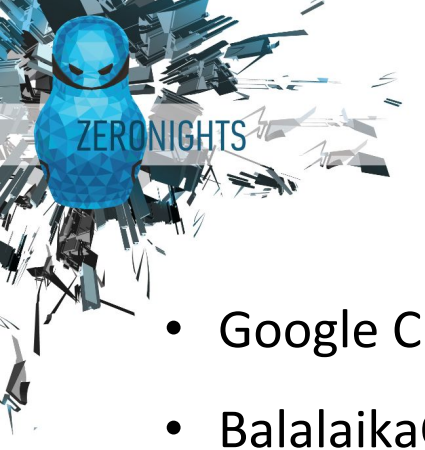
Modern Fuzzing of C/C++ projects

Max Moroz
Google



Bio

- Google Chrome Security team, Bugs--
- BalalaikaCr3w, LC↯BC
- CTF, BugBounty, etc

Agenda

1. TODO: write agenda
2. ???
3. Slides
4. Workshop
5.

My first year in university

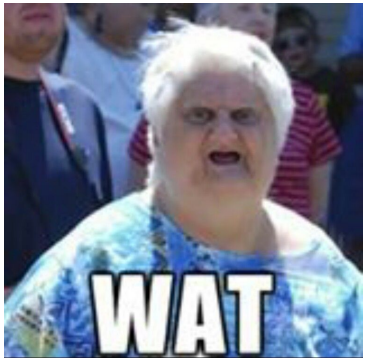
```
$ ./fact
Enter n to compute n! : 5
5! = 120
```

My first year in university

```
$ ./fact
Enter n to compute n! : 5
5! = 120

$ ./fact
Enter n to compute n! :
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
Segmentation fault (core dumped)
```

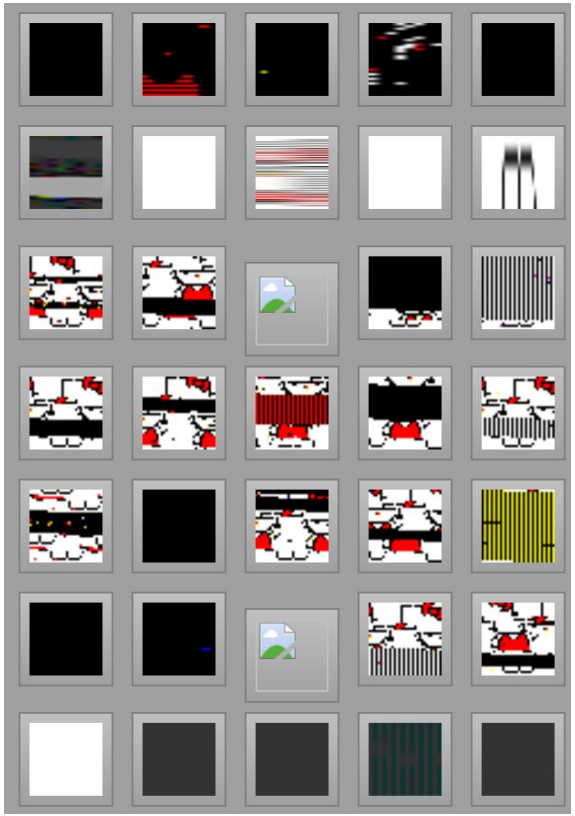
My first year in university

```
$ ./fact
Enter n to compute n! : 5
5! = 120

$ ./fact
Enter n to compute n! :
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
Segmentation fault (core dumped)

$ ./fact
Enter n to compute n! : 12345678990
-539222898! = 1
```



Fuzzing

Fuzzing



A software testing technique, often automated or semi-automated, that involves passing invalid, unexpected or random input to a program and monitor result for crashes, failed assertions, races, leaks, etc.

Unit testing vs. Fuzz testing

	Unit Testing	Old Fuzzing
Test small parts of code	✔	X
Can be automated	✔	✔
Regression testing	✔	✔ / X
Easy to write	✔	X
Looking for new bugs	✔ / X	✔✔✔
Looking for vulnerabilities	X	✔

Unit testing vs. Fuzz testing

	Unit Testing	Old Fuzzing	Modern Fuzzing
Test small parts of code	✔	X	✔
Can be automated	✔	✔	✔
Regression testing	✔	✔ / X	✔
Easy to write	✔	X	✔
Looking for new bugs	✔ / X	✔✔✔	✔✔✔✔✔✔
Looking for vulnerabilities	X	✔	✔

Vocabulary

- **Target**
  - Consumes an array of bytes
  - Calls the code we want to test
- **Fuzzer**
  - A tool that feed the target with different random inputs
- **Corpus**
  - A set of valid & invalid inputs for the target
  - Collected manually, by fuzzing, or by crawling

Fuzzer types

Overview

Fuzzer types

Generation Based

Generate from scratch with no prior state

Example

https://bugs.webkit.org/show_bug.cgi?id=60831

```
<script>
document.body = document.createElement('iframe');
</script>
```

Fuzzer types

Mutation Based

Mutate existing state based on some rules

Example

crbug.com/552046

```
--- orig.pdf
+++ crash.pdf
@@ -57,7 +57,7 @@
 /DecodeParms [null 8 0 R]
 /Type /XObject
 /Width 1760
-/Filter [/FlateDecode /DCTDecode]
+/Filter [/JBIG2Decode /DCTDecode]
 /Height 1248
 /Length 2277
```

Fuzzer types

Evolutionary

Generation or mutation based or both, in-process with code coverage feedback

Example

crbug.com/575205

```
SELECT'\xef(\xfb;DS\x1aLEETABL\xfeES'REGEX
P';0\t\tC LE|A*(\xc8*.+!*)*h*00\x0b$T''&'
```

Fuzzing in the past

Old school fuzzing

Routine

1. Generate an HTML page

Routine

1. Generate an HTML page
2. Write it to the disk

Routine

1. Generate an HTML page
2. Write it to the disk
3. Launch browser

Routine

1. Generate an HTML page
2. Write it to the disk
3. Launch browser
4. Open the page or serve it over HTTP

Routine

1. Generate an HTML page
2. Write it to the disk
3. Launch browser
4. Open the page or serve it over HTTP
5. Check if the browser crashed

Routine

1. Generate an HTML page
2. Write it to the disk
3. Launch browser
4. Open the page or serve it over HTTP
5. Check if the browser crashed
6. Close the browser

Let’s write some code

Lesson 02