

# Testing the Binding Code of Scripting Languages with Cooperative Mutation

Peng Xu   Yanhao Wang   Hong Hu   Purui Su

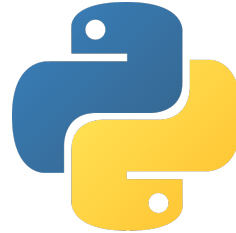


**PennState**

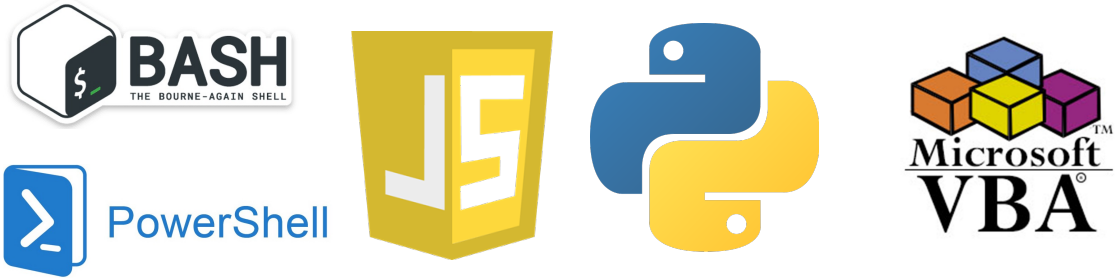


# Scripting Language

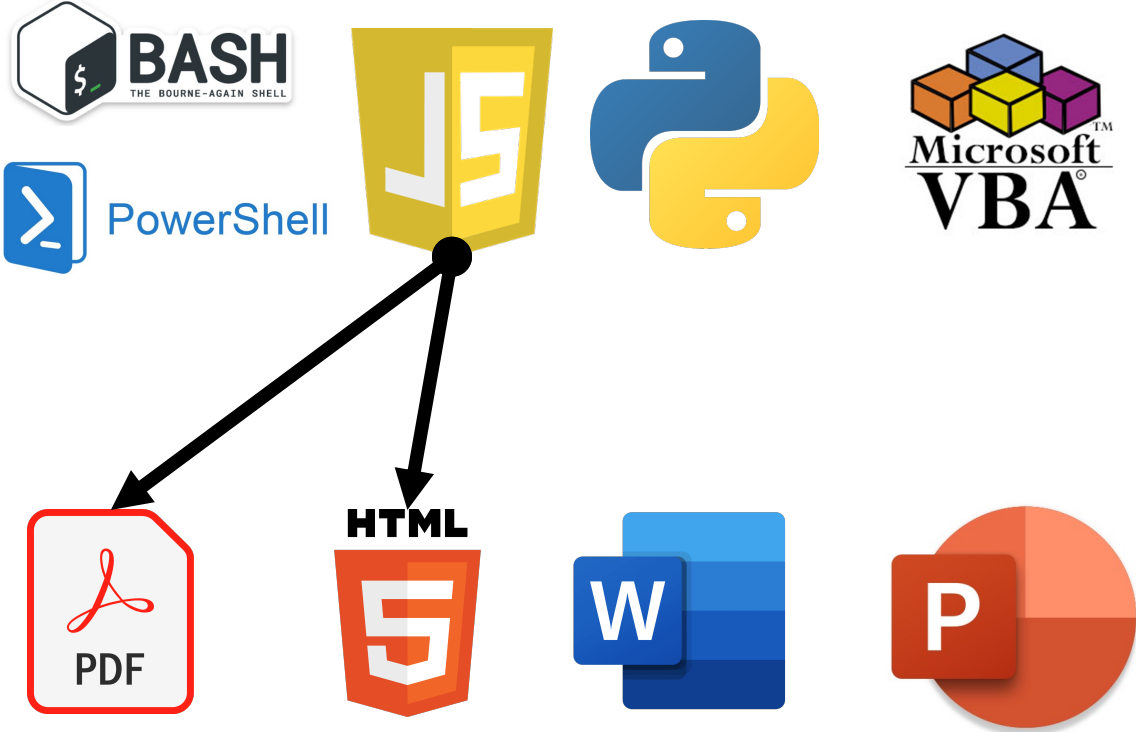
# Scripting Language



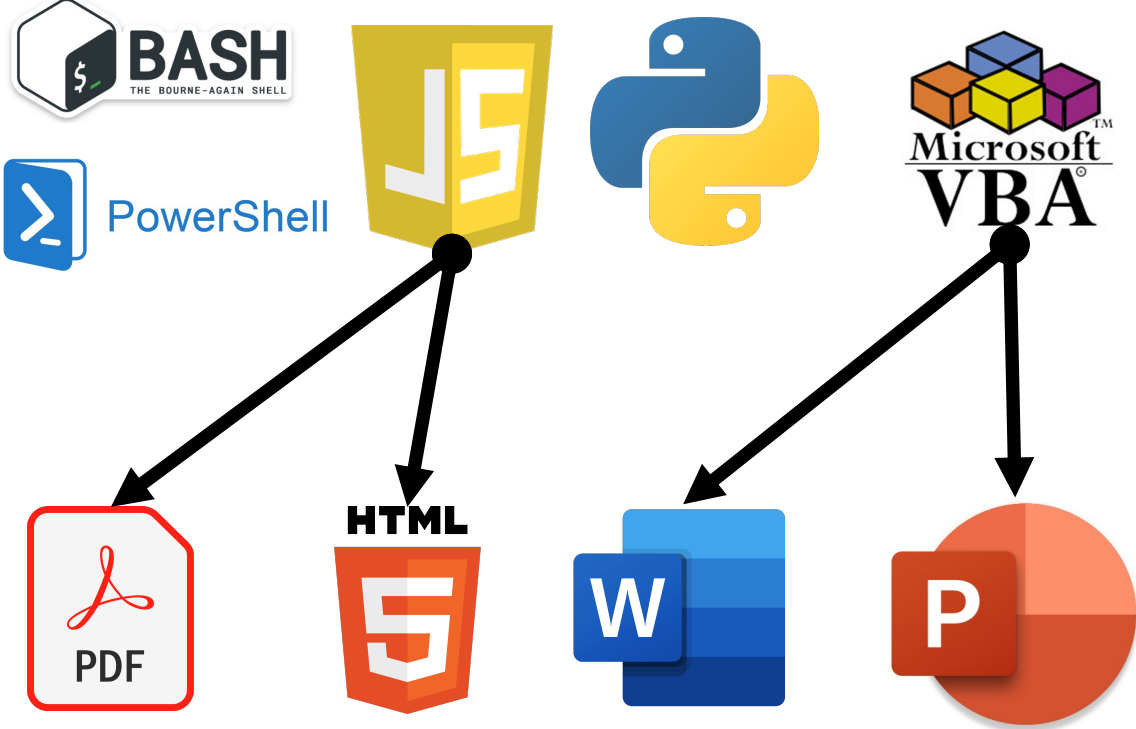
# Scripting Language integrated to documents



# Scripting Language integrated to documents



# Scripting Language integrated to documents



# Vulnerability in Embedded Scripting Language

- Dangerous and Common

# Vulnerability in Embedded Scripting Language

- Dangerous and Common

## Multiple Vulnerabilities in Adobe Acrobat and Adobe Reader Could Allow for Arbitrary Code Execution (APSB20-67)

MS-ISAC ADVISORY NUMBER:  
2020-150

DATE(S) ISSUED:  
11/03/2020

### OVERVIEW:

Multiple vulnerabilities have been discovered in Adobe Acrobat and Adobe Reader, the most severe of which could allow for arbitrary code execution. Adobe Acrobat is a family of software developed by Adobe Inc. to view, create, manipulate, print, and manage files in PDF format. Adobe Reader is the free version within the Adobe Acrobat family of software. Successful exploitation of the most severe of these vulnerabilities could result in arbitrary code execution. Depending on the privileges associated with the user, an attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. If this application has been configured to have fewer user rights on the system, exploitation of the most severe of these vulnerabilities could have less impact than if it was configured with administrative rights.

## Pwn2Own 2020 – Participants hacked Adobe Reader, Oracle VirtualBox, and Windows

March 20, 2020 By Pierluigi Paganini

Home / Security / News

NEWS

## Update Google Chrome right now to patch a dangerous exploit

The latest version patches a bug that's being actively attacked.



By **Michael Crider**  
Staff Writer, PCWorld | FEB 15, 2022 7:58 AM PST

## Vulnerability in Adobe Acrobat and Reader being actively exploited

Adobe has released a patch to fix critical vulnerabilities in Adobe Acrobat and Adobe Reader. CVE-2021-28550 has been actively exploited and is a use-after-free arbitrary code execution vulnerability. This vulnerability can be exploited by opening a specially crafted PDF file in a vulnerable version of Adobe Acrobat or Adobe Reader.

CERT NZ recommends all users of these programs to immediately update Adobe Acrobat and Adobe Reader.



# Our work: Cooper

# Our work: Cooper

- Cooperative mutation
  - effectively test binding code of scripting languages

# Our work: Cooper

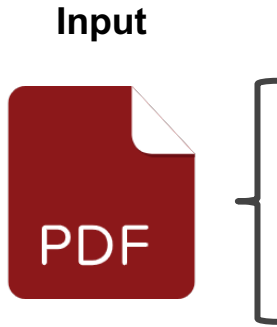
- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty

# Our work: Cooper

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty
- Open-sourced at: <https://github.com/TCA-ISCAS/Cooper>

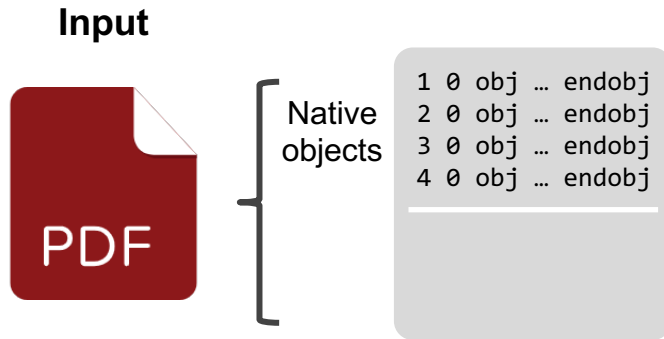
# Document processing programs

- Input: Native objects + Scripts code



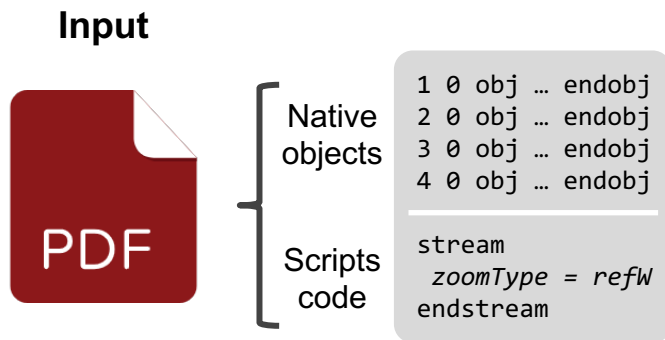
# Document processing programs

- Input: Native objects + Scripts code



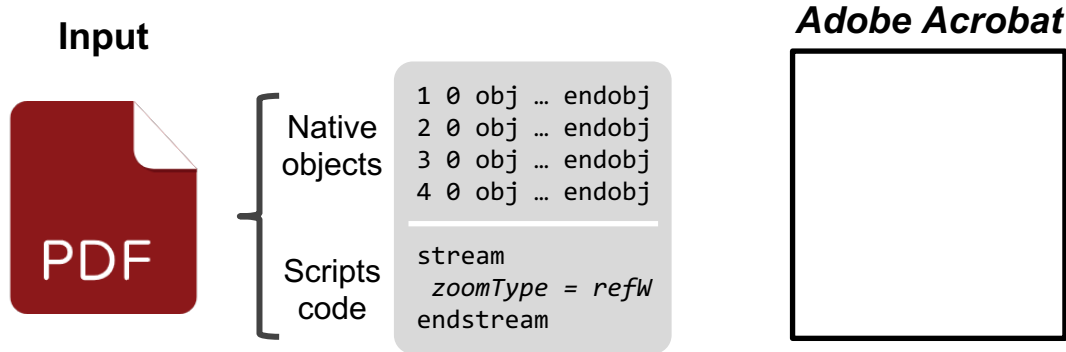
# Document processing programs

- Input: Native objects + Scripts code



# Document processing programs

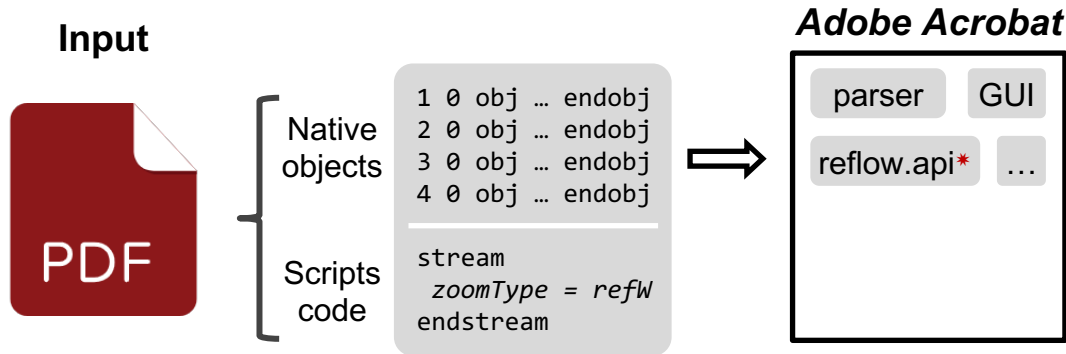
- Input: Native objects + Scripts code
- Two components for processing inputs





