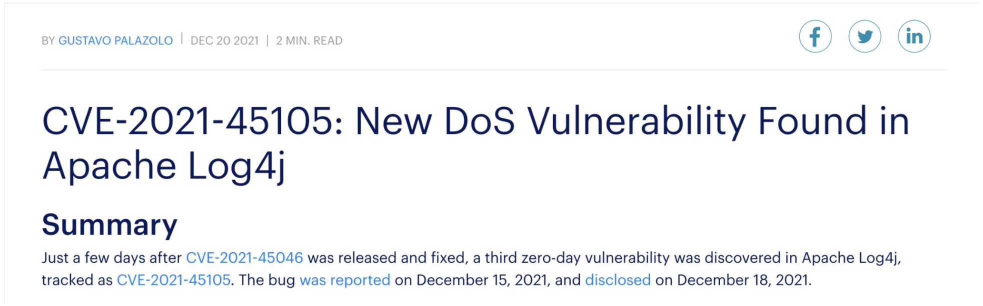
CONFETTI: Amplifying Concolic Guidance for Fuzzers

James Kukucka, Luís Pina, Paul Ammann, Jonathan Bell

Motivation - CVE-2021-45105 log4j DoS Vulnerability

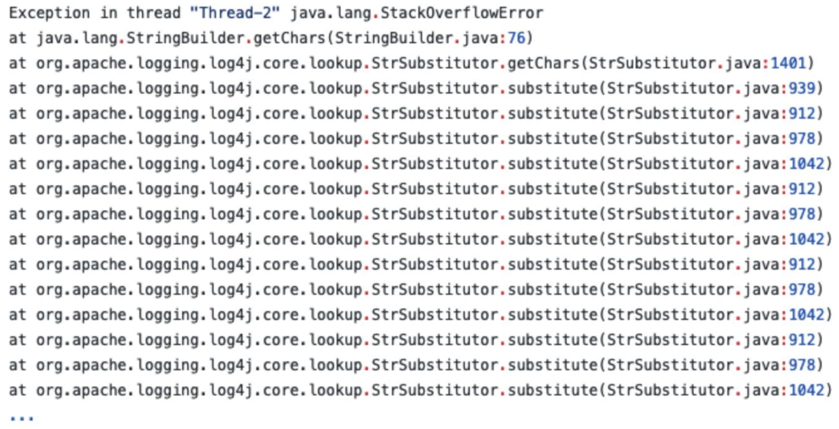


BY GUSTAVO PALAZOLO | DEC 20 2021 | 2 MIN. READ

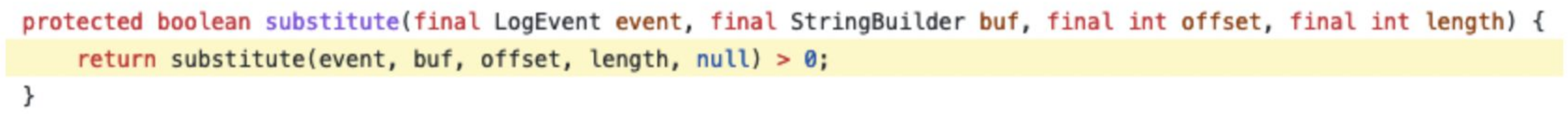
CVE-2021-45105: New DoS Vulnerability Found in Apache Log4j

Summary

Just a few days after CVE-2021-45046 was released and fixed, a third zero-day vulnerability was discovered in Apache Log4j, tracked as CVE-2021-45105. The bug was reported on December 15, 2021, and disclosed on December 18, 2021.



