Testing the Binding Code of Scripting Languages with Cooperative Mutation

Peng Xu    Yanhao Wang    Hong Hu    Purui Su
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-IAC 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

Update Google Chrome right now to patch a dangerous exploit

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

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

Input

PDF
Document processing programs

- Input: Native objects + Scripts code
Document processing programs

- Input: Native objects + Scripts code

```
1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
4 0 obj ... endobj
stream
  zoomType = refW
endstream
```
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
Motivating Example

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

How to trigger this vulnerability?
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.

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

```plaintext
%PDF-1.3
1 0 obj << /Pages 2 0 R >> endobj
2 0 obj << /Kids [ 3 0 R ] >> endobj
3 0 obj << /Resources << /Font << /TT1 4 0 R >> >>
   /AA << /O << /S /JavaScript
   /JS 5 0 R >> >> >> endobj
4 0 obj << /FirstChar 0
   /Widths [ 778 778 ... 556 500 ]
   /LastChar 255 >> endobj
5 0 obj << /Length 539 >>
stream
   this.zoomType=zoomtype.refW;
% Trigger the bug
endstream
endobj
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;`

One-dimensional-mutation cannot trigger this vulnerability
Our Solution: Cooperative Mutation
Our Solution: Cooperative Mutation
Our Solution: Cooperative Mutation

Native mutation

Cooperative Mutation

Scripts mutation

PDF file

native objects

1 0 obj ... endobj
2 0 obj ... endobj
3 0 obj ... endobj
4 0 obj ... endobj

JS code

stream
zoomType = refW
endstream

Adobe Acrobat

parser
GUI
reflow.api
binding layer
JavaScript engine
How to mutate objects & code *cooperatively*?

Native objects

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

Binding layer

Scripts code

stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream

Mutate Native objects

Mutate Scripts code
How to mutate objects & code *cooperatively*?

Native objects

| 1 0 obj ... endobj |
| 2 0 obj ... endobj |
| 3 0 obj ... endobj |
| 4 0 obj ... endobj |
| 5 0 obj ... endobj |
| 6 0 obj ... endobj |
| ... |
| n 0 obj ... endobj |

Binding layer

Scripts code

stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream

Mutate Native objects  Relationship  Mutate Scripts code
How to mutate objects & code *cooperatively*?

Native objects

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

Binding layer

stream
statement1;
statement2;
statement3;
statement4;
statement5;
...
endstream

Scripts code

Mutate Native objects

Relationship

Mutate Scripts code
Challenges

Native objects

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

Binding layer

stream
statement1;
statement2;
statement3;
statement4;
statement5;
...  
endstream

Scripts code
Challenges

1. How to infer relationship?
Challenges

1. How to infer relationship?

2. How to guide mutation?

Native objects

Binding layer

Scripts code
Challenges

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

1. How to infer relationship?
2. How to guide mutation?

Native objects

Binding layer

Scripts code
Challenges

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

1. How to infer relationship?

2. How to guide 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.

1. How to infer relationship?

2. How to guide mutation?
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?

1. How to infer relationship?

2. How to guide mutation?
Cooper Overview
Cooper Overview

Cluster input objects to semantic-similar classes.
infer relation between native input and script code
Cooper Overview

Use the inferred relation to guide mutation.
Object Clustering
Object Clustering

\[ O: \text{object} = \{ A_0: \text{name}_0 = \text{object}_0, \\
A_1: \text{name}_1 = \text{object}_1, \\
A_2: \text{name}_2 = \text{object}_2, \\
... = ... \} \]
Object Clustering

\[ O: \text{object} = \left\{ \begin{array}{l}
A_0: \text{name}_0 = \text{object}_0, \\
A_1: \text{name}_1 = \text{object}_1, \\
A_2: \text{name}_2 = \text{object}_2, \\
\ldots &= \ldots
\end{array} \right\} \]

Name contains semantic information
Object Clustering

O:object = \[
\begin{align*}
A_0: \text{name}_0 &= \text{object}_0, \\
A_1: \text{name}_1 &= \text{object}_1, \\
A_2: \text{name}_2 &= \text{object}_2, \\
\ldots &= \ldots
\end{align*}
\]

Name contains semantic information

- Clustering objects with \text{name}
Object Clustering

**O**: object = \{ A_0: \text{name}_0 = \text{object}_0, A_1: \text{name}_1 = \text{object}_1, A_2: \text{name}_2 = \text{object}_2, \ldots = \ldots \}

Name contains semantic information

- Clustering objects with **name**
Object Clustering

$O: object = \{ A_0: \text{name}_0 = \text{object}_0, \newline A_1: \text{name}_1 = \text{object}_1, \newline A_2: \text{name}_2 = \text{object}_2, \newline \ldots = \ldots \}$

- **Name contains semantic information**

- Clustering objects with **name**
Object Clustering

O: object = \{ A_0: \text{name}_0 = \text{object}_0, A_1: \text{name}_1 = \text{object}_1, \ldots \}

- Clustering objects with \text{name}

Name contains semantic information
Object Clustering

- Spliting and merging classes with attribute similarity
Object Clustering

- Spliting and merging classes with attribute similarity

\[
\text{Sim}(A, B) = \frac{2(|A \cap B|)}{|A| + |B|}
\]
Object Clustering

- Spliting and merging classes with attribute similarity

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

if sim < \(\theta_s\)

Big Class → Small Classes
Object Clustering

- Spliting and merging classes with attribute similarity

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

- If \( \text{sim} < \theta_s \)
- If \( \text{sim} > \theta_m \)

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

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); }

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>obj</th>
<th>...</th>
<th>endobj</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>obj</td>
<td>...</td>
<td>endobj</td>
</tr>
</tbody>
</table>
try {
    var annots = this.getAnnot();
    app.alert(annots.length + " Annots Found");
} catch (e) { app.alert("ERROR" + e); }
Run & Record

Warning: JavaScript Window -

2 Annots Found

Success Set

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

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

**Warning: JavaScript Window -**

- 2 Annots Found

#### Success Set

**Warning: JavaScript Window -**

- 0 Annots Found

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

Run & Record

Warning: JavaScript Window -

- 2 Annots Found

Success Set

Warning: JavaScript Window -

- 0 Annots Found

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

Warning: JavaScript Window -

- ERROR
### Run & Record

**Success Set**

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

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

**Failure Set**