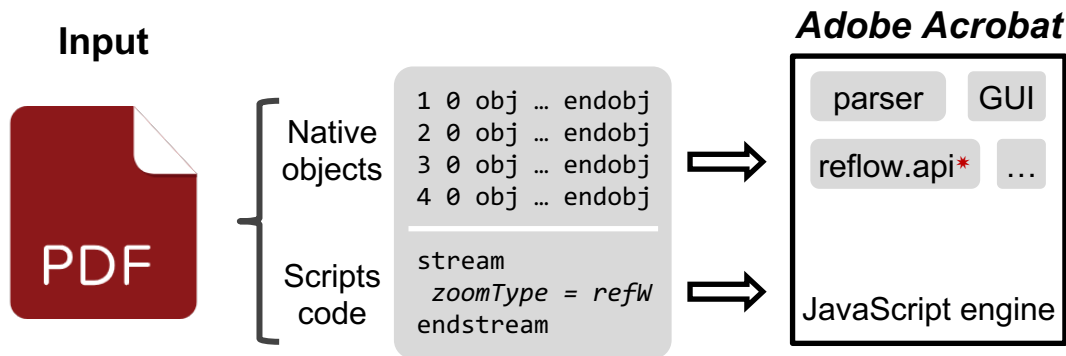
# Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs



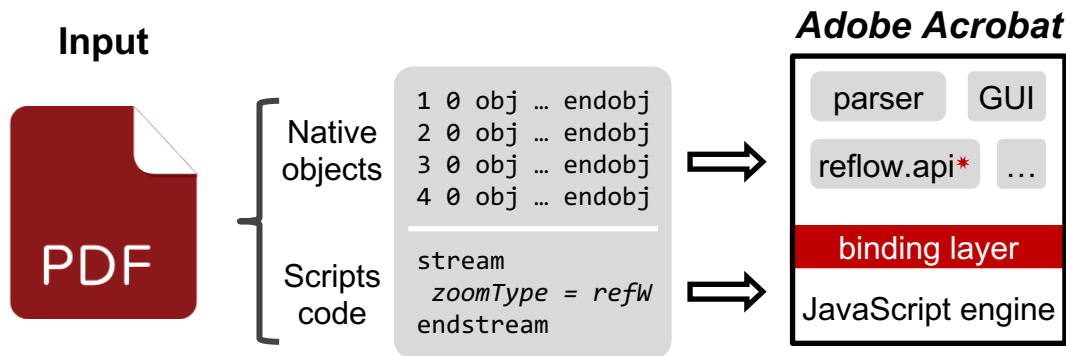
# Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs



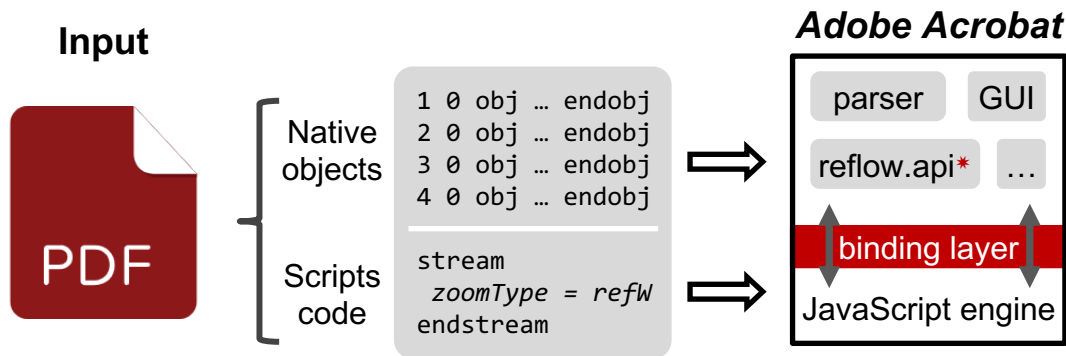
# Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



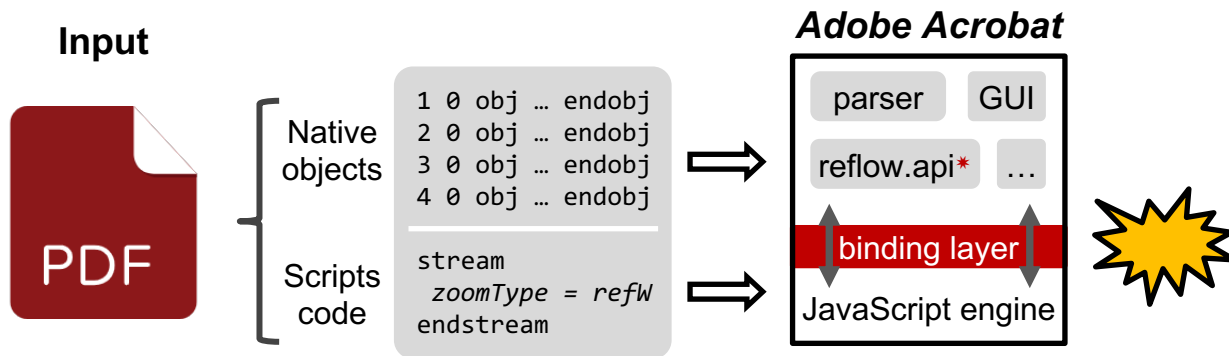
# Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



# Document processing programs

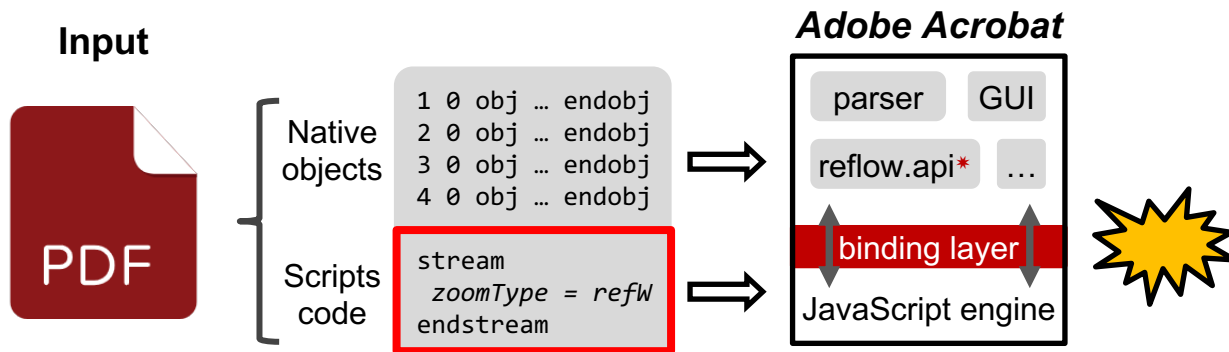
- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



- Binding layer is too complicated, leading to BUGS

# Document processing programs

- Input: Native objects + Scripts code
- Two components for processing inputs
- Binding layer connects two components



Previous work only mutates scripts code

- Binding layer is too complicated, leading to BUGS

# Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- \$2.5K bounty

```
1 %PDF-1.3
2 1 0 obj << /Pages 2 0 R >> endobj
3 2 0 obj << /Kids [ 3 0 R ] >> endobj
4 3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5     /AA << /O << /S /JavaScript
6         /JS 5 0 R >> >> >> endobj
7 4 0 obj << /FirstChar 0
8     /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9     /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11     stream
12         this.zoomType=zoomtype.refW; % Trigger the bug
13     endstream
14     endobj
15 trailer << /Root 1 0 R >>
```

# Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- \$2.5K bounty

How to trigger this vulnerability?

```
1 %PDF-1.3
2 1 0 obj << /Pages 2 0 R >> endobj
3 2 0 obj << /Kids [ 3 0 R ] >> endobj
4 3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5     /AA << /O << /S /JavaScript
6         /JS 5 0 R >> >> >> endobj
7 4 0 obj << /FirstChar 0
8     /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9     /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11     stream
12         this.zoomType=zoomtype.refW; % Trigger the bug
13     endstream
14     endobj
15 trailer << /Root 1 0 R >>
```



# Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- \$2.5K bounty

## How to trigger this vulnerability?

- Native Objects:  
Insert an extra element into Font's Widths array.

```
1 %PDF-1.3
2 1 0 obj << /Pages 2 0 R >> endobj
3 2 0 obj << /Kids [ 3 0 R ] >> endobj
4 3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5     /AA << /O << /S /JavaScript
6         /JS 5 0 R >> >> >> endobj
7 4 0 obj << /FirstChar 0
8     /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9     /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11     stream
12         this.zoomType=zoomtype.refW; % Trigger the bug
13     endstream
14     endobj
15 trailer << /Root 1 0 R >>
```

# Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- \$2.5K bounty

## How to trigger this vulnerability?

- Native Objects:  
Insert an extra element into Font's Widths array.
- Scripts Code:  
Invoke `this.zoomType=zoomtype.refW;`

```
1 %PDF-1.3
2 1 0 obj << /Pages 2 0 R >> endobj
3 2 0 obj << /Kids [ 3 0 R ] >> endobj
4 3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5     /AA << /O << /S /JavaScript
6         /JS 5 0 R >> >> >> endobj
7 4 0 obj << /FirstChar 0
8     /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9     /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11     stream
12     this.zoomType=zoomtype.refW; % Trigger the bug
13     endstream
14     endobj
15 trailer << /Root 1 0 R >>
```

# Motivating Example

- Heap Overflow in Adobe Acrobat
- Remote Code Execution
- \$2.5K bounty

## How to trigger this vulnerability?

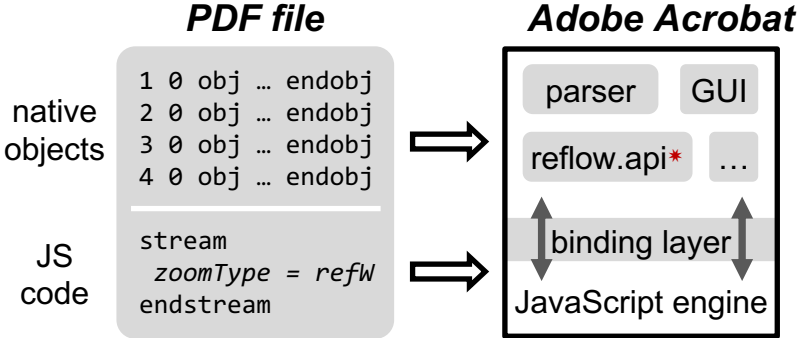
- Native Objects:  
Insert an extra element into Font's Widths array.
- Scripts Code:  
Invoke `this.zoomType=zoomtype.refW;`

```
1 %PDF-1.3
2 1 0 obj << /Pages 2 0 R >> endobj
3 2 0 obj << /Kids [ 3 0 R ] >> endobj
4 3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
5     /AA << /O << /S /JavaScript
6         /JS 5 0 R >> >> >> endobj
7 4 0 obj << /FirstChar 0
8     /Widths [ 778 778 ... 556 500 ] % 256 + 1 elements
9     /LastChar 255 >> endobj
10 5 0 obj << /Length 539 >>
11     stream
12     this.zoomType=zoomtype.refW; % Trigger the bug
13     endstream
14     endobj
15 trailer << /Root 1 0 R >>
```

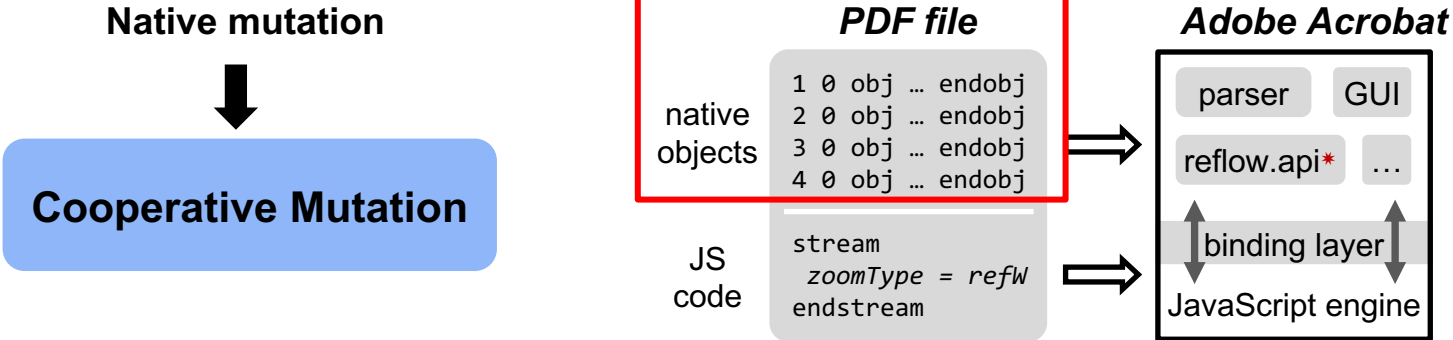
One-dimension-mutation cannot trigger this vulnerability