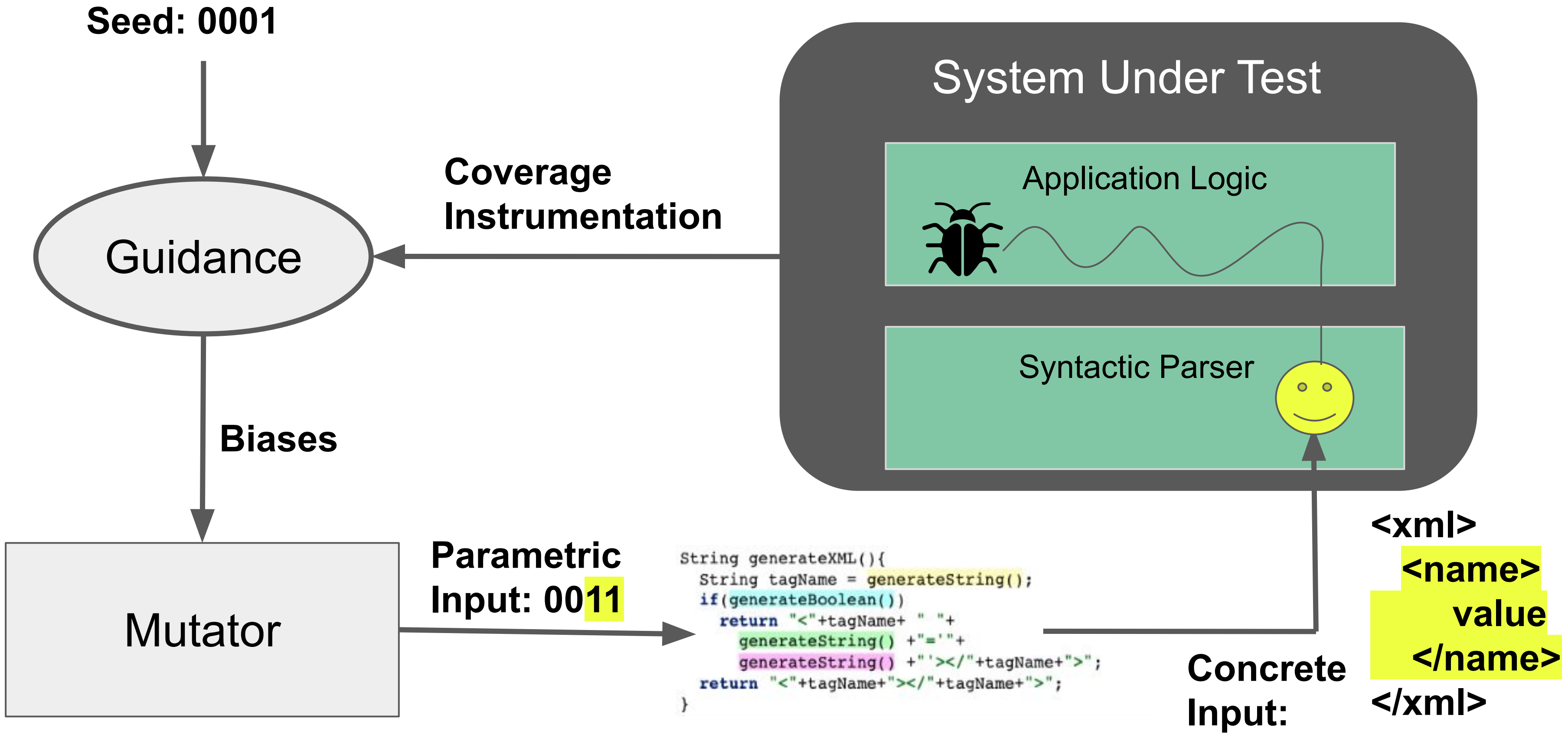
```
Exception in thread "Thread-2" java.lang.StackOverflowError
at java.lang.StringBuilder.getChars(StringBuilder.java:76)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.getChars(StrSubstitutor.java:1401)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:939)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:912)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:978)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:1042)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:912)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:978)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:1042)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:912)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:978)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:1042)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:912)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:978)
at org.apache.logging.log4j.core.lookup.StrSubstitutor.substitute(StrSubstitutor.java:1042)
...
```



```
protected boolean substitute(final LogEvent event, final StringBuilder buf, final int offset, final int length) {
    return substitute(event, buf, offset, length, null) > 0;
}
```

https://www.netskope.com/blog/cve-2021-45105-new-dos-vulnerability-found-in-apache-log4j

The state-of-the-art method for finding these bugs is parametric fuzzing.



```
String generateXML(){
  String tagName = generateString();
  if(generateBoolean())
    return "<"+tagName+ " "+
      generateString() +"='"+
      generateString() +"'></"+tagName+">";
  return "<"+tagName+"></"+tagName+">";
}
```

```
<xml>
  <name>
      value
    </name>
</xml>
```

