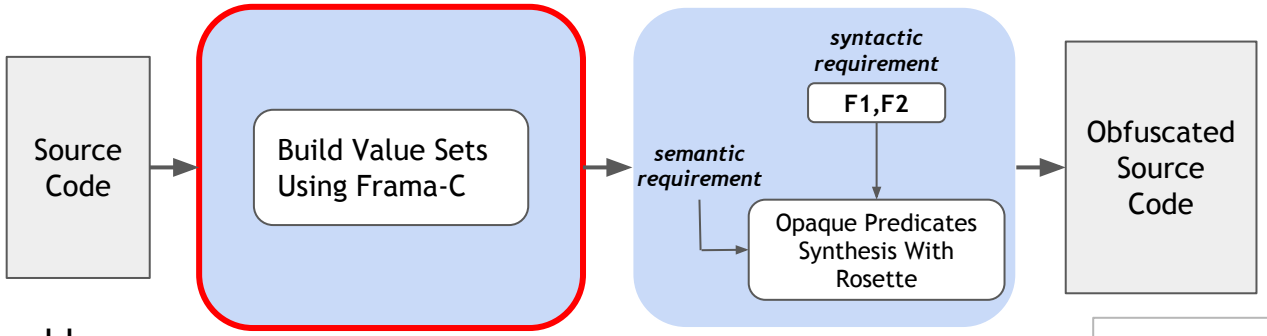
- Using naturally occurring invariants as our opaque predicate's invariant
- Using synthesis to apply syntactic biases of real predicates to the search space

Semantically

- Using value sets as invariants to exhibit behaviors of real predicates

Zero Footprint Opaque Predicates

Our Approach!



How To Resemble Real Predicates?

Previous works' opaque predicates use synthetic invariants which are prone to heuristic attacks!!

Syntactically

- Using naturally occurring invariants as our opaque predicate's invariant
- Using synthesis to apply syntactic biases of real predicates to the search space

What are they?

Semantically

- Using value sets as invariants to exhibit behaviors of real predicates

How to use value sets as invariants?

Syntactic Requirement

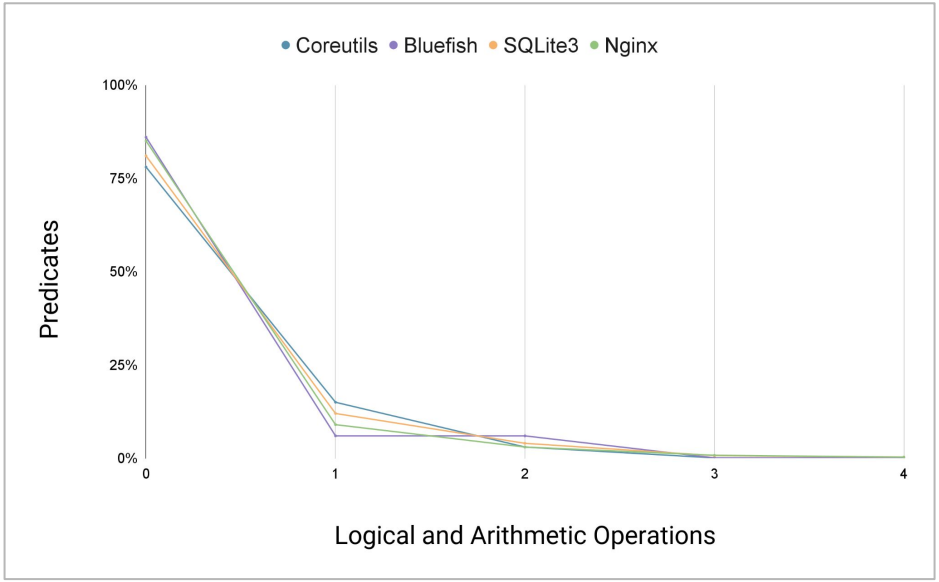
Syntactic Biases of Real Predicates

Zero or Few Logical and Arithmetic Operations

Coreutils (101254 predicates)
SQLite3 (27339 predicates)
Nginx (12660 predicates)
Bluefish (5942 predicates)

To reflect how reverse engineers perceive predicates:

- Analysis performed on binaries instead of source
- Basic block-level analysis



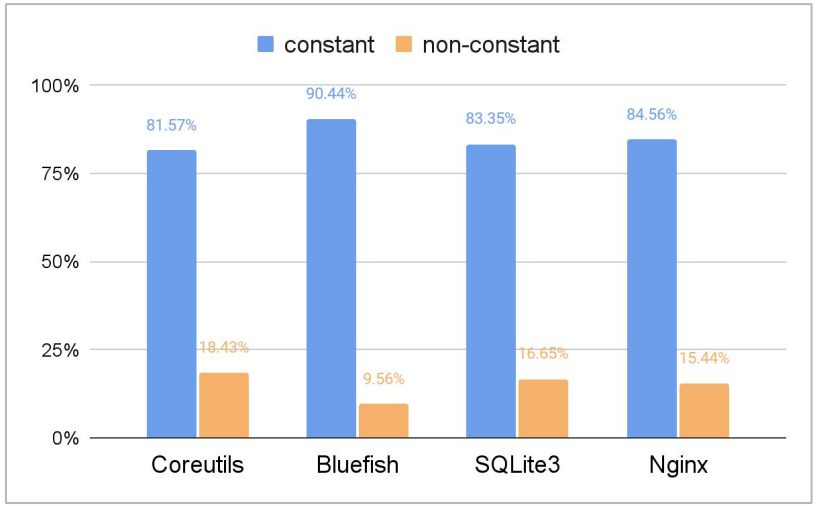
Ref: D. Votipka, S. Rabin, K. Micinski, J. Foster, and M. Mazurek. “An Observational Investigation of Reverse Engineers Processes” in USENIX Security '20

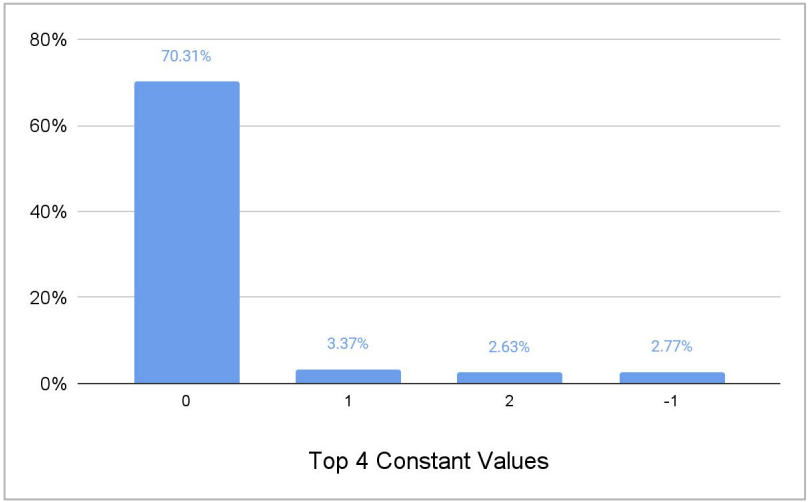
Comparison with a Constant of Value 0, -1, 1, or 2

Coreutils (101254 predicates)
SQLite3 (27339 predicates)
Nginx (12660 predicates)
Bluefish (5942 predicates)

To reflect how reverse engineers perceive predicates:

- Analysis performed on binaries instead of source
- Basic block-level analysis





Ref: D. Votipka, S. Rabin, K. Micinski, J. Foster, and M. Mazurek. “An Observational Investigation of Reverse Engineers Processes” in USENIX Security '20

Semantic Requirement

Invariant Property Imposed by Value Sets

Abstract Interpretation (Frama-C)

Inferring correct value sets by reasoning in a less precise abstract domain

```
...
for (i=2; i<=n; i++) {
  c = a+b;
  a = b;
  b = c;
}
...
```

Inferred Value Sets → *Synthesized Opaque Predicates*

```
c = {1,2,3,4,5,6,7,8}  ⟶  c>=0
a = {1,2,3,4,5}        ⟶  a<0
b = {1,2,3,4,5,6,7,8}  ⟶  b!=-1
```

Guarantees Zero False Negative!

Abstract Interpretation (Frama-C)

Inferring correct value sets by reasoning in a less precise abstract domain

```
...
for (i=2; i<=n; i++) {
  c = a+b;
  a = b;
  b = c;
}
...
```

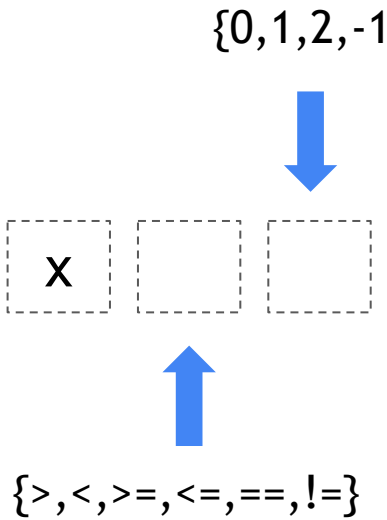
Still Correct!