# Our Solution: Cooperative Mutation

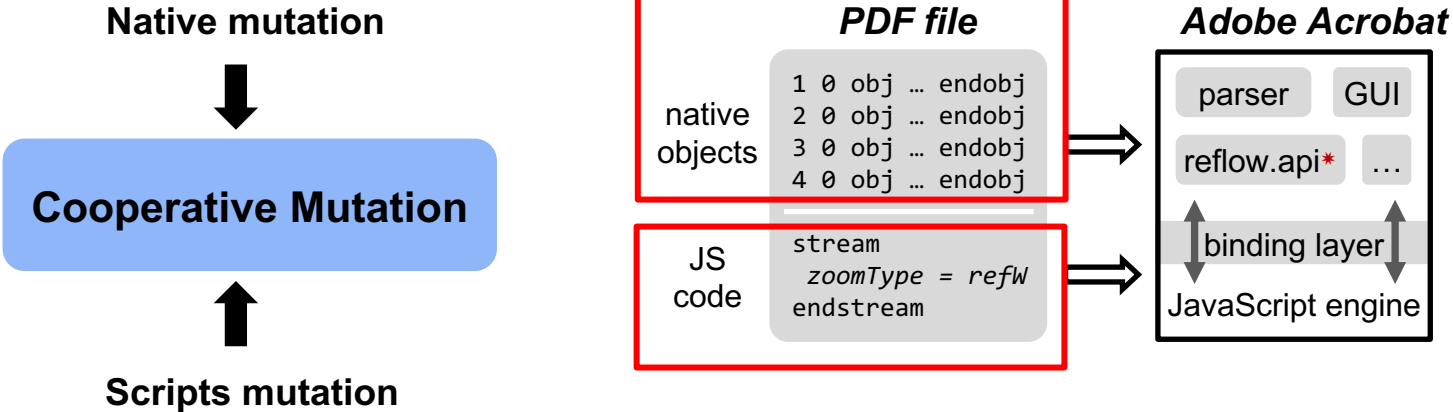
**Cooperative Mutation**



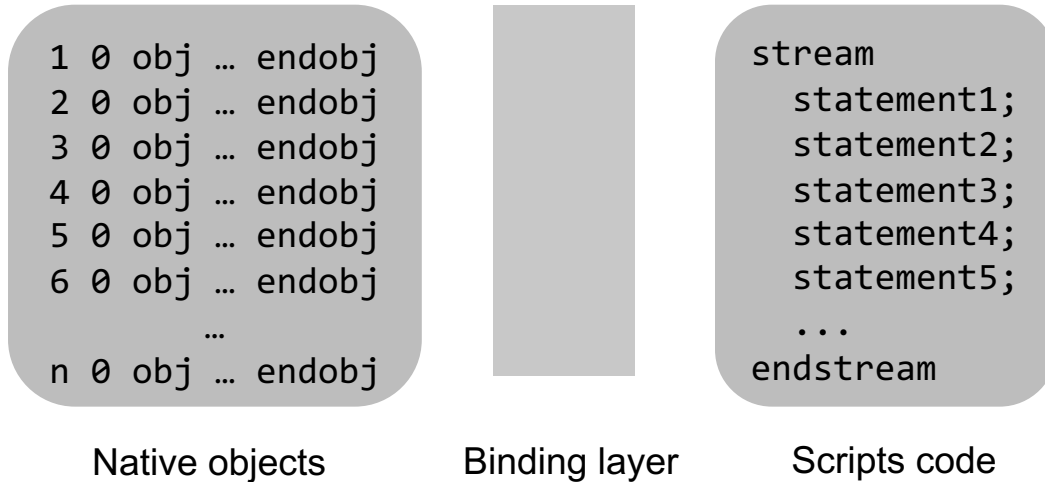
# Our Solution: Cooperative Mutation



# Our Solution: Cooperative Mutation



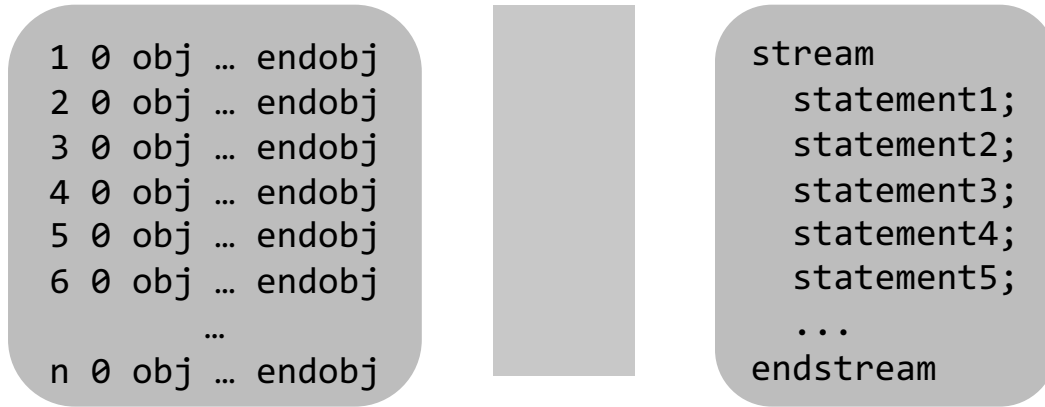
# How to mutate objects & code cooperatively?



