DifFuzz: Differential Fuzzing for Side-Channel Analysis

Shirin Nilizadeh[1], Yannic Noller[2], Corina S. Păsăreanu[3]

[1] University of Texas at Arlington, USA
[2] Humboldt-Universität zu Berlin, Germany
[3] Carnegie Mellon University Silicon Valley, NASA Ames Research Center, Moffett Field, USA

Problem: Side-Channel Vulnerabilities

- ***secure*** if the secret data can not be inferred by an attacker through their observations of the system (aka ***non-interference***)
- ***observables***: execution time, memory consumption, response size, …
- can be solved by self-composition

  $\forall pub, sec_1, sec_2 : c(P[pub, sec_1]) = c(P[pub, sec_2])$
- ϵ-bounded non-interference

  $\forall pub, sec_1, sec_2 : |c(P[pub, sec_1]) - c(P[pub, sec_2])| < \epsilon$

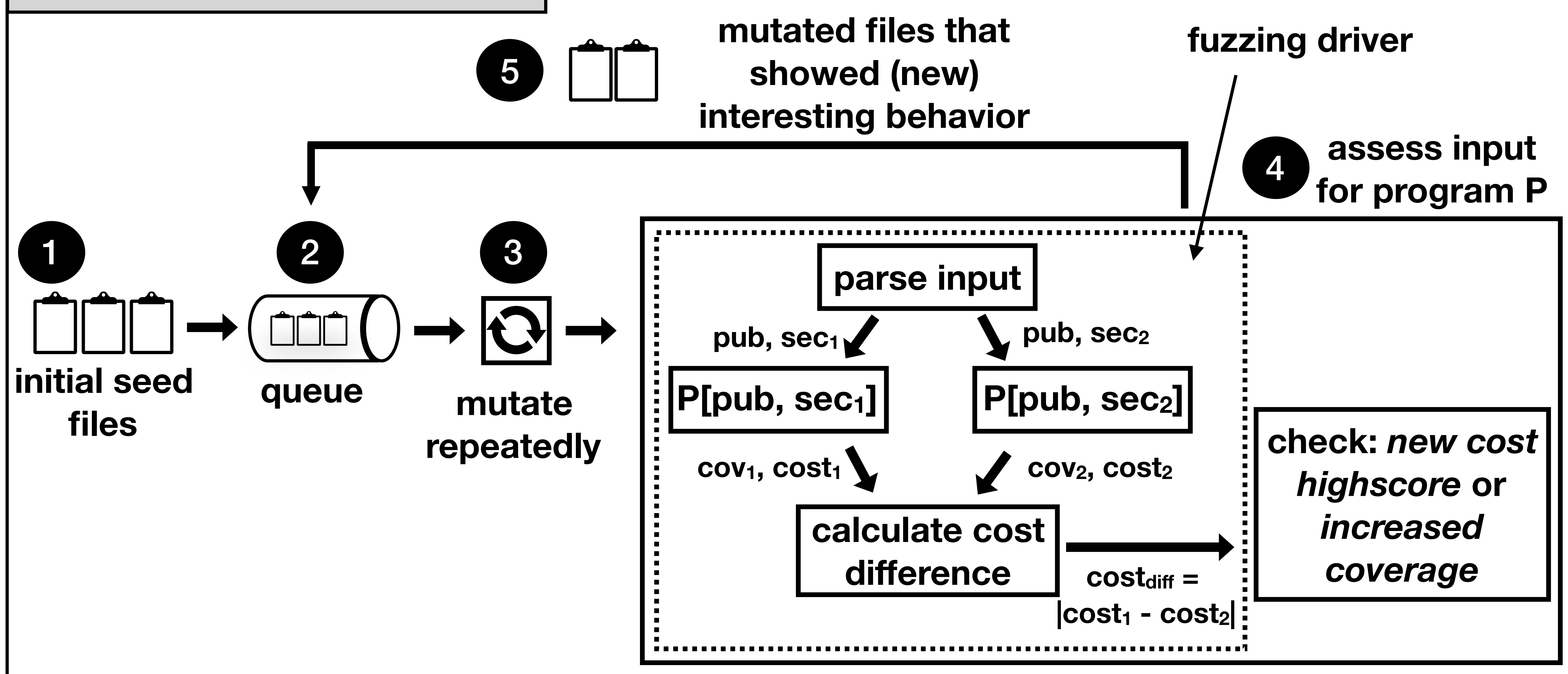
Example (Timing Side-Channel)

```
boolean pwcheck (byte[] pub, byte[] sec) {
    if (pub.length != sec.length) {
        return false;
    }
    for (int i = 0; i < pub.length; i++) {
        if (pub[i] != sec[i]) {
            return false;
        }
    }
    return true;
}
```

leaks information about length

leaks information about the actual bytes

Solution Concept


5
mutated files that showed (new) interesting behavior
fuzzing driver
4
assess input for program P
1
2
3
initial seed files
queue
mutate repeatedly
parse input
pub, sec₁
pub, sec₂
P[pub, sec₁]
P[pub, sec₂]
cov₁, cost₁
cov₂, cost₂
calculate cost difference
costdiff = |cost₁ - cost₂|
check: new cost highscore or increased coverage

maximize: $\max_{pub,sec_1,sec_2} \delta = |c(P[pub, sec_1]) - c(P[pub, sec_2])|$