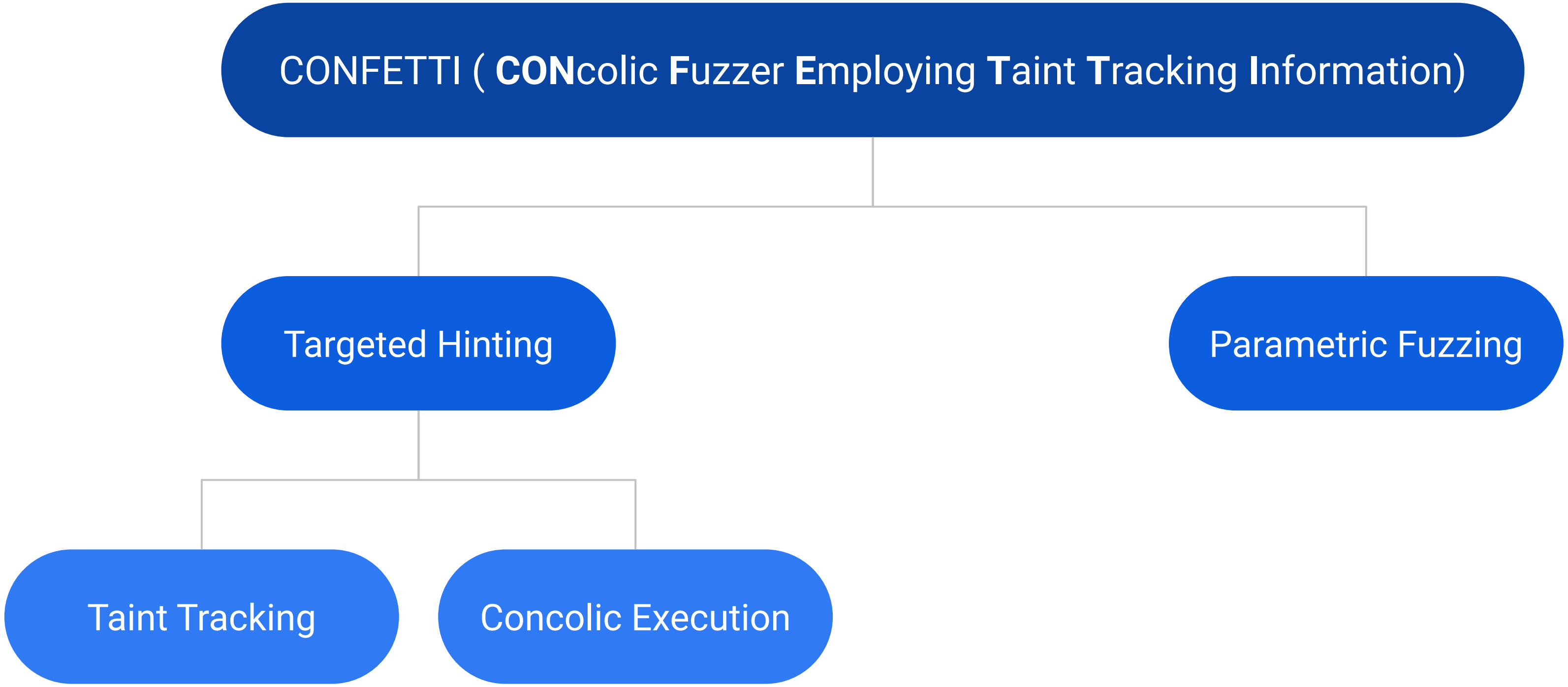
Our solution, CONFETTI, leverages state-of-the-art parametric fuzzing and novel hinting to increase coverage.

CONFETTI (**CON**colic **F**uzzer **E**mploying **T**aint **T**racking **I**nformation)