Actual Value Sets		*Synthesized Opaque Predicates*
c = {1,2,3,4,5,6,7,8}	➡	c>=0
a = {1,2,3,4,5}	➡	a<0
b = {1,2,3,4,5,6,7,8}	➡	b!=-1

Guarantees Zero False Negative!

Program Synthesis (Rosette)

Generating program from a specification defined *wrt* syntactic and semantic requirements



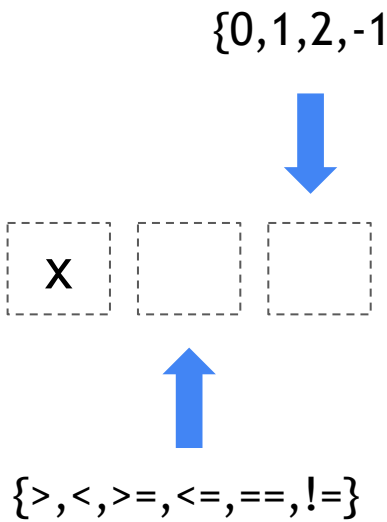
Defines the space of programs to search in!

Ref: S. Gulwani. “Dimensions in Program Synthesis” in PPDP’10

Program Synthesis (Rosette)

Generating program from a specification defined *wrt* syntactic and semantic requirements

```
(define sketch
    (expr 'x [choose > < >= <= eq? neq?] [choose 0 1 2 -1])
```



Defines the space of programs to search in!

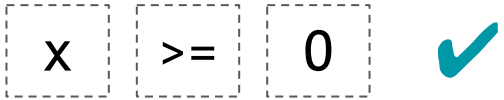
Ref: S. Gulwani. "Dimensions in Program Synthesis" in PPDP'10

Program Synthesis (Rosette)

Generating program from a specification defined *wrt* syntactic and semantic requirements

```
(for ([i value-set])
  (assert (interpret sketch i)))
```

{0,1,2,3,5,6,8,9}



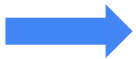
Ref: S. Gulwani. “Dimensions in Program Synthesis” in PPDP’10

Program Synthesis (Rosette)

Generating program from a specification defined *wrt* syntactic and semantic requirements

```
(for ([i value-set])
  (assert (interpret sketch i)))
```

{0,1,2,3,5,6,8,9}



{T,T,T,T,T,T,T,T}

Behavior of the program to search for!

x >= 0 ✔

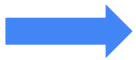
Ref: S. Gulwani. “Dimensions in Program Synthesis” in PPDP’10

Program Synthesis (Rosette)

Generating program from a specification defined *wrt* syntactic and semantic requirements

```
(for ([i value-set])
  (assert (not (interpret sketch i))))
```

{0,1,2,3,5,6,8,9}



{F,F,F,F,F,F,F,F}

Behavior of the program to search for!