Example Results

Initial Input: cost$_{\text{Diff}}$ = 0 (measured in #instructions)

```
secret₁ = [72, 101, 108, 108, 111,  32,  67]
secret₂ = [97, 114, 110, 101, 103, 105, 101]
public  = [32,  77, 101, 108, 108, 111, 110]
```

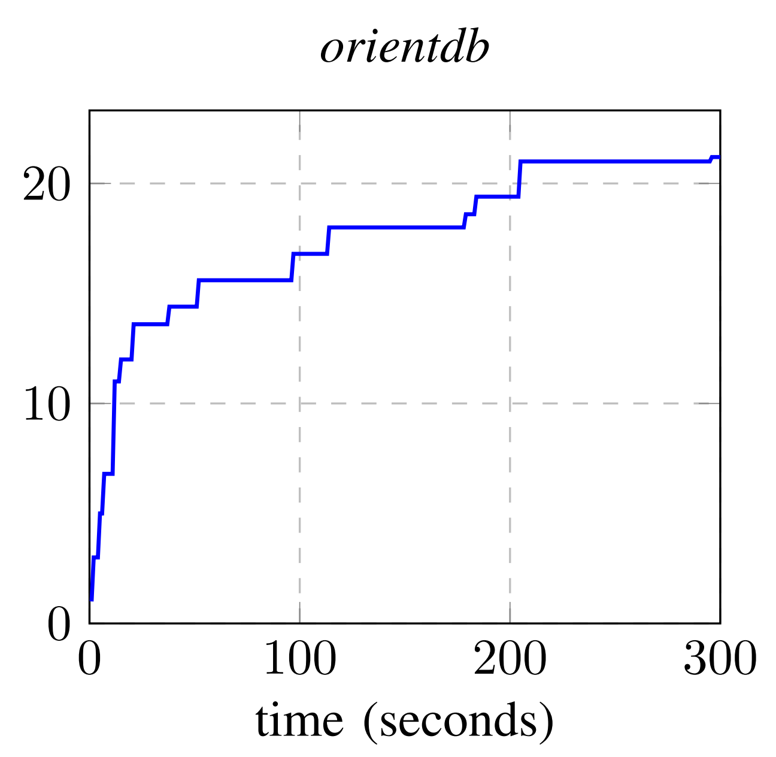
cost$_{\text{Diff}}$ > 0 after ~ 5 sec

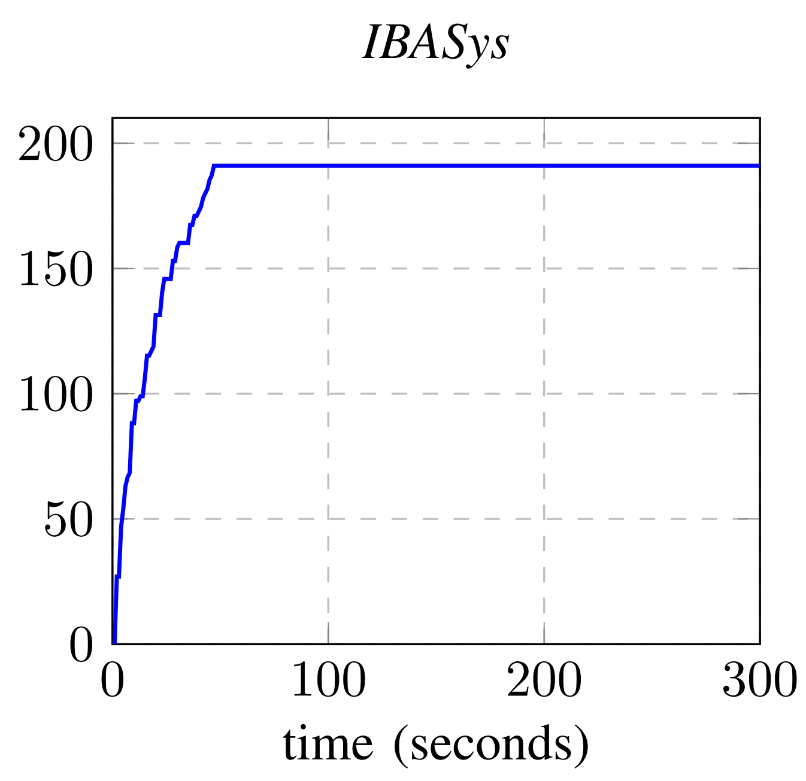
Input with highscore cost$_{\text{Diff}}$ = 47 after ~ 69 sec (maximum length = 16 bytes):