Mutate Native objects

Mutate Scripts code

# How to mutate objects & code cooperatively?



Native objects

Binding layer

Scripts code

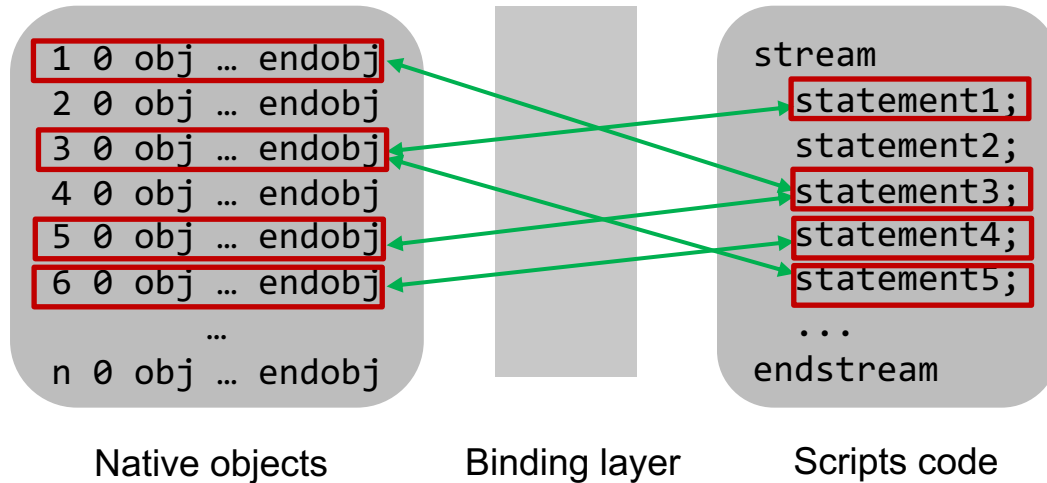
Mutate Native objects

Relationship

Mutate Scripts code



# How to mutate objects & code cooperatively?

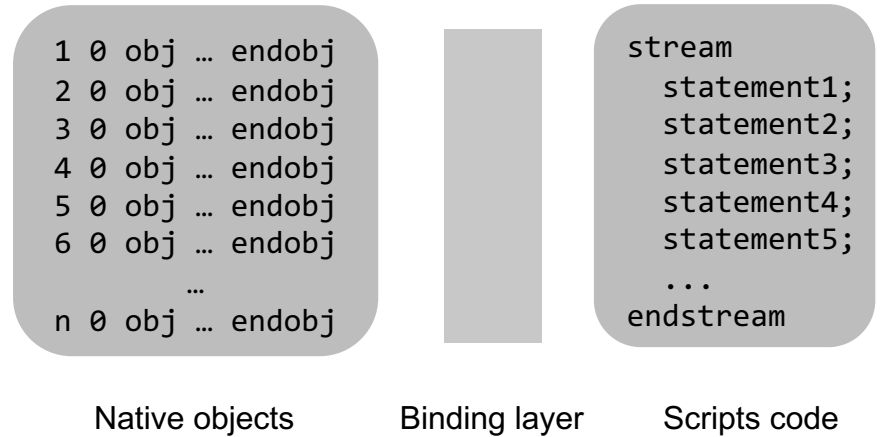


Mutate Native objects

Relationship

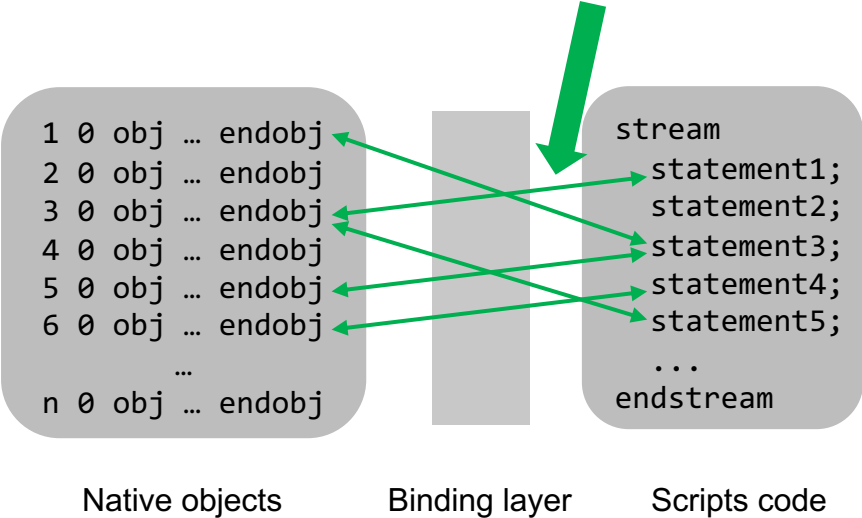
Mutate Scripts code

# Challenges

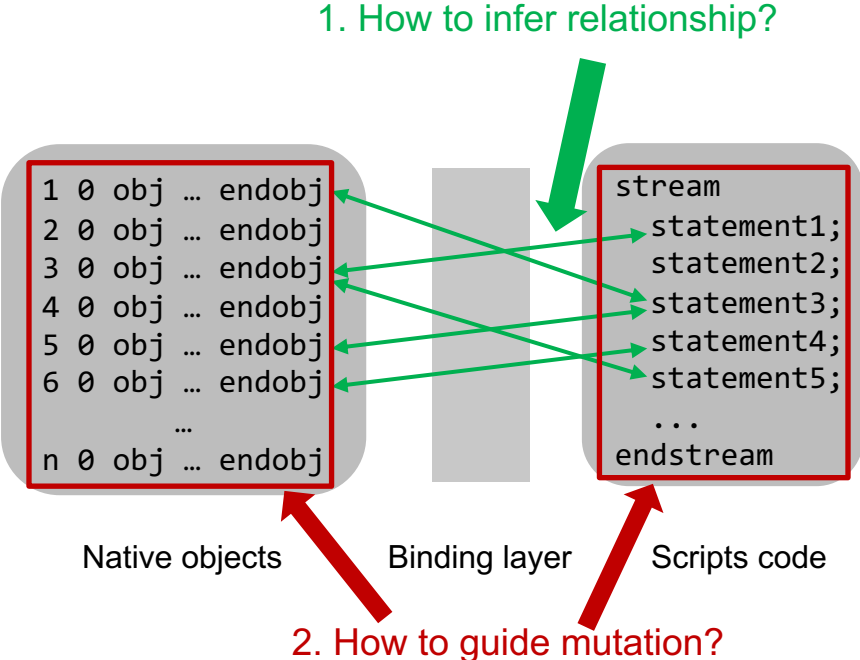


# Challenges

## 1. How to infer relationship?

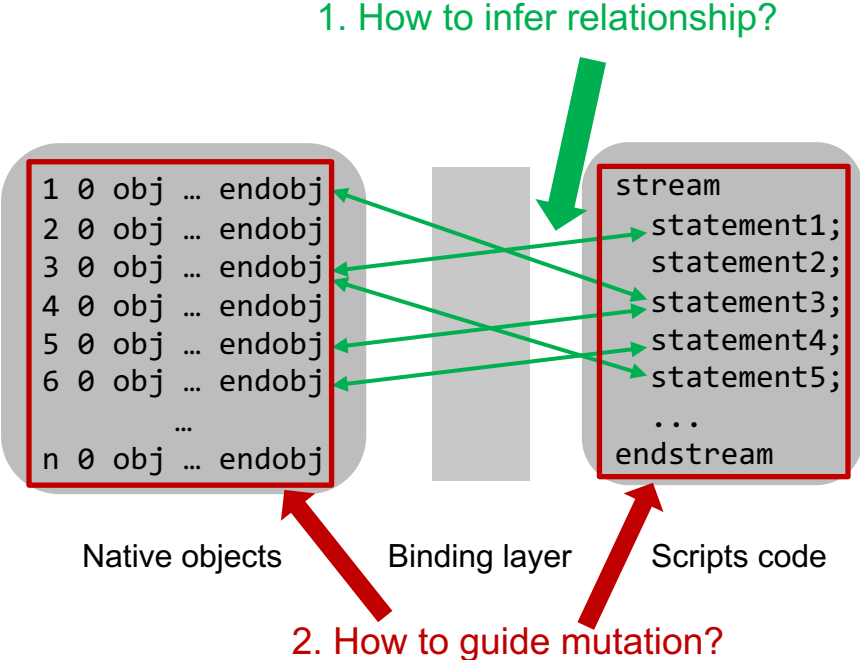


# Challenges



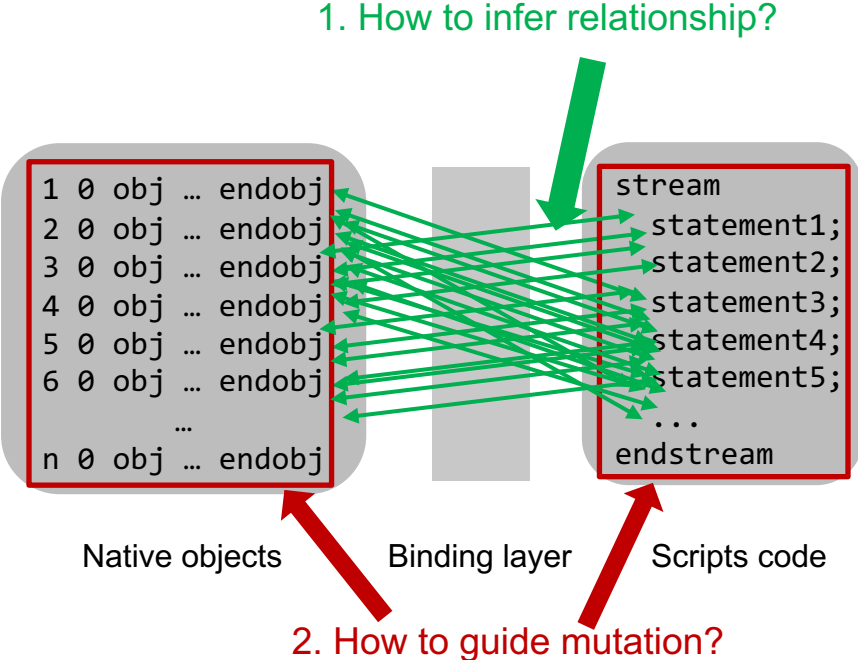
# Challenges

Too many objects!!! Makes it hard for inferring and mutation



# Challenges

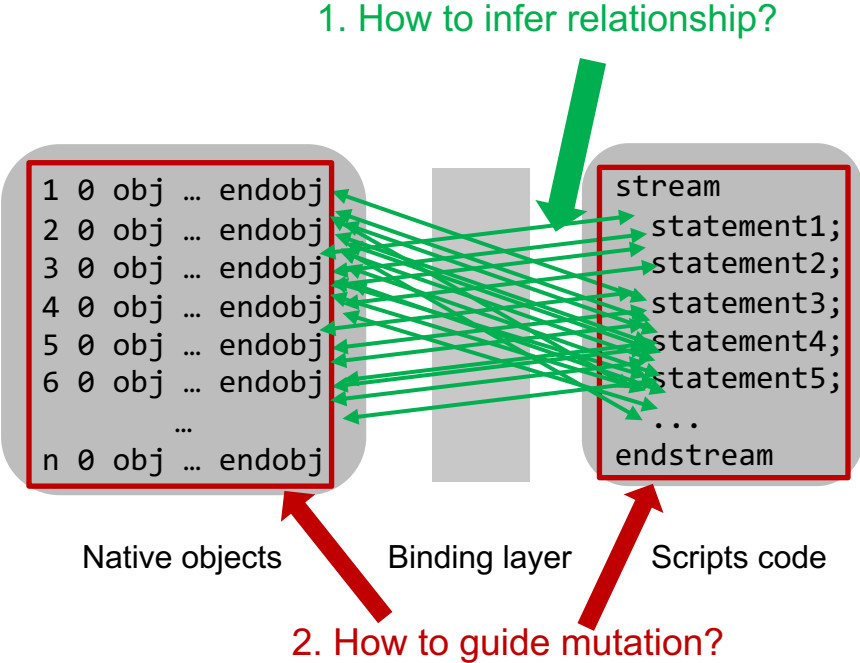
Too many objects!!! Makes it hard for inferring and mutation



# Challenges

Too many objects!!! Makes it hard for inferring and mutation

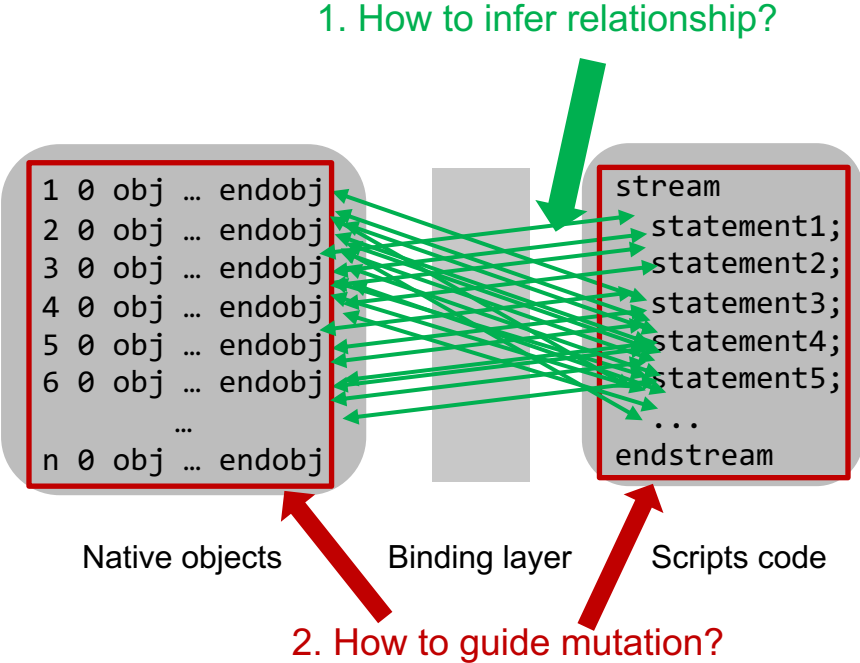
We need to reduce the search space of native objects.



# Challenges

Too many objects!!! Makes it hard for inferring and mutation

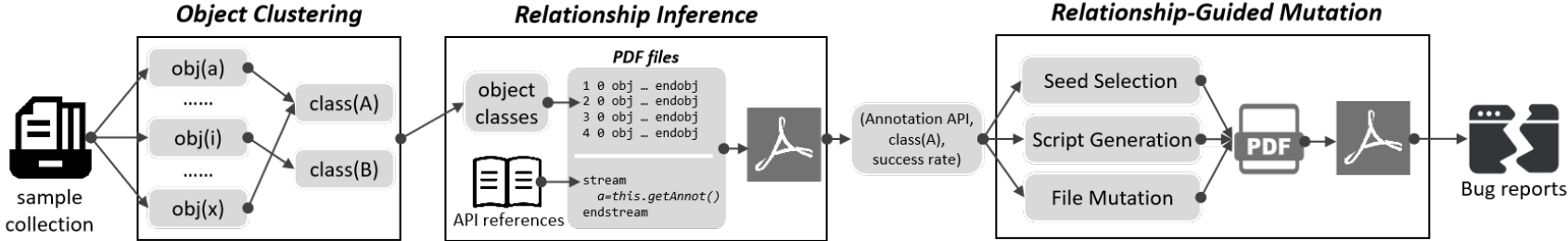
We need to reduce the search space of native objects.



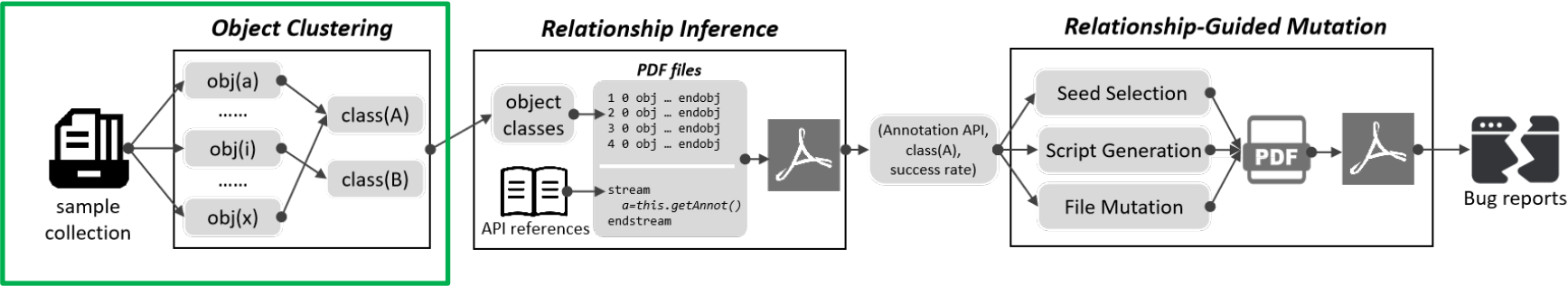
0. How to cluster native objects?



# Cooper Overview

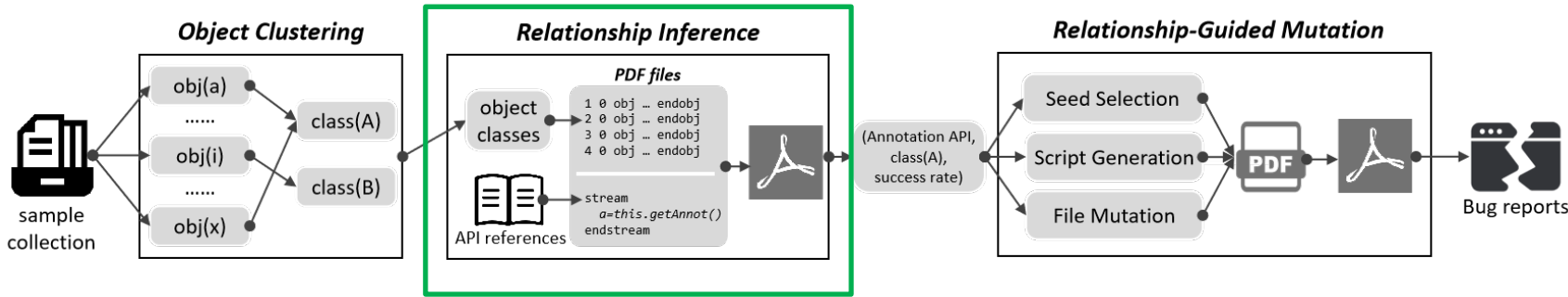


# Cooper Overview



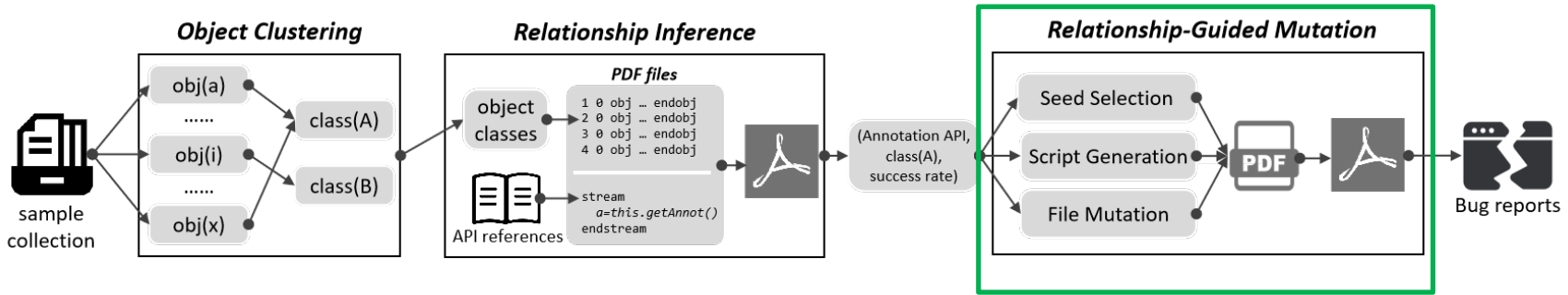
cluster input objects to semantic-similar classes

# Cooper Overview



infer relation between native input and script code

# Cooper Overview



use the inferred relation to guide mutation

# Object Clustering

# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \quad \dots = \dots \end{array} \right\}$$

# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \quad \dots = \dots \end{array} \right\}$$

**Name contains semantic information**

# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \quad \dots = \dots \end{array} \right\}$$

**Name contains semantic information**

- Clustering objects with **name**



# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \quad \dots = \dots \end{array} \right\}$$

**Name contains semantic information**

- Clustering objects with **name**

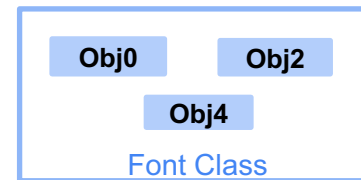
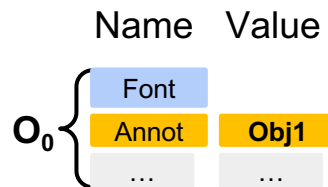
	Name	Value
$O_0$	Font	Obj0
	Annot	Obj1
	...	...
$O_1$	Font	Obj2
	Annot	Obj3
	...	...
$O_2$	Font	Obj4
	...	...

# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \dots = \dots \end{array} \right\}$$

**Name contains semantic information**

- Clustering objects with **name**

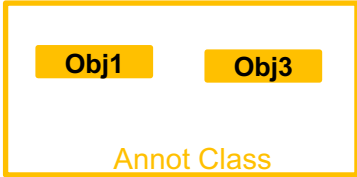
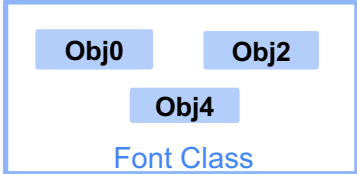
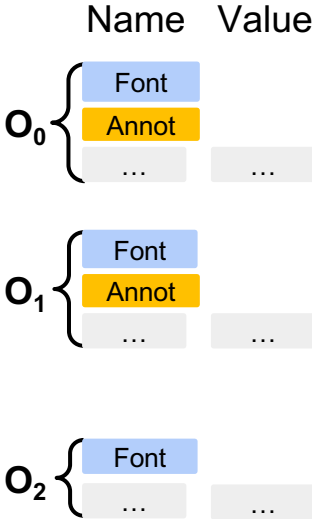


# Object Clustering

$$O:\text{object} = \left\{ \begin{array}{l} A_0: \text{name}_0 = \text{object}_0, \\ A_1: \text{name}_1 = \text{object}_1, \\ A_2: \text{name}_2 = \text{object}_2, \\ \dots = \dots \end{array} \right\}$$

**Name contains semantic information**

- Clustering objects with **name**



# Object Clustering

- Splitting and merging classes with attribute similarity

# Object Clustering

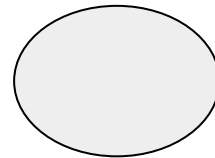
- Splitting and merging classes with attribute similarity

$$Sim(A, B) = \frac{2(|A \cap B|)}{|A| + |B|}$$

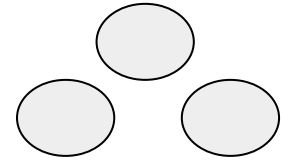
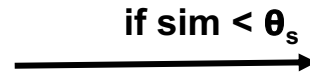
# Object Clustering

- Splitting and merging classes with attribute similarity

$$Sim(A, B) = \frac{2(|A \cap B|)}{|A| + |B|}$$



Big Class

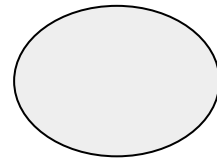


Small Classes

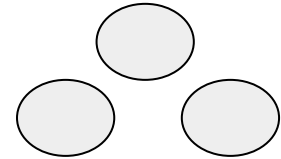
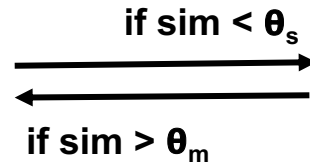
# Object Clustering

- Splitting and merging classes with attribute similarity

$$Sim(A, B) = \frac{2(|A \cap B|)}{|A| + |B|}$$



Big Class



Small Classes

# Relationship Inference

- Run & Record



# Relationship Inference

- Run & Record

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

# Relationship Inference

- Run & Record

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

# Relationship Inference

- Run & Record

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```


```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -

 2 Annots Found

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```


```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -

 2 Annots Found

Success Set

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```

```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -



2 Annots Found

Success Set

Warning: JavaScript Window -



0 Annots Found

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```


```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -

 2 Annots Found

Success Set

Warning: JavaScript Window -


 0 Annots Found

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```

```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```


Warning: JavaScript Window -

 ERROR

# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -

 2 Annots Found

Success Set

Warning: JavaScript Window -

 0 Annots Found


Failure Set

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```

```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

Warning: JavaScript Window -

 ERROR



# Relationship Inference

## ● Run & Record

Warning: JavaScript Window -

● 2 Annots Found

Success Set

Warning: JavaScript Window -

● 0 Annots Found

Failure Set

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
```

```
try{
  var annots = this.getAnnot();
  app.alert(annots.length+" Annots Found");
}catch(e){ app.alert("ERROR" + e); }
```