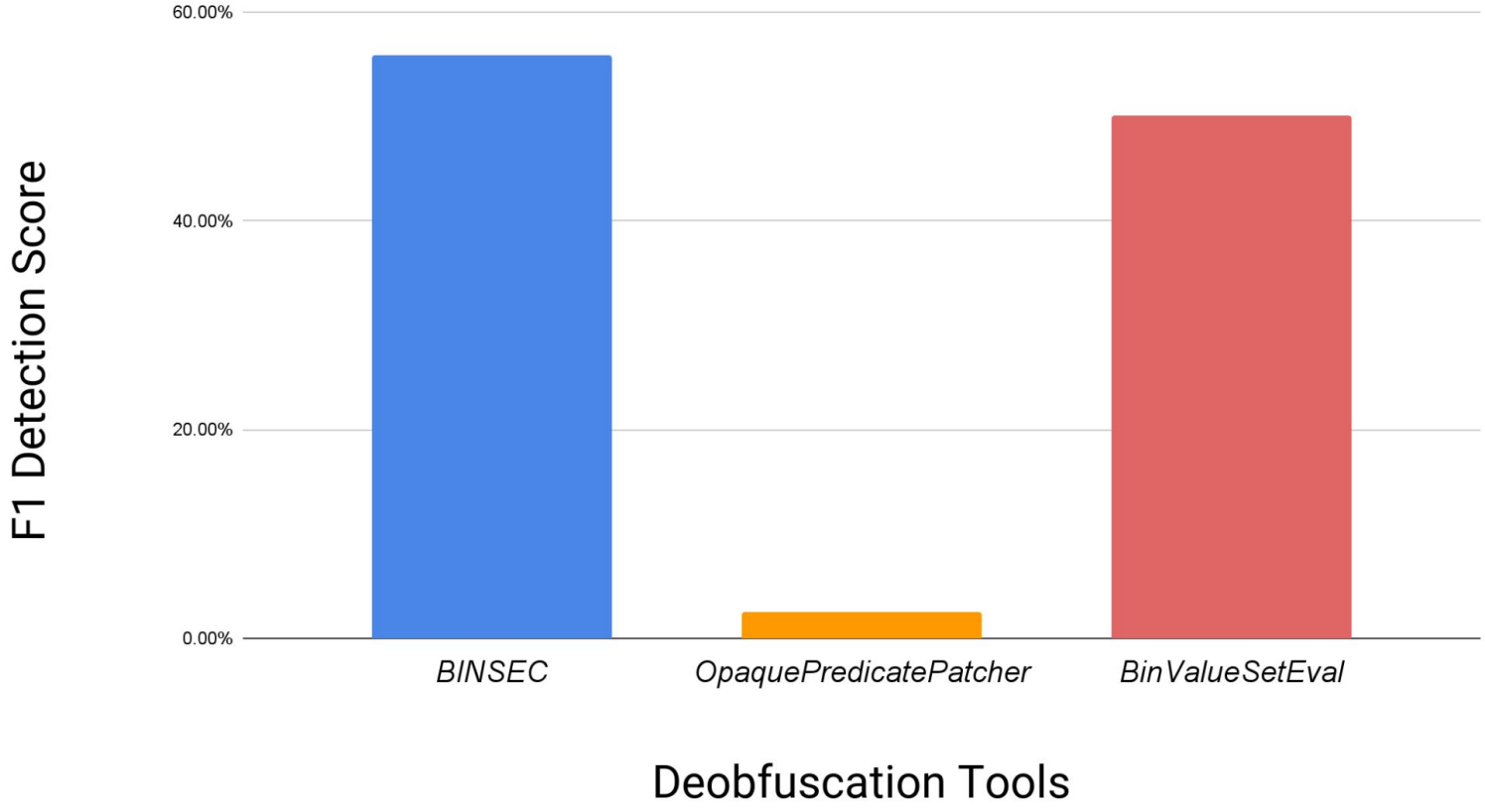
x < 0 ✔

Ref: S. Gulwani. “Dimensions in Program Synthesis” in PPDP’10

Frama-C Open Source Case Studies

2048
Solitaire
Tweetnacl-usable
Kilo
Bench-moerman2018

Total of 488 Opaque Predicates



Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

Compiler can remove the opaque predicates we insert

Features of real predicates based on predetermined set of real-world programs

No evaluation on manual analysis

Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

- *Synthesize more than one opaque predicate from one value set!*

Compiler can remove the opaque predicates we insert

Features of real predicates based on predetermined set of real-world programs

No evaluation on manual analysis

Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

- *Synthesize more than one opaque predicate from one value set!*

Compiler can remove the opaque predicates we insert

- *Out of 490 opaque predicates inserted at source-level, only two are removed!*

Features of real predicates based on predetermined set of real-world programs

No evaluation on manual analysis

Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

- *Synthesize more than one opaque predicate from one value set!*

Compiler can remove the opaque predicates we insert

- *Out of 490 opaque predicates inserted at source-level, only two are removed!*

Features of real predicates based on predetermined set of real-world programs

- *Identify features from the program being obfuscated*

No evaluation on manual analysis

Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

- *Synthesize more than one opaque predicate from one value set!*

Compiler can remove the opaque predicates we insert

- *Out of 490 opaque predicates inserted at source-level, only two are removed!*

Features of real predicates based on predetermined set of real-world programs

- *Identify features from the program being obfuscated*

No evaluation on manual analysis

- *Employ human studies*

Limitation/Future Work

Number of opaque predicates synthesized depends on number of value sets inferred

- *Synthesize more than one opaque predicate from one value set!*

Compiler can remove the opaque predicates we insert

- *Out of 490 opaque predicates inserted at source-level, only two are removed!*

Features of real predicates based on predetermined set of real-world programs

- *Identify features from the program being obfuscated*

No evaluation on manual analysis

- *Employ human studies*

Stay Tuned!

Questions?