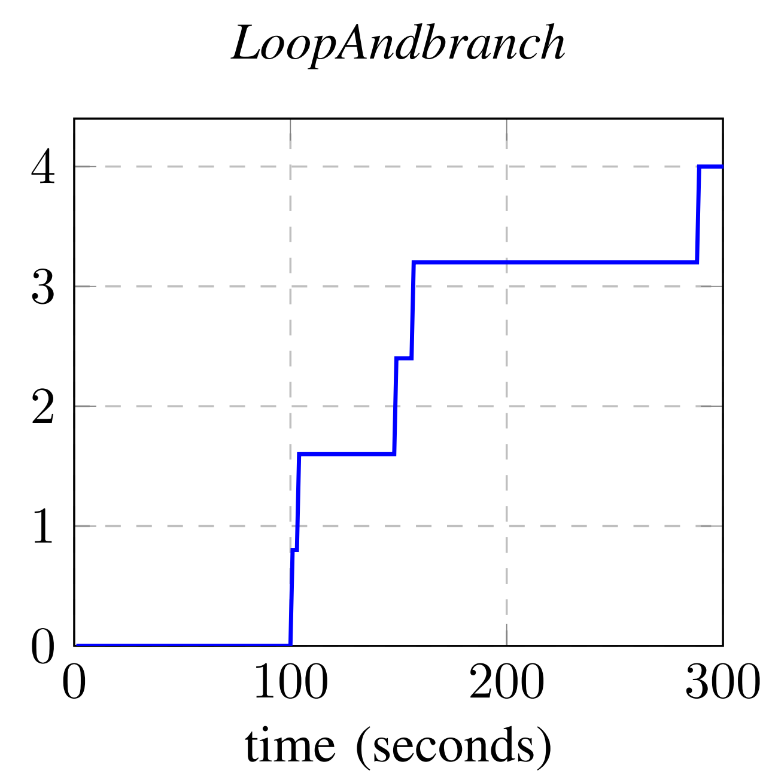
```
secret₁ = [72, 77, -16, -66, -48, -48, -48, -48,
secret₂ = [-48, -4, -48, 7, 17, 0, -24, -48, -48,
public  = [-48, -4, -48, 7, 17, 0, -24, -48, -48,

             -28, 0, 100, 0, 0, 0, 0, -48]
        ↪    16, -48, -3, 108, 72, 32, 0]
             16, -48, -3, 108, 72, 32, 0]
```

Evaluation

- comparison with **Blazer** and **Themis,** two state-of-the-art static analysis tools for detecting side-channel vulnerabilities in JAVA programs; extract is shown on the right →
- additional benchmarks from DARPA Space/Time Analysis für Cybersecurity (STAC) program
- new vulnerabilities found in Apache ftpserver
- 3 different timing behaviors observed:







Benchmark	Version	DifFuzz	Themis	
		Average δ	ϵ = 64	ϵ = 0
Spring-Security	Safe	1.00	✓	✓
	Unsafe	149.00	✓	✓
JDK-MsgDigest	Safe	1.00	✓	✓
	Unsafe	10,215.00	✓	✓
Picketbox	Safe	1.00	✓	X
	Unsafe	4,954.00	✓	✓
Tomcat	Safe	12.20	✓	X
	Unsafe	*33,20*	✓	✓
Jetty	*Safe*	*5454.00*	✓	✓
	Unsafe	10,786.60	✓	✓
oriented	Safe	6.00	✓	X
	Unsafe	6,604.00	✓	✓
pac4j	Safe	10.00	✓	X
	Unsafe	*11.00*	✓	✓
	Unsafe*	39.00	-	-
boot-auth	Safe	5.00	✓	X
	Unsafe	101.00	✓	✓
tourPlanner	Safe	0.00	✓	✓
	Unsafe	522.40	✓	✓
DynaTable	Unsafe	95.80	✓	✓
Advanced_table	Unsafe	92.40	✓	✓
OpenMRS	Unsafe	206.00	✓	✓
OACC	*Unsafe*	*47.00*	✓	✓

`git clone https://github.com/isstac/diffuzz.git`

`yannic.noller@hu-berlin.de`