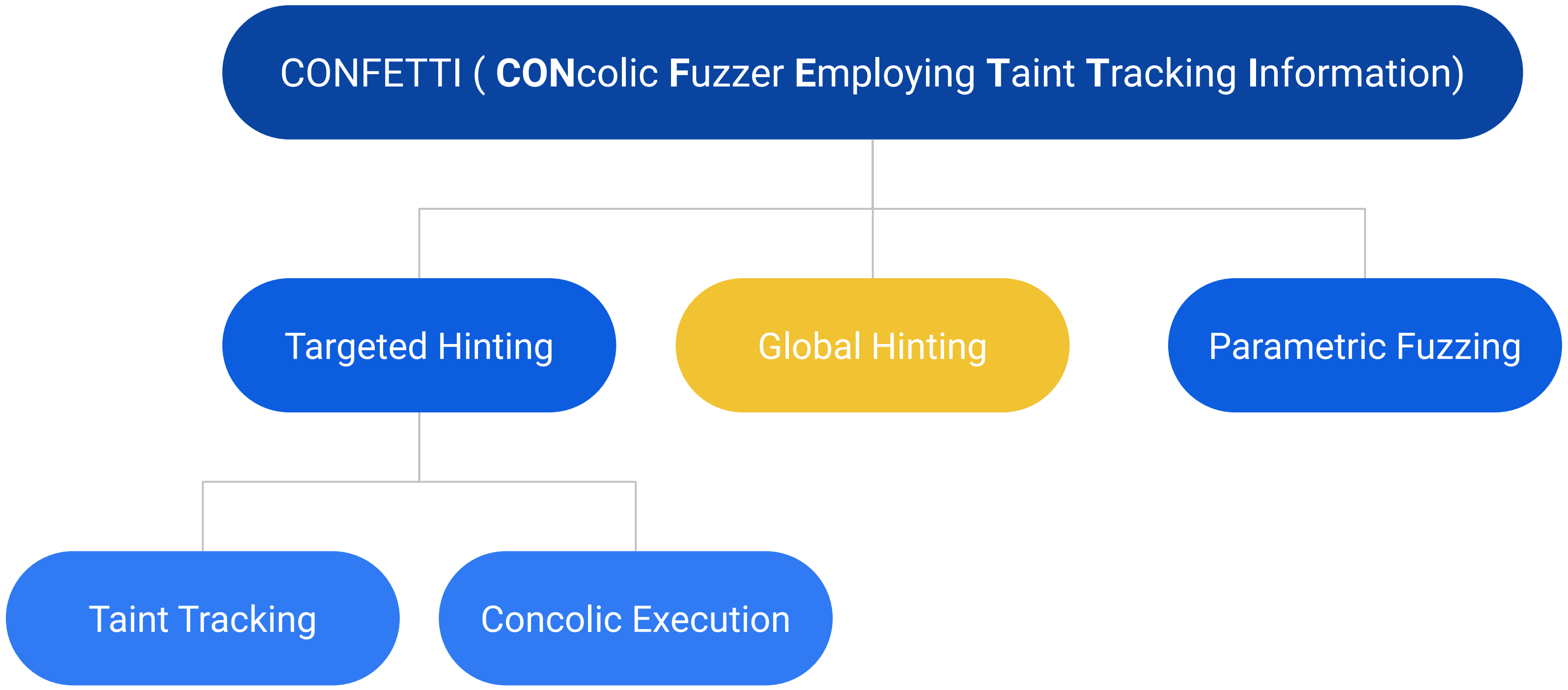
Our solution, CONFETTI, leverages state-of-the-art parametric fuzzing and novel hinting to increase coverage.

CONFETTI (**CON**colic **F**uzzer **E**mploying **T**aint **T**racking **I**nformation)
Parametric Fuzzing

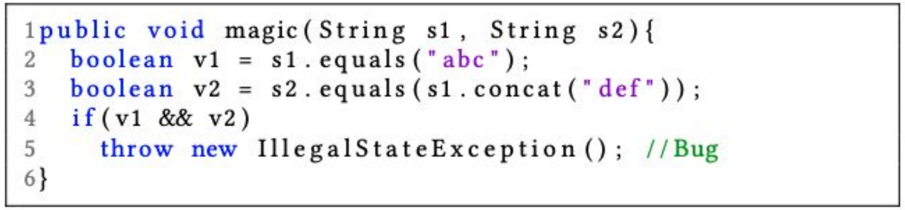
Our solution, CONFETTI, leverages state-of-the-art parametric fuzzing and novel hinting to increase coverage.


CONFETTI (**CON**colic **F**uzzer **E**mploying **T**aint **T**racking **I**nformation)
Targeted Hinting
Parametric Fuzzing
Taint Tracking
Concolic Execution

Our solution, CONFETTI, leverages state-of-the-art parametric fuzzing and novel hinting to increase coverage.

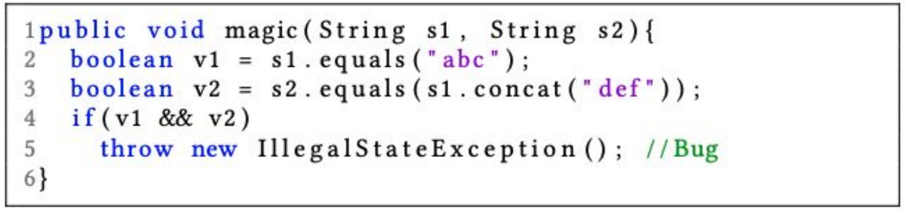

CONFETTI (**CON**colic **F**uzzer **E**mploying **T**aint **T**racking **I**nformation)
Targeted Hinting
Global Hinting
Parametric Fuzzing
Taint Tracking
Concolic Execution

Global hinting allows CONFETTI to explore branches it could not otherwise.



```java
public void magic(String s1, String s2){
  boolean v1 = s1.equals("abc");
  boolean v2 = s2.equals(s1.concat("def"));
  if(v1 && v2)
    throw new IllegalStateException(); //Bug
}
```

Global hinting allows CONFETTI to explore branches it could not otherwise.