```
4 0 obj ... endobj
5 0 obj ... endobj
6 0 obj ... endobj
```

Warning: JavaScript Window -

● ERROR



Samples

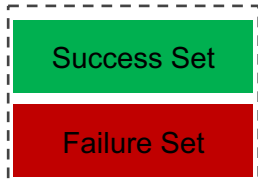


Success Set

Failure Set

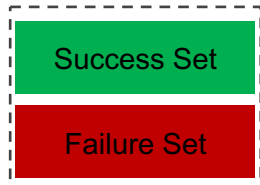
# Relationship Inference

- Run & Record
- Statistical Inference



# Relationship Inference

- Run & Record
- Statistical Inference



Statistical Inference

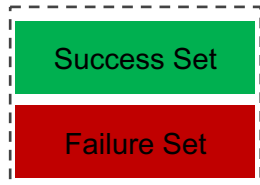


	API <sub>0</sub>			API <sub>1</sub>			API <sub>2</sub>			...
	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	...
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

# Relationship Inference

- Run & Record
- Statistical Inference



Statistical Inference

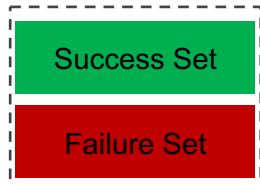


	API <sub>0</sub>			API <sub>1</sub>			API <sub>2</sub>			...
	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	...
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

# Relationship Inference

- Run & Record
- Statistical Inference



Statistical Inference

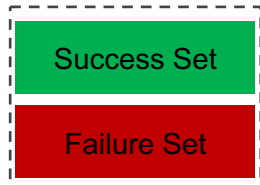


	API <sub>0</sub>			API <sub>1</sub>			API <sub>2</sub>			...
	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	...
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

# Relationship Inference

- Run & Record
- Statistical Inference



Statistical Inference

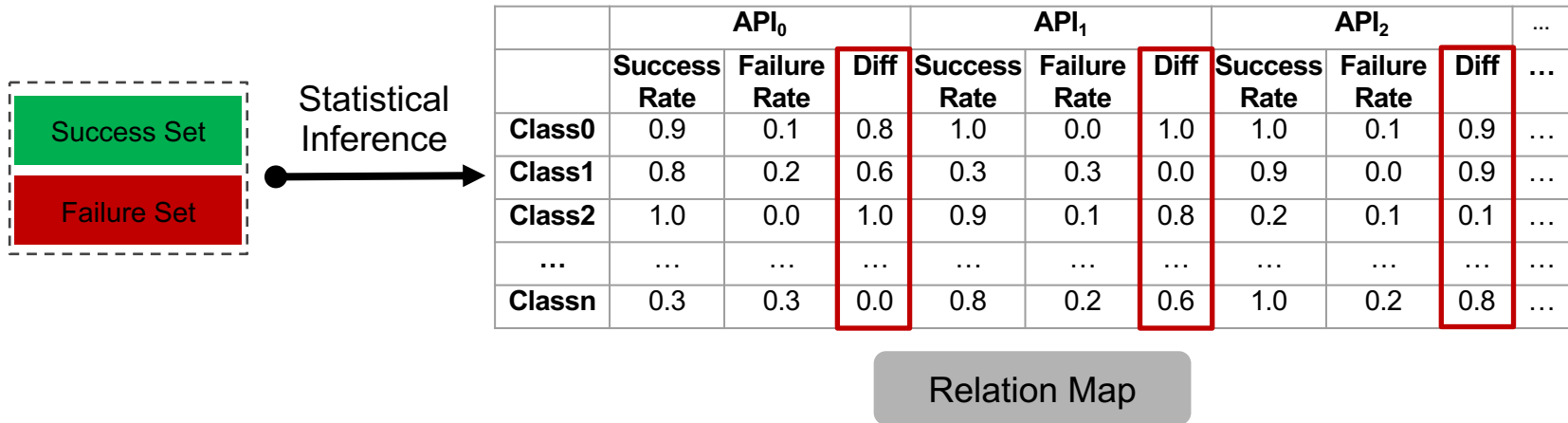


	API <sub>0</sub>			API <sub>1</sub>			API <sub>2</sub>			...
	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	Success Rate	Failure Rate	Diff	...
Class0	0.9	0.1	0.8	1.0	0.0	1.0	1.0	0.1	0.9	...
Class1	0.8	0.2	0.6	0.3	0.3	0.0	0.9	0.0	0.9	...
Class2	1.0	0.0	1.0	0.9	0.1	0.8	0.2	0.1	0.1	...
...	...	...	...	...	...	...	...	...	...	...
Classn	0.3	0.3	0.0	0.8	0.2	0.6	1.0	0.2	0.8	...

Relation Map

# Relationship Inference

- Run & Record
- Statistical Inference



# Relationship Guided Mutation

	<b>API<sub>0</sub></b>	<b>API<sub>1</sub></b>	<b>API<sub>2</sub></b>	...
<b>Class0</b>	0.8	1.0	0.9	...
<b>Class1</b>	0.6	0.0	0.9	...
<b>Class2</b>	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map



# Relationship Guided Mutation

	<b>API<sub>0</sub></b>	<b>API<sub>1</sub></b>	<b>API<sub>2</sub></b>	...
<b>Class0</b>	0.8	1.0	0.9	...
<b>Class1</b>	0.6	0.0	0.9	...
<b>Class2</b>	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

Target API  
group

# Relationship Guided Mutation

	<b>API<sub>0</sub></b>	<b>API<sub>1</sub></b>	<b>API<sub>2</sub></b>	...
<b>Class0</b>	0.8	1.0	0.9	...
<b>Class1</b>	0.6	0.0	0.9	...
<b>Class2</b>	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map



# Relationship Guided Mutation

	<b>API<sub>0</sub></b>	<b>API<sub>1</sub></b>	<b>API<sub>2</sub></b>	...
<b>Class0</b>	0.8	1.0	0.9	...
<b>Class1</b>	0.6	0.0	0.9	...
<b>Class2</b>	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