```
0 Annots Found
```

**ERROR**
Relationship Inference

● Run & Record

Warning: JavaScript Window -
2 Annots Found

Success Set

Warning: JavaScript Window -
0 Annots Found

Failure Set

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

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

Success Set

Failure Set
Relationship Inference

- Run & Record
- Statistical Inference
## Relationship Inference

- **Run & Record**
- **Statistical Inference**

<table>
<thead>
<tr>
<th></th>
<th>$API_0$</th>
<th></th>
<th>$API_1$</th>
<th></th>
<th>$API_2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Success Rate</td>
<td>Failure Rate</td>
<td>Diff</td>
<td>Success Rate</td>
<td>Failure Rate</td>
<td>Diff</td>
</tr>
<tr>
<td>Class0</td>
<td>0.9</td>
<td>0.1</td>
<td>0.8</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Class1</td>
<td>0.8</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td>Class2</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Classn</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>0.8</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**Relation Map**
## Relationship Inference

- Run & Record
- Statistical Inference

| Class  | API₀ Success Rate | API₀ Failure Rate | API₀ Diff | API₁ Success Rate | API₁ Failure Rate | API₁ Diff | API₂ Success Rate | API₂ Failure Rate | API₂ 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

| Class | Success Rate | Failure Rate | API\(_0\) | Success Rate | Failure Rate | Diff | API\(_1\) | Success Rate | Failure Rate | Diff | API\(_2\) | Success Rate | Failure Rate | Diff | ...
|-------|--------------|--------------|-----------|--------------|--------------|------|-----------|--------------|--------------|------|-----------|--------------|--------------|------|------
| Class0| 0.9          | 0.1          | 0.8       | 1.0          | 0.0          | 1.0  | 1.0       | 0.1          | 0.9          | 0.9  | ...
| Class1| 0.8          | 0.2          | 0.6       | 0.3          | 0.3          | 0.0  | 0.9       | 0.0          | 0.9          | 0.9  | ...
| Class2| 1.0          | 0.0          | 1.0       | 0.9          | 0.1          | 0.8  | 0.2       | 0.1          | 0.1          | 0.1  | ...
| Classn| 0.3          | 0.3          | 0.0       | 0.8          | 0.2          | 0.6  | 1.0       | 0.2          | 0.8          | 0.8  | ...

#### Relation Map
Relationship Inference

- Run & Record
- Statistical Inference

| Class  | API₀ | API₁ | API₂ | ...
|--------|------|------|------|------
|        | 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

<table>
<thead>
<tr>
<th>Class</th>
<th>API₀ Success Rate</th>
<th>API₀ Failure Rate</th>
<th>API₀ Diff</th>
<th>API₁ Success Rate</th>
<th>API₁ Failure Rate</th>
<th>API₁ Diff</th>
<th>API₂ Success Rate</th>
<th>API₂ Failure Rate</th>
<th>API₂ Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class₀</td>
<td>0.9</td>
<td>0.1</td>
<td>0.8</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>Class₁</td>
<td>0.8</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>0.9</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Class₂</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.1</td>
<td>0.8</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Classₙ</td>
<td>0.3</td>
<td>0.3</td>
<td>0.0</td>
<td>0.8</td>
<td>0.2</td>
<td>0.6</td>
<td>1.0</td>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

### Relation Map
Relationship Guided Mutation

<table>
<thead>
<tr>
<th></th>
<th>API₀</th>
<th>API₁</th>
<th>API₂</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class₀</td>
<td>0.8</td>
<td>1.0</td>
<td>0.9</td>
<td>…</td>
</tr>
<tr>
<td>Class₁</td>
<td>0.6</td>
<td>0.0</td>
<td>0.9</td>
<td>…</td>
</tr>
<tr>
<td>Class₂</td>
<td>1.0</td>
<td>0.8</td>
<td>0.1</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Relation Map
Relationship Guided Mutation

<table>
<thead>
<tr>
<th>API_0</th>
<th>API_1</th>
<th>API_2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class0</td>
<td>0.8</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Class1</td>
<td>0.6</td>
<td>0.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Class2</td>
<td>1.0</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Relationship Guided Mutation

<table>
<thead>
<tr>
<th></th>
<th>API₀</th>
<th>API₁</th>
<th>API₂</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class₀</td>
<td>0.8</td>
<td>1.0</td>
<td>0.9</td>
<td>...</td>
</tr>
<tr>
<td>Class₁</td>
<td>0.6</td>
<td>0.0</td>
<td>0.9</td>
<td>...</td>
</tr>
<tr>
<td>Class₂</td>
<td>1.0</td>
<td>0.8</td>
<td>0.1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Relation Map

Target API group

Scripts code

Script Generation
Relationship Guided Mutation

| Class   | API₀ | API₁ | API₂ | ...