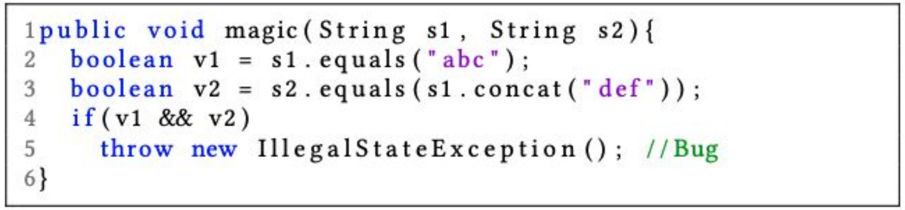


```java
public void magic(String s1, String s2){
  boolean v1 = s1.equals("abc");
  boolean v2 = s2.equals(s1.concat("def"));
  if(v1 && v2)
    throw new IllegalStateException(); //Bug
}
```

Static Dictionary

“abc”
“def”

Global hinting allows CONFETTI to explore branches it could not otherwise.



```java
public void magic(String s1, String s2){
  boolean v1 = s1.equals("abc");
  boolean v2 = s2.equals(s1.concat("def"));
  if(v1 && v2)
    throw new IllegalStateException(); //Bug
}
```

Taint Tracking

Global Hints

”abcdef”

Static Dictionary

“abc”
“def”

```
public String generateString(ParametricInputArray r) {
        if( r.nextBoolean() )
        {
                return static_dict[r.nextInt()];
        }
        return global_hints[r.nextInt()];
}

s1 = generateString(r); // picks randomly from static dictionary to yield “abc”
s2 = generateString(r); // picks randomly from global hints to yield “abcdef”
```

On most benchmark programs, the use of CONFETTI's global hinting with targeted hinting resulted in higher branch coverage and more bugs found.

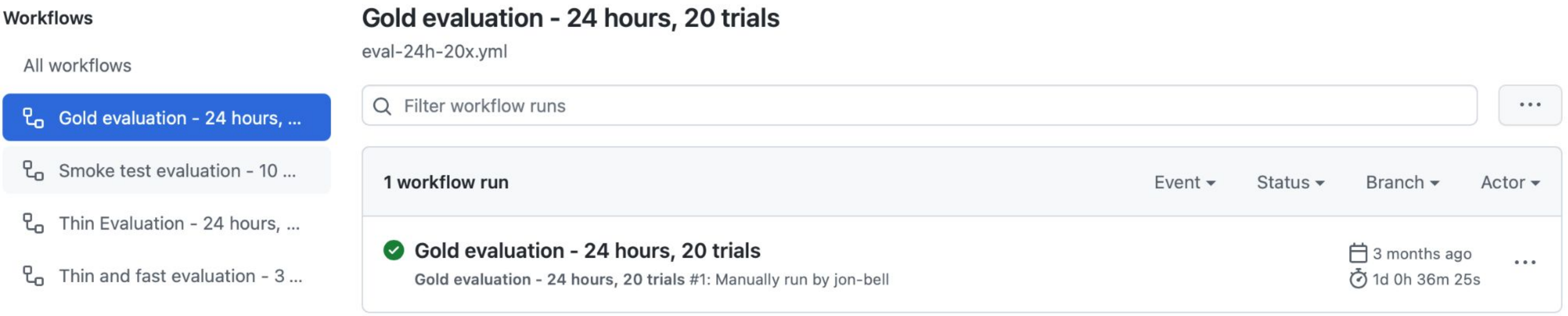
		Total Branch Coverage			Bugs Found		
Program	Total Branches	Zest	CONFETTI_tgt	CONFETTI	Zest	CONFETTI_tgt	CONFETTI
ant	23,361	859	871	872	1	1	1
bcel	6,220	1361	1423	1421	2	3	5
closure	49,602	10,545	10,640	11,458	4	8	15
maven	5,858	821	853	857	0	0	0
rhino	25,035	3,757	3,534	3,744	4	4	4

Our evaluation, all data, and CONFETTI itself are archived and open-source

https://doi.org/10.6084/m9.figshare.16563776

https://github.com/neu-se/confetti

Continuous integration workflow allows for easy evaluation



Workflows

All workflows

Gold evaluation - 24 hours, ...

Smoke test evaluation - 10 ...

Thin Evaluation - 24 hours, ...

Thin and fast evaluation - 3 ...

Gold evaluation - 24 hours, 20 trials

eval-24h-20x.yml

Filter workflow runs

1 workflow run

Event Status Branch Actor

✓ Gold evaluation - 24 hours, 20 trials

Gold evaluation - 24 hours, 20 trials #1: Manually run by jon-bell

3 months ago

1d 0h 36m 25s