Native objects

Target API  
group

Objects  
Selection

Scripts code

Script Generation

# Relationship Guided Mutation

	API <sub>0</sub>	API <sub>1</sub>	API <sub>2</sub>	...
Class0	0.8	1.0	0.9	...
Class1	0.6	0.0	0.9	...
Class2	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

$$P(\text{class } c) = \frac{\text{rate}(c)}{\sum_{\text{API}} \sum_i \text{RelationshipMap}[\text{API}][i].\text{rate}}$$

Calculate mutation probability

Native objects

Target API group

Objects Selection

Scripts code

Script Generation

# Relationship Guided Mutation

	API <sub>0</sub>	API <sub>1</sub>	API <sub>2</sub>	...
Class0	0.8	1.0	0.9	...
Class1	0.6	0.0	0.9	...
Class2	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

$$P(\text{class } c) = \frac{\text{rate}(c)}{\sum_{\text{API}} \sum_i \text{RelationshipMap}[\text{API}][i].\text{rate}}$$

Calculate mutation probability

Native objects

Target API group

Scripts code

Objects Selection

Objects Mutation

Script Generation

# Relationship Guided Mutation

	API <sub>0</sub>	API <sub>1</sub>	API <sub>2</sub>	...
Class0	0.8	1.0	0.9	...
Class1	0.6	0.0	0.9	...
Class2	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

$$P(\text{class } c) = \frac{\text{rate}(c)}{\sum_{\text{API}} \sum_i \text{RelationshipMap}[\text{API}][i].\text{rate}}$$

Calculate mutation probability

Attribute Mutation

Whole-object Mutation

Universal Mutation

Native objects

Target API group

Scripts code

Objects Selection

Objects Mutation

Script Generation

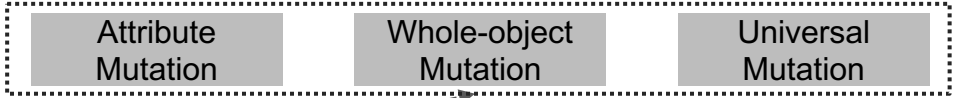
# Relationship Guided Mutation

	API <sub>0</sub>	API <sub>1</sub>	API <sub>2</sub>	...
Class0	0.8	1.0	0.9	...
Class1	0.6	0.0	0.9	...
Class2	1.0	0.8	0.1	...
...	...	...	...	...

Relation Map

$$P(\text{class } c) = \frac{\text{rate}(c)}{\sum_{\text{API}} \sum_i \text{RelationshipMap}[\text{API}][i].\text{rate}}$$

Calculate mutation probability



Native objects

Target API group

Scripts code

Objects Selection

Objects Mutation

Script Generation



# Implementation

- 4.3K lines of code in Python



# Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing

# Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing
- For scripts generation
  - modify Domato, and add block-level template

# Implementation

- 4.3K lines of code in Python
- Currently support
  - PDF: use PyPDF2 for parsing
  - Word: use zipfile and xml for parsing
- For scripts generation
  - modify Domato, and add block-level template
- Extensible and Portable

# Evaluations

- New bugs
- Comparison
  - Bug finding
  - Branch coverage

# New Bugs (in four months)

---

	Adobe Acrobat	Foxit Reader	Microsoft Word	<b>Total</b>
use-after-free	12	18	3	<b>33</b>
buffer overflow	4	8	5	<b>17</b>