|---------|------|------|------|------
| Class₀  | 0.8  | 1.0  | 0.9  | ...
| Class₁  | 0.6  | 0.0  | 0.9  | ...
| Class₂  | 1.0  | 0.8  | 0.1  | ...
|         |      |      |      | ...

Native objects

Target API group

Objects Selection

Scripts code

Script Generation
Relationship Guided Mutation

Calculate mutation probability

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

Relation Map

Native objects

Target API group

Objects Selection

Script Generation

Scripts code
Relationship Guided Mutation

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

| Class  | API0 | API1 | API2 | ...
|--------|------|------|------|------
| Class0 | 0.8  | 1.0  | 0.9  | ...
| Class1 | 0.6  | 0.0  | 0.9  | ...
| Class2 | 1.0  | 0.8  | 0.1  | ...
| ...    | ...  | ...  | ...  | ...

Native objects

Target API group

Scripts code

Objects Selection → Objects Mutation

Script Generation
Calculate mutation probability

Relation Map

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

Native objects

Target API group

Objects Selection

Objects Mutation

Script Generation

Attribute Mutation

Whole-object Mutation

Universal Mutation
Calculate mutation probability

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

<table>
<thead>
<tr>
<th>Bug Type</th>
<th>Adobe Acrobat</th>
<th>Foxit Reader</th>
<th>Microsoft Word</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>use-after-free</td>
<td>12</td>
<td>18</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>buffer overflow</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>buffer error</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>null ptr deref</td>
<td>30</td>
<td>22</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>stack exhaustion</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>access violation</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>others</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>60</strong></td>
<td><strong>56</strong></td>
<td><strong>18</strong></td>
<td><strong>134</strong></td>
</tr>
</tbody>
</table>
## New Bugs (in four months)

<table>
<thead>
<tr>
<th>Bug Type</th>
<th>Adobe Acrobat</th>
<th>Foxit Reader</th>
<th>Microsoft Word</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>use-after-free</td>
<td>12</td>
<td>18</td>
<td>3</td>
<td>33</td>
</tr>
<tr>
<td>buffer overflow</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>buffer error</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>null ptr deref</td>
<td>30</td>
<td>22</td>
<td>8</td>
<td>60</td>
</tr>
<tr>
<td>stack exhaustion</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>access violation</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>others</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>60</strong></td>
<td><strong>56</strong></td>
<td><strong>18</strong></td>
<td><strong>134</strong></td>
</tr>
</tbody>
</table>
## New Bugs (in four months)

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

- 33 CVE, 59 fixed
- 22K dollars bounty
- 90 APIs & 11 object classes
Comparing different configurations and tools

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Relation Guidance</th>
<th>Object Mutation</th>
<th>Script Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooper-full</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Cooper-random</td>
<td>○</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Cooper-object</td>
<td>●</td>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>Cooper-script</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
<tr>
<td>Domato</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
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)

Cooper-full: 18 bugs
w/o relation: 12 bugs
Bug finding with different configurations (one week)

- Cooper-full: 18 bugs
- w/o relation: 12 bugs
- Script-only: 8 bugs
Bug finding with different configurations (one week)

Cooper-full: 18 bugs

w/o relation: 12 bugs

Script-only: 8 bugs

Domato: 6 bugs
Bug finding with different configurations (one week)

- Cooper-full: 18 bugs
- w/o relation: 12 bugs
- Script-only: 8 bugs
- Domato: 6 bugs
- Object-only: 4 bugs
Bug finding with different configurations (one week)

- Cooper-full: 14 bugs
- w/o relation: 9 bugs
- Domato: 6 bugs
- Script-only: 5 bugs
- Object-only: 3 bugs
Branch Coverage (one week)
Branch Coverage (one week)
Branch Coverage (one week)

Adobe script coverage

- Cooper-full
- Object-only
Branch Coverage (one week)

Adobe script coverage

- Cooper-full
- w/o relation
- Domato
- Script-only
- Object-only
Branch Coverage (one week)

Adobe script coverage

Adobe native coverage
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](https://github.com/TCA-ISCAS/Cooper)
Question?