buffer error	6	1	0	<b>7</b>
null ptr deref	30	22	8	<b>60</b>
stack exhaustion	6	4	0	<b>10</b>
access violation	2	2	1	<b>5</b>
others	0	1	1	<b>2</b>
<b>Total</b>	<b>60</b>	<b>56</b>	<b>18</b>	<b>134</b>

---

# New Bugs (in four months)

	Adobe Acrobat	Foxit Reader	Microsoft Word	Total
use-after-free	12	18	3	<b>33</b>
buffer overflow	4	8	5	<b>17</b>
buffer error	6	1	0	<b>7</b>
null ptr deref	30	22	8	<b>60</b>
stack exhaustion	6	4	0	<b>10</b>
access violation	2	2	1	<b>5</b>
others	0	1	1	<b>2</b>
<b>Total</b>	<b>60</b>	<b>56</b>	<b>18</b>	<b>134</b>

# New Bugs (in four months)

	Adobe Acrobat	Foxit Reader	Microsoft Word	Total
use-after-free	12	18	3	<b>33</b>
buffer overflow	4	8	5	<b>17</b>
buffer error	6	1	0	<b>7</b>
null ptr deref	30	22	8	<b>60</b>
stack exhaustion	6	4	0	<b>10</b>
access violation	2	2	1	<b>5</b>
others	0	1	1	<b>2</b>
<b>Total</b>	<b>60</b>	<b>56</b>	<b>18</b>	<b>134</b>

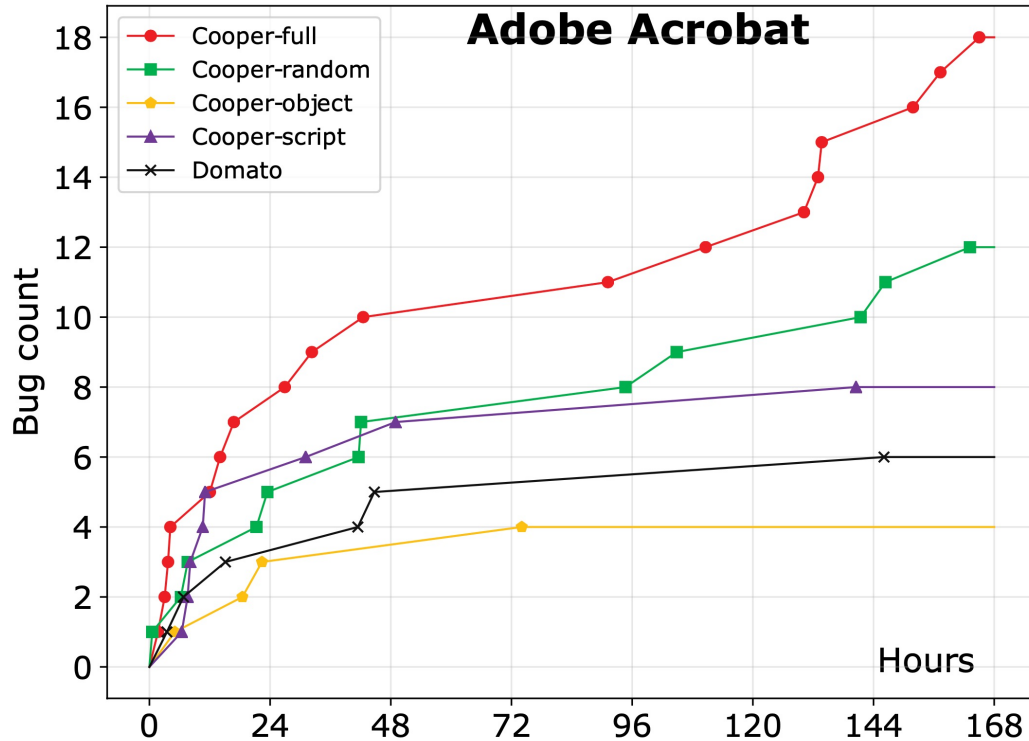
- 33 CVE, 59 fixed
- 22K dollars bounty
- 90 APIs & 11 object classes

# Comparing different configurations and tools

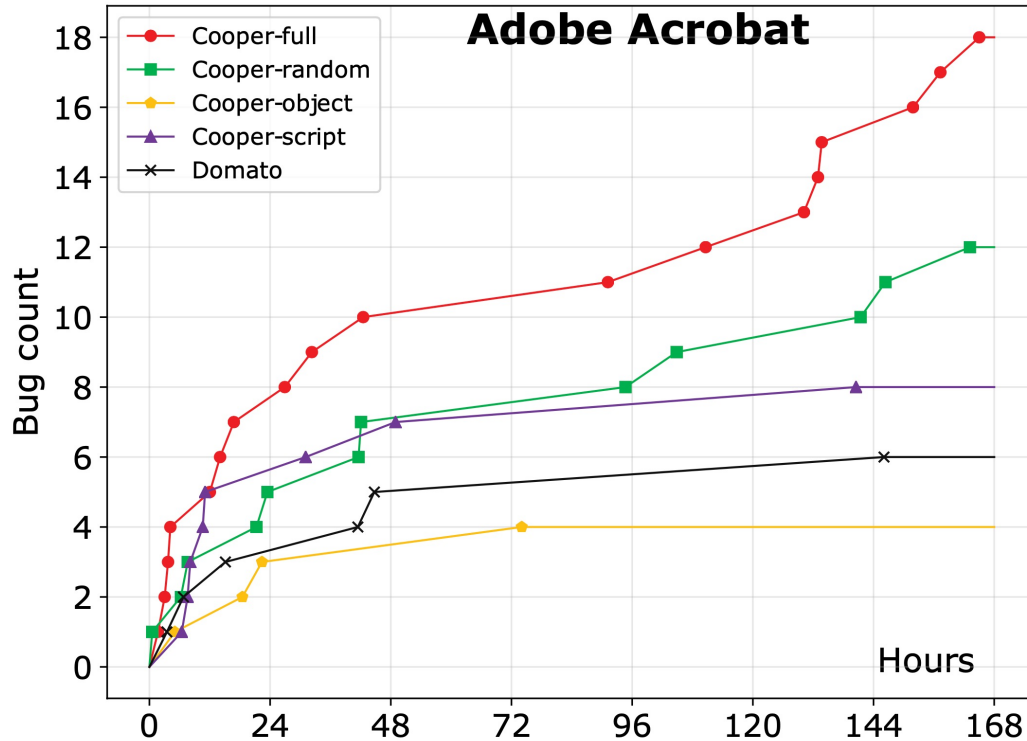
Experiment	Relation Guidance	Object Mutation	Script Generation
Cooper-full	●	●	●
Cooper-random	○	●	●
Cooper-object	●	●	○
Cooper-script	○	○	●
Domato	○	○	◐



# Bug finding with different configurations (one week)

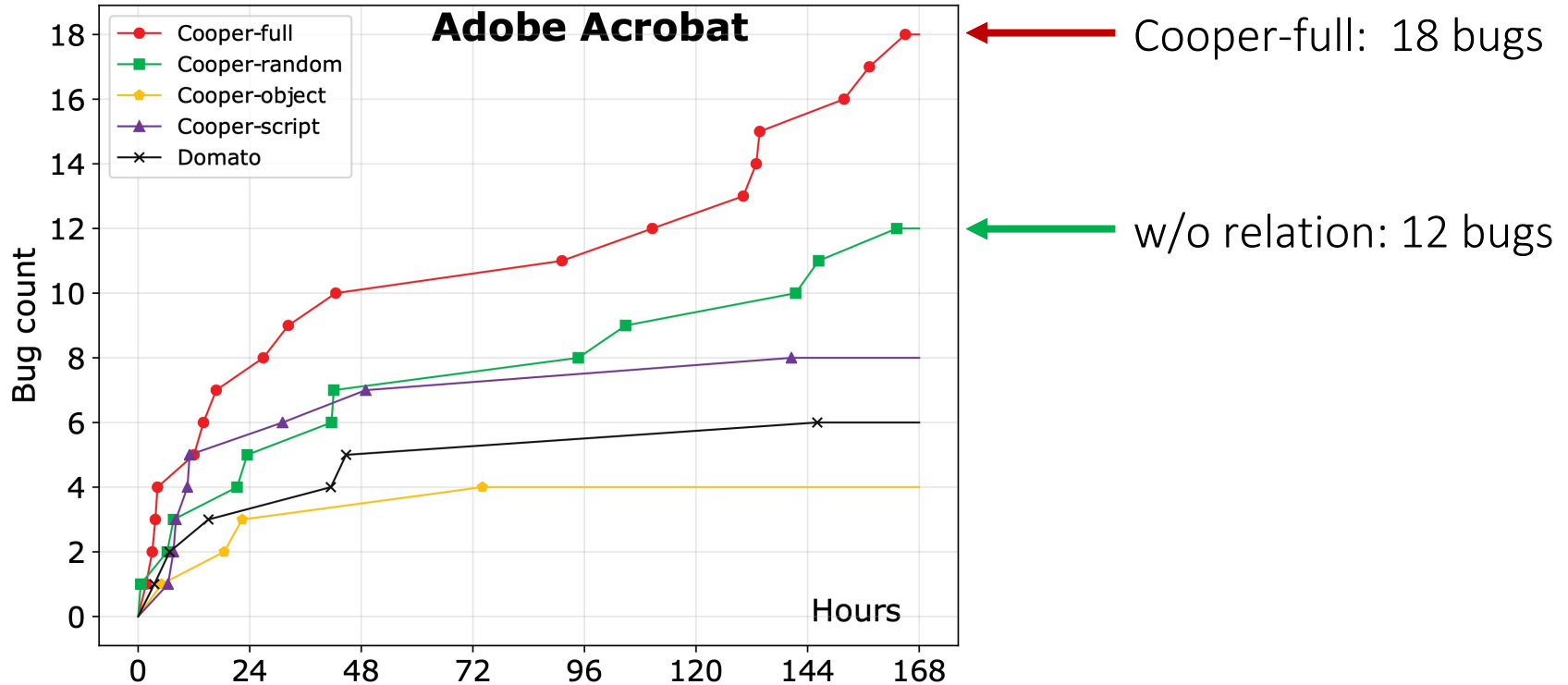


# Bug finding with different configurations (one week)

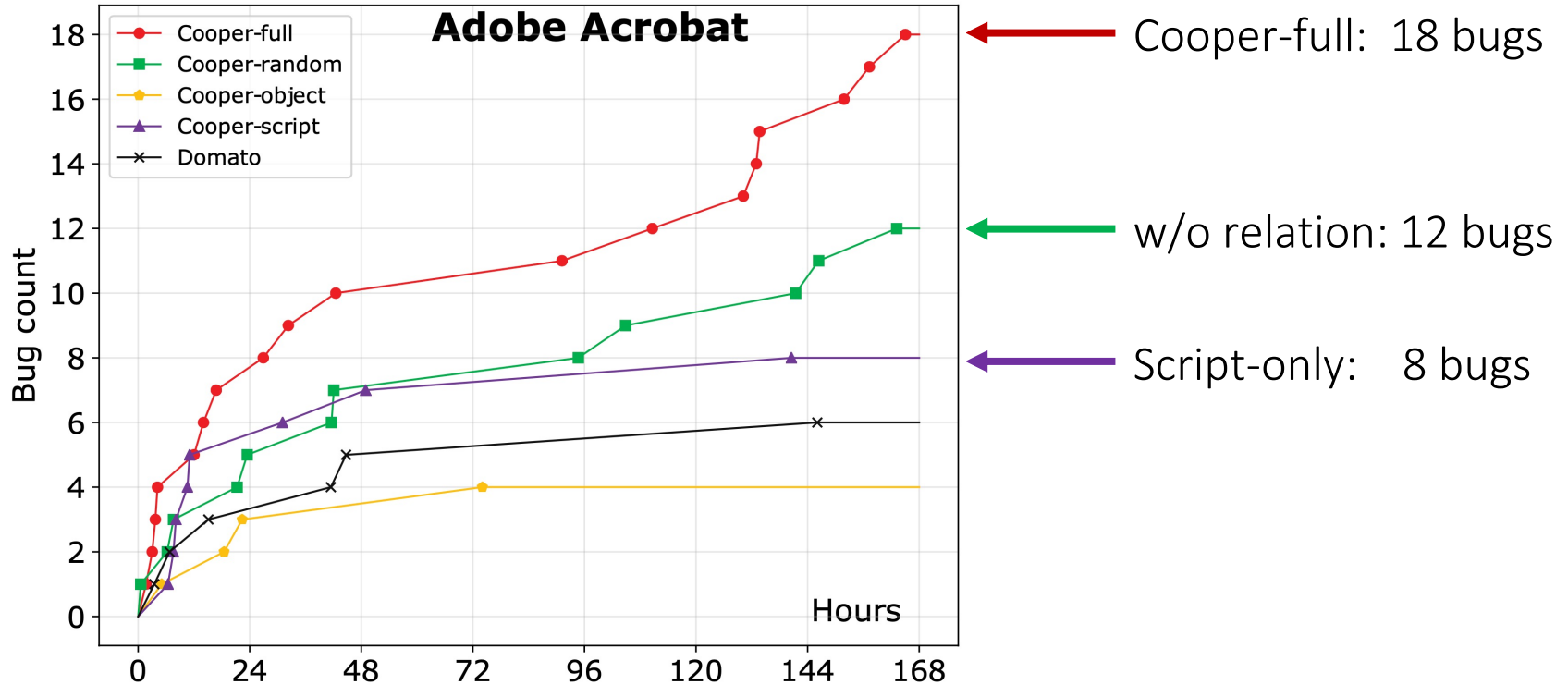


← Cooper-full: 18 bugs

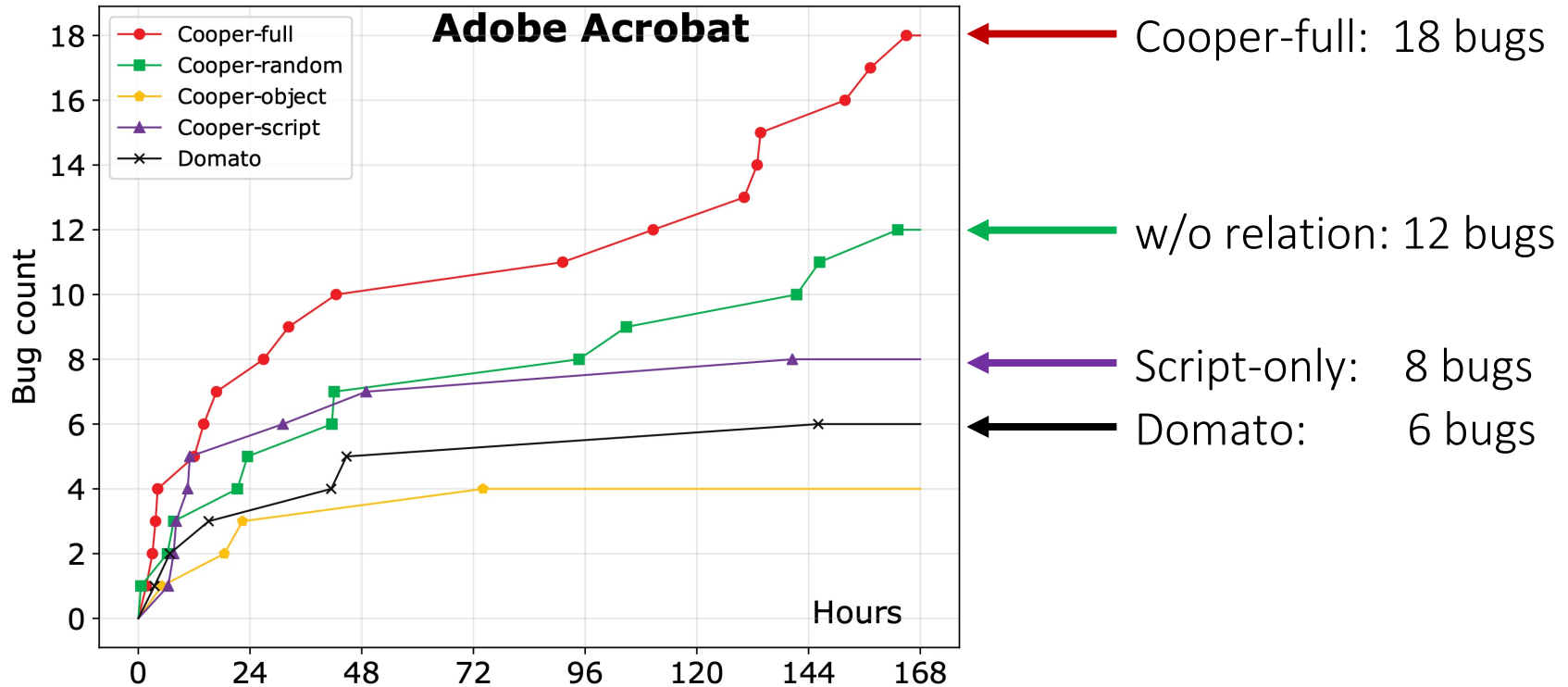
# Bug finding with different configurations (one week)



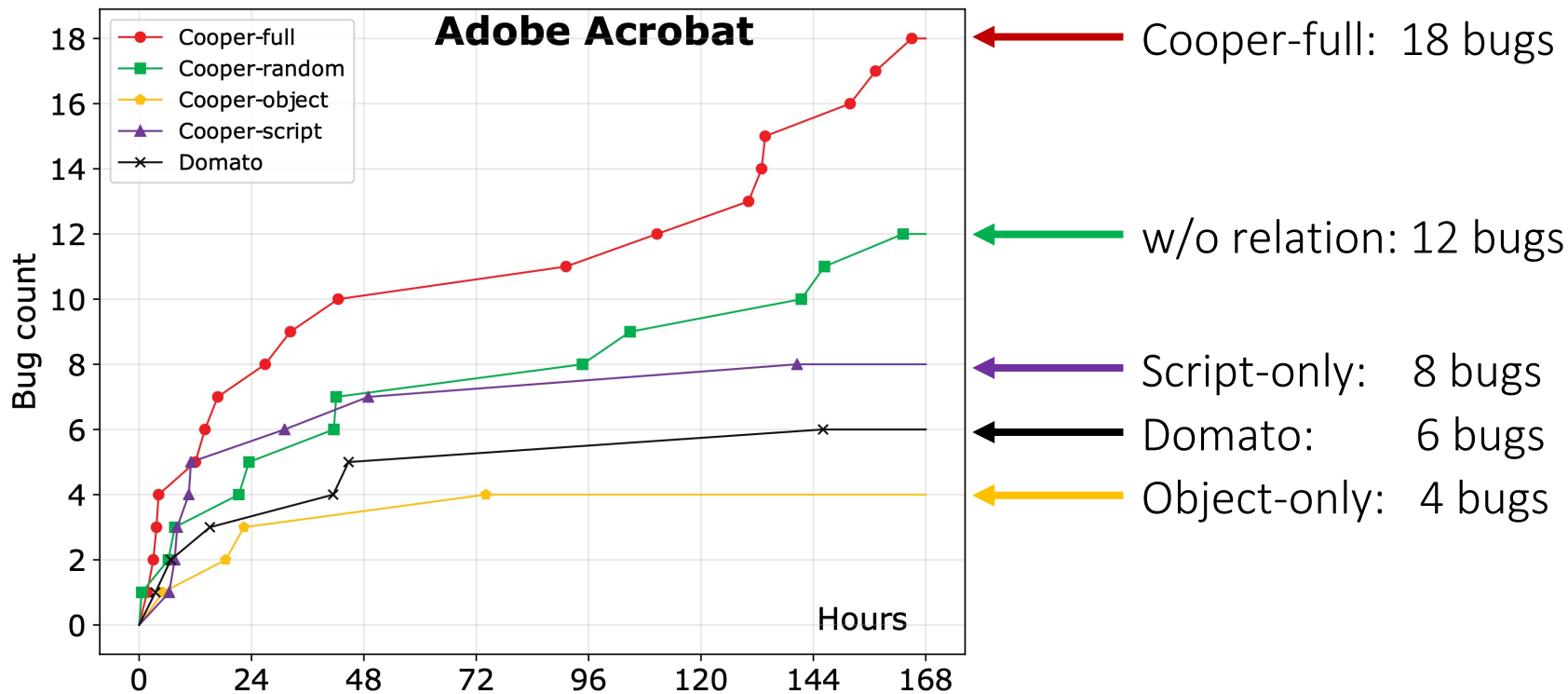
# Bug finding with different configurations (one week)



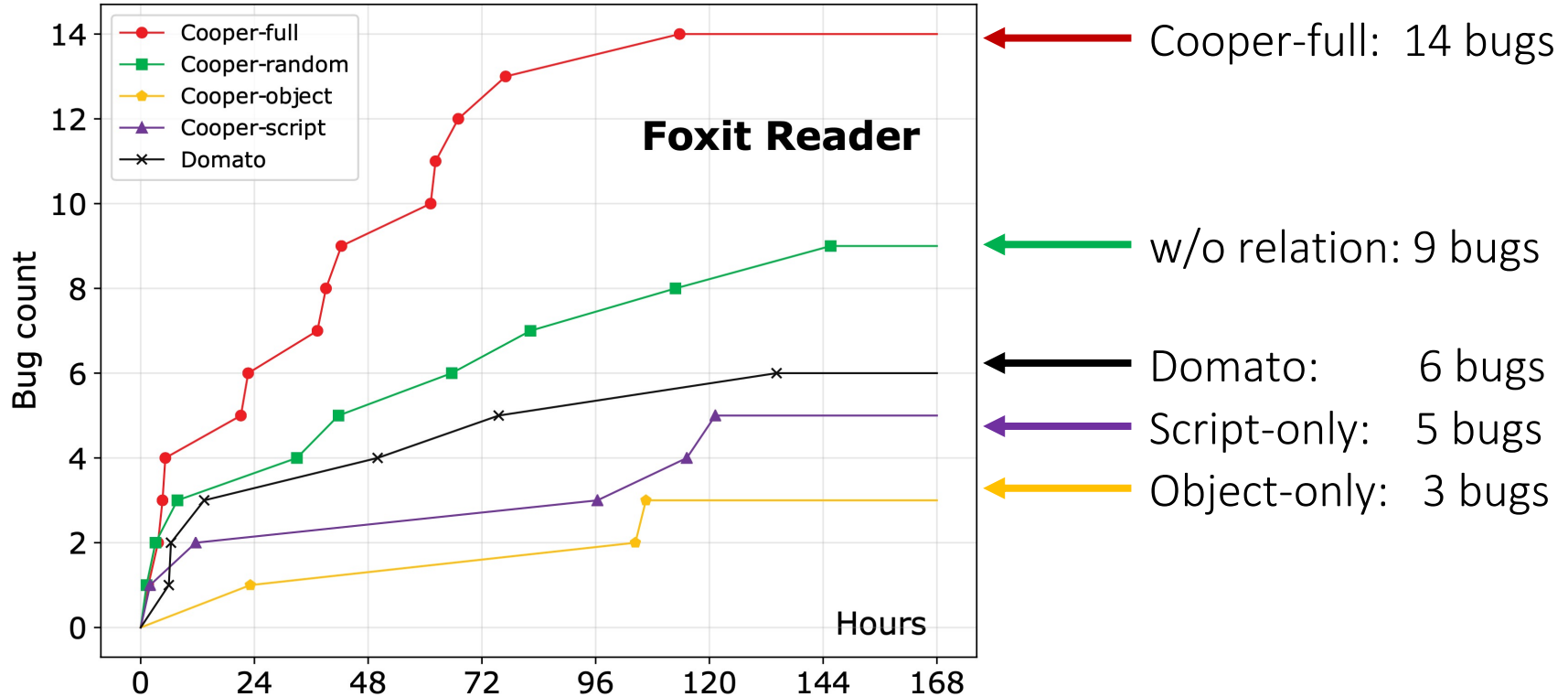
# Bug finding with different configurations (one week)



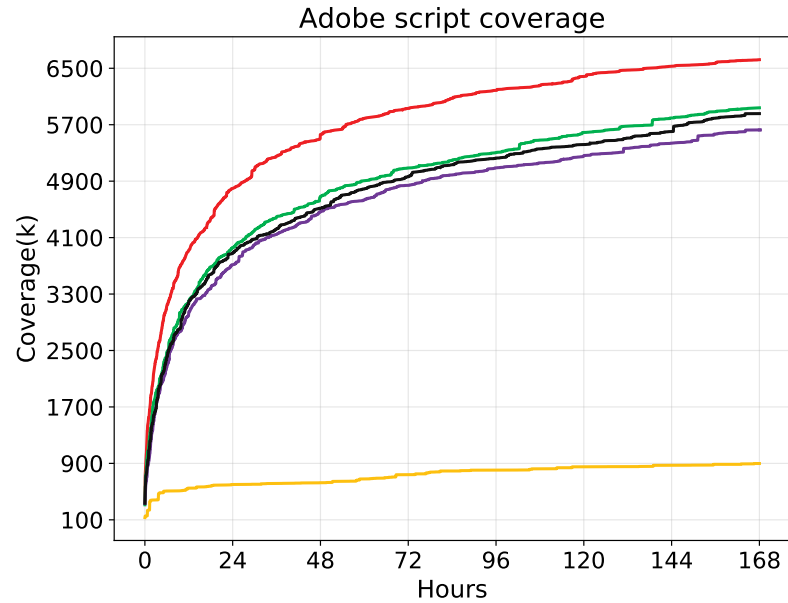
# Bug finding with different configurations (one week)



# Bug finding with different configurations (one week)

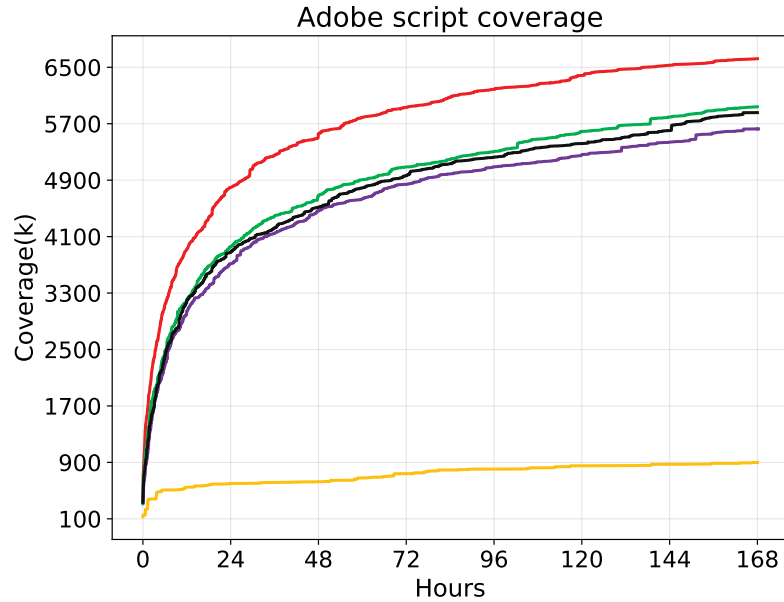


# Branch Coverage (one week)



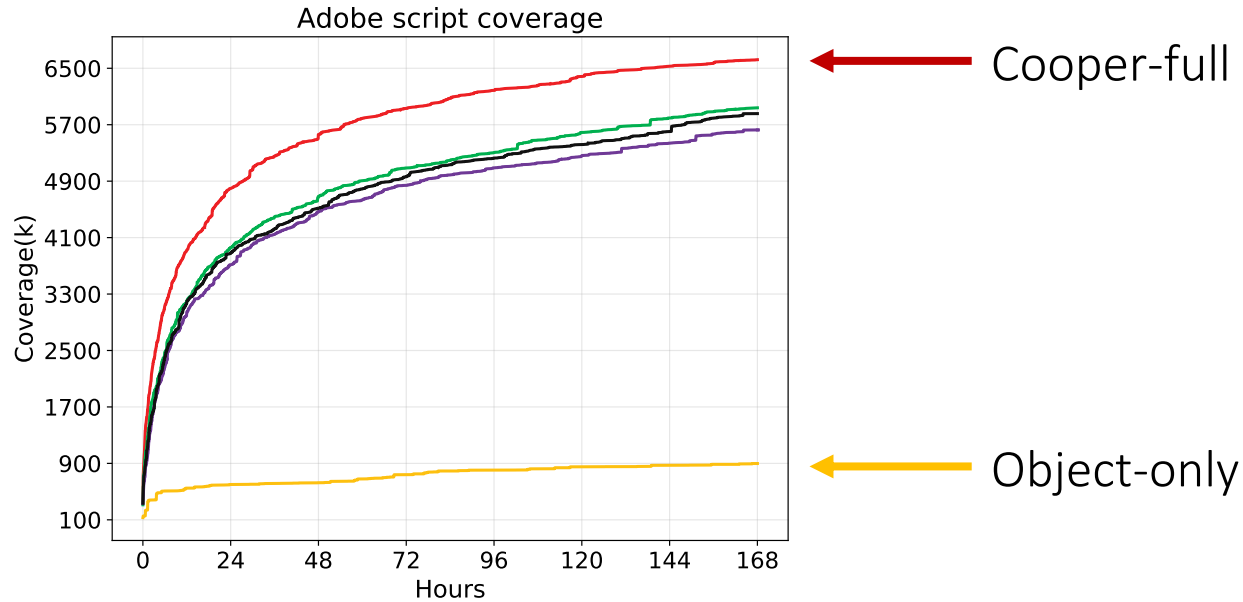


# Branch Coverage (one week)

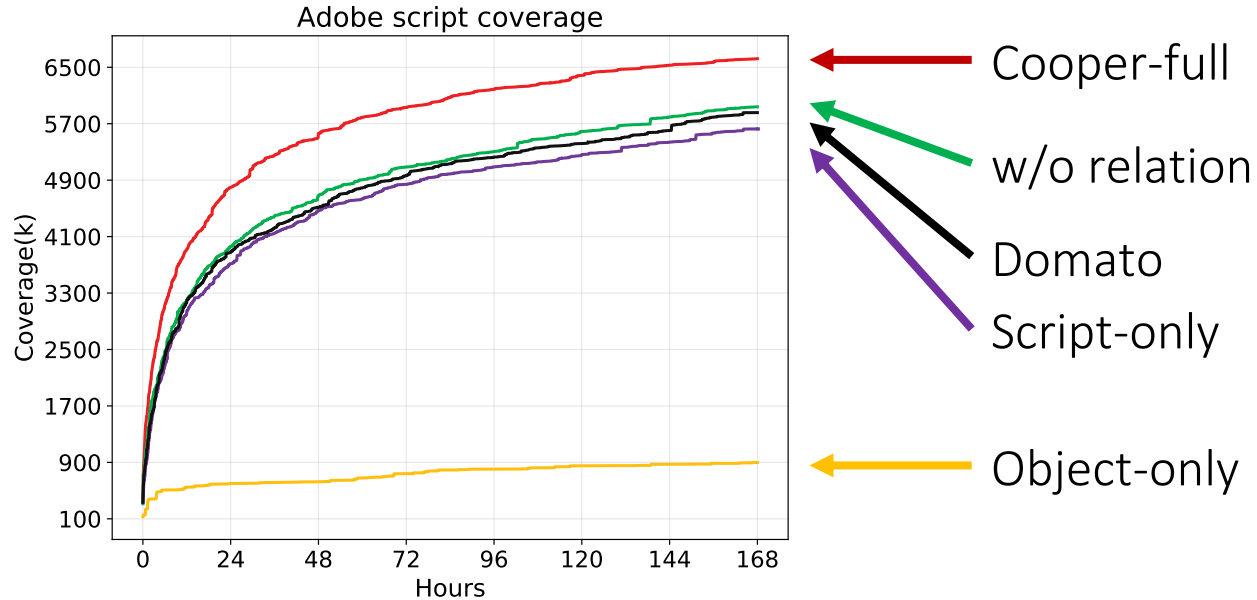


← Cooper-full

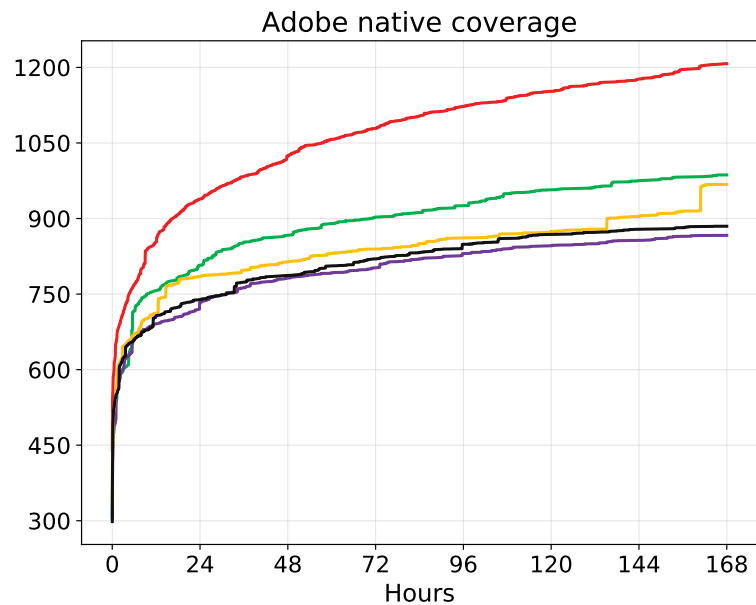
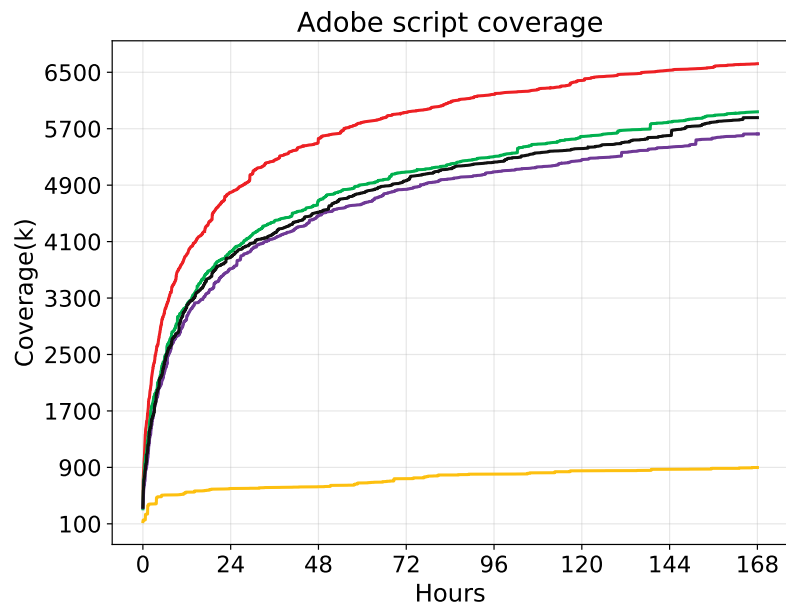
# Branch Coverage (one week)



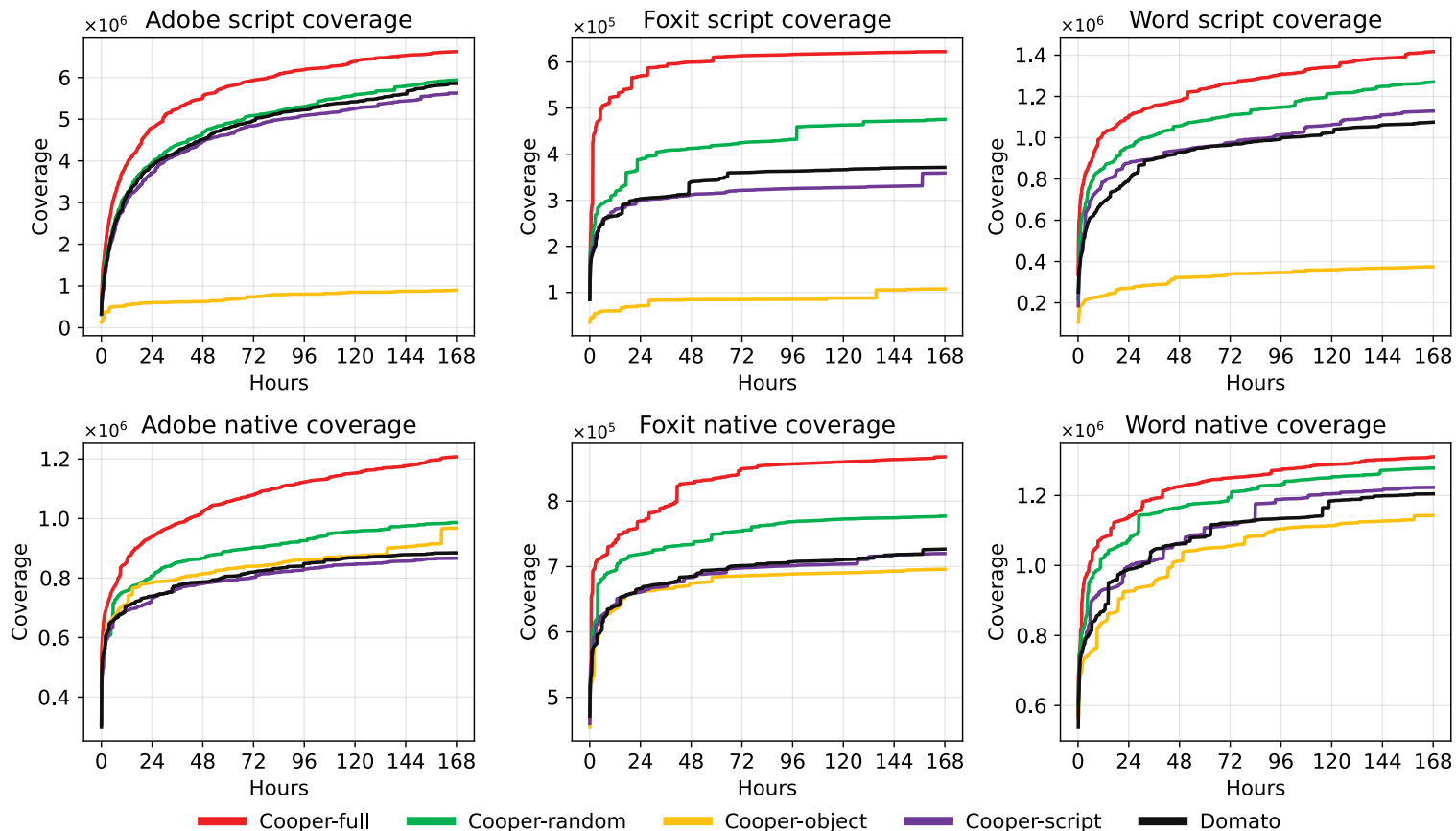
# Branch Coverage (one week)



# Branch Coverage (one week)



# Branch Coverage (one week)



# Conclusion

# Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages

# Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty



# Conclusion

- Cooperative mutation
  - effectively test binding code of scripting languages
- 134 bugs in Adobe Acrobat, Foxit Reader, and Microsoft Word
  - 33 CVE and 22K dollars bounty
- Code at: <https://github.com/TCA-ISCAS/Cooper>

Question?